

**ЎЗБЕКИСТОН РЕСПУБЛИКАСИ  
ОЛИЙ ВА ЎРТА МАХСУС ТАЪЛИМ ВАЗИРЛИГИ**

**ОЛИЙ ТАЪЛИМ ТИЗИМИ ПЕДАГОГ ВА РАЎБАР КАДРЛАРИНИ  
ҚАЙТА ТАЙЁРЛАШ ВА УЛАРНИНГ МАЛАКАСИНИ ОШИРИШНИ  
ТАШКИЛ ЭТИШ БОШ ИЛМИЙ - МЕТОДИК МАРКАЗИ**

**ТОШКЕНТ АХБОРОТ ТЕХНОЛОГИЯЛАРИ УНИВЕРСИТЕТИ  
ХУЗУРИДАГИ ПЕДАГОГ КАДРЛАРНИ ҚАЙТА ТАЙЁРЛАШ ВА  
УЛАРНИНГ МАЛАКАСИНИ ОШИРИШ ТАРМОҚ МАРКАЗИ**

**“ДАСТУРИЙ ИНЖИНИРИНГ”**

**йўналиши**

**“WEB ДАСТУРЛАШ”**

**МОДУЛИ БЎЙИЧА**

**Ў Қ У В – У С Л У Б И Й М А Ж М У А**

**ТОШКЕНТ - 2017**

**ЎЗБЕКИСТОН РЕСПУБЛИКАСИ**  
**ОЛИЙ ВА ЎРТА МАХСУС ТАЪЛИМ ВАЗИРЛИГИ**

**ОЛИЙ ТАЪЛИМ ТИЗИМИ ПЕДАГОГ ВА РАЎБАР КАДРЛАРИНИ**  
**ҚАЙТА ТАЙЁРЛАШ ВА УЛАРИНИГ МАЛАКАСИНИ ОШИРИШНИ**  
**ТАШКИЛ ЭТИШ БОШ ИЛМИЙ - МЕТОДИК МАРКАЗИ**

**ТОШКЕНТ АХБОРОТ ТЕХНОЛОГИЯЛАРИ УНИВЕРСИТЕТИ**  
**ҲУЗУРИДАГИ ПЕДАГОГ КАДРЛАРНИ ҚАЙТА ТАЙЁРЛАШ ВА**  
**УЛАРИНИГ МАЛАКАСИНИ ОШИРИШ ТАРМОҚ МАРКАЗИ**



---

**“WEB ДАСТУРЛАШ” МОДУЛИ**  
**БЎЙИЧА**

---

**ЎҚУВ – УСЛУБИЙ МАЖМУА**



**ТОШКЕНТ - 2017**

Мазкур ўқув-услугий мажмуа Олий ва ўрта махсус таълим  
вазирлигининг 201\_ йил \_\_\_\_\_даги \_\_\_\_-сонли буйруғи билан  
тасдиқланган ўқув режа ва дастур асосида тайёрланди.

---

Тузувчилар: ТАТУ “Тизимли ва амалий дастурлаштириш”  
Кафедраси доценти К.Керимов

ТАТУ “Тизимли ва амалий дастурлаштириш”  
кафедраси катта ўқитувчиси М.Бабажанов

Такризчи: ТАТУ, АКТ бўйича маслаҳатчи проректори,  
Жанубий Кореялик мутахассис Ли Чул Су

---

Ўқув -услугий мажмуа Тошкент ахборот технологиялари  
университети Кенгашининг қарори билан нашрга тавсия  
қилинган (201\_ йил \_\_\_\_\_даги \_\_\_\_ - сонли баённома)

# TO CURRICULUM FOR THE «SOFTWARE ENGINEERING» COURSE OF PROFESSIONAL DEVELOPMENT AND RETRAINING ACADEMIC STAFF OF HIGHER EDUCATION

## REVIEW

The training program of direction "Software Engineering" contains 8 blocks. A typical curriculum includes 5 modules on special direction that operating systems, system analysis, software engineering, mobile application development and web application development.

Actually, a typical curriculum includes legal framework and legal university standards, modern educational technology and high pedagogical skills, use of information and communication technologies in pedagogical process, a foreign language, the basics of system analysis and application of the decision, the scientific practical work on the basis of special directions, new methods of creating education and process, creativity and competence of the teacher.

The program includes the training of teachers of subjects in the field of education, training and skills development, quality and preparation of the general qualification requirements and training plans formed the basis of the teaching staff of higher education institutions in the sphere of modern education and innovative technologies, architecture of computer, basics of operating systems, process management and file systems, introduction to software engineering, methods of development and prediction software, models of the process of creating software, software testing, software quality the principles of developing mobile applications that to study of basic principles of user interface applications for mobile devices and new technology of web programming as building secure web applications.

The title and content of the curriculum of direction "Software Engineering" corresponds to the typical curriculum specialty and educational standards, qualification requirements to a specialist.

Vice rector of ICT, TUIT



Chul Soo LEE

## EXPERT CONCLUSION

### TO THE EDUCATIONAL-METHODOLOGICAL COMPLEX FOR THE COURSE OF RETRAINING PEDAGOGUE CADRES OF HIGHER EDUCATION ORGANIZATIONS IN THE DIRECTION OF “SOFTWARE ENGINEERING”

This educational-methodological complex was developed in accordance with defined requirements.

It consists of the:

- syllabus;
- theoretical and practical materials;
- assessment;
- presentations on every topic;
- glossary;
- tests;
- list of references.

The syllabus is written correctly. The sequence of topics proposed for study, focused on high-quality learning. Calendar-thematic plan corresponds to its content of the working program on discipline. Tests vary, allow to adequately assess the level of students' knowledge on the subject. Methodical recommendations for practical exercises provide the formation of basic skills to carry out research in the process of scientific knowledge and the theoretical foundation of professional tasks.

Slide support lecture material is accurate and specific, it promotes better assimilation of discipline. The presented educational-methodical complexes in the direction of "Software Engineering" informative, has a practical orientation, includes a sufficient number of diverse elements aimed at developing the mental and creative abilities of students.

In general, educational-methodological complexes "Software Engineering" promotes quality possession students professional competence.

Vice rector of ICT, TUE



Chul Soo LEE

**ТОШКЕНТ АХБОРОТ ТЕХНОЛОГИЯЛАРИ УНИВЕРСИТЕТИ**  
**ХУЗУРИДАГИ ПЕДАГОГ КАДРЛАРНИ ҚАЙТА ТАЙЁРЛАШ ВА УЛАРНИНГ**  
**МАЛАКАСИНИ ОШИРИШ ТАРМОҚ МАРКАЗИДА “ДАСТУРИЙ**  
**ИНЖИНИРИНГ” ЙЎНАЛИШИ БЎЙИЧА Т.Ф.Н. К.Ф.КЕРИМОВ, КАТТА ИЛМИЙ**  
**ХОДИМ-ИЗЛАНУВЧИ Т.А.КУЧКОРОВ, АССИСТЕНТ З.Б.ЮЛДАШЕВЛАР**  
**ТОМОНИДАН ЯРАТИЛГАН ЎҚУВ ДАСТУРИГА**

**ТА Қ Р И З**

Ушбу тақризланаётган ўқув дастур тармоқ марказида “Дастурий инжиниринг” йўналиши бўйича малака оширувчилар учун мўлжалланган бўлиб, мобулла бўйича назарий ва амалий билимлар кўрсатиб ўтилган.

Дастур мазмуни олий таълимнинг норматив-ҳуқуқий асослари ва қонунчилик нормалари, илғор таълим технологиялари ва педагогик маҳорат, таълим жараёнларида ахборот-коммуникация технологияларини қўллаш, амалий хорижий тил, тизимли таҳлил ва қарор қабул қилиш асослари, махсус фанлар негизида илмий ва амалий тадқиқотлар, технологик таракқиёт ва ўқув жараёнини ташкил этишнинг замонавий услублари бўйича сўнги ютуқлар, педагогнинг касбий компетентлиги ва креативлиги, глобал Интернет тармоғи, мультимедиа тизимлари ва масофадан ўқитиш усулларини ўзлаштириш бўйича янги билим, кўникма ва малакаларини шакллантиришни назарда тутди.

Дастурда мутахассислақ бўйича Операцион тизимлар, Дастурлашда тизимли таҳлил, Дастурий инжиниринг, Мобил иловалар яратиш, Веб иловалар яратиш каби модулла назарда тутилган бўлиб, ҳар бир модул бўйича асосий ва долзарб бўлган назарий ва амалий билимлар кўрсатилган. Жумладан, модулла бўйича операцион тизимлар (ОТ) вазифаси, функцияси, ҳисоблаш тизимлари таркибий қисмлари, ОТ ни синфларга ажратиш, ОТ архитектураси, ОТ қуриш асосий тамойиллари, ОТларда жараён ва ресурс тушунчаси, дастурий маҳсулотни баҳолашнинг асосий тамойиллари, дастурий маҳсулотни баҳолаш усуллари, дастурий маҳсулотларни сертификатлаш, дастурий маҳсулотни сотиш ва хизмат кўрсатиш, дастурий таъминотнинг асосий функциялари, тузилиши ва таркибий қисми, Web иловаларни ишлаб чиқиш асосий тушунчалари. HTML, XML, XHTML, WML. "клиент-сервер" технологияси. CSS (Cascading Style Sheets) асослари. JavaScript ифодалар, ўзгарувчилар, объектлар. PHP тилида дастурлаш асослари. MySQL маълумотлар базасида ишлаш, мобил иловалар яратиш учун дастурлаш муҳити. Андроид тизими учун Java дастурлаш тили. Андроид фойдаланувчи интерфейсини яратиш. Мобил иловаларда маълумотлар базаси. Мобил иловаларда тармоқли дастурлаш. Сервер билан ишлаш. JSON хизмати. Дастурий таъминот архитектураси. Талаблар ва уларни бошқариш. Дастурий таъминот ишлаб чиқиш усуллари. СММІ тушунчаси. СММІ бўйича жараёнларни етуклилик даражалари. Дастурий таъминот верификацияси ва валидацияси. Дастурий таъминотни тестлаш, маҳсулот нарҳини баҳолаш. Маҳсулот сифатини бошқариш. Дастурий маҳсулотни такомиллаштириш бўйича назарий ва амалий билимлар назарда тутилган.

Яратилган дастур ўз камрови билан “Дастурий инжиниринг” йўналишлиши бўйича кўрсатилган модулла бўйича тингловчиларга тўлақонли билим бериш даражасида тузилган бўлиб, шу дастур миқёсида дарсларни ташкил қилиш мақсадга мувофиқдир деб ҳисоблайман.

Дастурий инжиниринг факультети декани,  
т.ф.н. ассистент



Акбаралиев Б.Б.

# МУНДАРИЖА

1

Ишчи Дастур

2

Модулни ўқитишда  
фойдаланиладиган  
интерфаол таълим  
Методлари

3

Назарий  
Материаллар

4

Амалий  
Машғулот  
Материаллари

5

Кейслар Банки

6

Мустақил  
Таълим  
Мавзулари

7

Глоссарий

8

Адабиётлар Рўйхати

І. БЎЉИМ

ИШЧИ ДАСТУР



## I. ИШЧИ ДАСТУР

### Кириш

Дастур Ўзбекистон Республикаси Президентининг 2015 йил 12 июндаги “Олий таълим муассасаларининг раҳбар ва педагог кадрларини қайта тайёрлаш ва малакасини ошириш тизимини янада такомиллаштириш чора-тадбирлари тўғрисида” ги ПФ-4732-сон Фармонидаги устувор йўналишлар мазмунидан келиб чиққан ҳолда тузилган бўлиб, у замонавий талаблар асосида қайта тайёрлаш ва малака ошириш жараёнларининг мазмунини такомиллаштириш ҳамда олий таълим муассасалари педагог кадрларининг касбий компетентлигини мунтазам ошириб боришни мақсад қилади. Дастур мазмуни олий таълимнинг норматив-ҳуқуқий асослари ва қонунчилик нормалари, илғор таълим технологиялари ва педагогик маҳорат, таълим жараёнида ахборот-коммуникация технологияларини қўллаш, амалий хорижий тил, тизимли таҳлил ва қарор қабул қилиш асослари, махсус фанлар негизида илмий ва амалий тадқиқотлар, технологик тараққиёт ва ўқув жараёнини ташкил этишнинг замонавий услублари бўйича сўнгги ютуқлар, педагогнинг касбий компетентлиги ва креативлиги, глобал Интернет тармоғи, мультимедиа тизимлари ва масофадан ўқитиш усулларини ўзлаштириш бўйича янги билим, кўникма ва малакаларини шакллантиришни назарда тутди.

Ушбу модулда "Веб дастурлаш" дан барча соҳадаги мутахассислар тайёрлаш учун ўқув стандарти бўлиб ҳисобланади. Дастур ўз ичига фанни эгаллаш бўйича талабларни, HTML, CSS, Java Script, PHP ва замонавий CMS асосларини ҳамда юқоридагилар асосида Web иловаларни ишлаб чиқиш, лойиҳалаш ҳамда ишлаб чиқиш бўйича керакли билимларни олган бўлиб, ривожланган мамлакатларнинг ОТМ тажрибасига таянган ҳолда олиб борилади.

### Модулнинг мақсади ва вазифалари

Веб дастурлаш **модулининг мақсад ва вазифалари:**

- “ Веб дастурлаш ” модулининг мақсади талабаларга замонавий HTML-5, CSS, Java Script, PHP, CMS асосларида Web иловаларни ишлаб чиқиш ва лойиҳалаш алгоритмлар бўйича назарий ва амалий билимларни беришдан иборат.

Фаннинг вазифаси эса талабаларда HTML-5, CSS, Java Script, PHP ва замонавий CMS асосларини ўргатиш ва улар асосида Web иловаларни ишлаб чиқиш, лойиҳалаш кўникмаларини шакллантиришдан иборат.

**Модул бўйича тингловчиларнинг билими, кўникмаси, малакаси ва компетенцияларига қўйиладиган талаблар**

“Веб дастурлаш” модулини ўзлаштириш жараёнида амалга ошириладиган масалалар доирасида:

**Тингловчи:**

- HTML-5, CSS, Java Script, MySQL, PHP ва замонавий CMS асосларини, алгоритмларини, самарадорликлари **ҳақида тасаввурга эга бўлишлари;**
- HTML-5, CSS, Java Script, JQuery, PHP ва замонавий CMS асосларини ва усулларини ҳамда улар асосида Web иловаларни ишлаб чиқиш, лойихалашни **билиши ва улардан фойдалана олишлари;**
- масаланинг қўйилишига қараб янги Web иловаларни ишлаб чиқиш ҳамда ўқув ва ишлаб чиқариш жараёнига тадбиқ **этиш кўникмасига эга бўлишлари лозим.**

**Модулни ташкил этиш ва ўтказиш бўйича тавсиялар**

“Веб дастурлаш ” курси маъруза ва амалий машғулотлар шаклида олиб борилади.

Курсни ўқитиш жараёнида таълимнинг замонавий методлари, педагогик технологиялар ва ахборот-коммуникация технологиялари қўлланилиши назарда тутилган:

- маъруза дарсларида замонавий компьютер технологиялари ёрдамида презентацион ва электрон-дидактик технологиялардан;

- ўтказиладиган амалий машғулотларда техник воситалардан, экспресс-сўровлар, тест сўровлари, ақлий ҳужум, гуруҳли фикрлаш, кичик гуруҳлар билан ишлаш, коллоквиум ўтказиш, ва бошқа интерактив таълим усулларини қўллаш назарда тутилади.

**Модулнинг ўқув режадаги бошқа модуллар билан боғлиқлиги ва узвийлиги**

“Веб дастурлаш” модули мазмуни ўқув режадаги “Дастурий инжиниринг” ва “Операцион тизимлар” ўқув модуллари билан узвий боғланган ҳолда педагогларнинг Веб иловалар яратиш бўйича касбий педагогик тайёргарлик даражасини оширишга хизмат қилади.

**Модулнинг олий таълимдаги ўрни**

Модулни ўзлаштириш орқали тингловчилар Веб иловалар яратишни ўрганиш, амалда қўллаш ва баҳолашга доир касбий компетентликка эга бўладилар.

### Модул бўйича соатлар тақсимоти

№	Модул мавзулари	Тингловчининг ўқув юкламаси, соат				
		Ҳаммаси	Аудитория ўқув юкламаси			Мустақил таълим
			Жами	жумладан		
				Назарий	Амалий машғулот	
1.	Тизимли дастурий таъминотнинг асосий функциялари, тузилиши ва таркибий қисми. Дастурлаш тизими структураси ва тушунчаси. Дастурлаш тизимини келиб чиқиш тарихи.	4	4	2	2	
2.	Замонавий дастурлаш тизими структураси. PHP, MySQL, JavaScript ва CSS имкониятлари. HTML 5. Apache веб сервери. PHP дастурлаш тили. PHP тузилмаси. Изоҳлар. Асосий синтаксис.	4	4	2	2	
3.	Массивлар. Ўзгарувчилар номини ўзлаштириш қоидаси. Операторлар. Шартли оператор. Switch инструкцияси. MySQLнинг асосий тушунаси.	6	4	2	4	
4.	Буйруқлар сатридан MySQL га мурожаат қилиш. MySQL буйруқлари. MySQL маълумотлар базасига PHP дан мурожаат қилиш. JavaScript га кириш. JavaScript ва HTML.	8	6	2	4	2
5.	JavaScript ни файлга жойлаш. JavaScript кодларни отлатка қилиш. HTML5 ва CSS3 ўрганиш. CSS асослари. PHP тилида дастурлаш асослари. MySQL маълумотлар базасида ишлаш. MVC технологиялари	8	6	2	4	2
	<b>Жами:</b>	<b>30</b>	<b>26</b>	<b>10</b>	<b>16</b>	<b>4</b>

## **НАЗАРИЙ МАШҒУЛОТЛАР МАЗМУНИ**

**1 – мавзу. Тизимли дастурий таъминотнинг асосий функциялари, тузилиши ва таркибий қисми. Дастурлаш тизими структураси ва тушунчаси. Дастурлаш тизимини келиб чиқиш тарихи.**

Web иловаларни ишлаб чиқиш асосий тушунчалари. Web-саҳифа, Web-сайт, Web-сервер. Разметкали тиллар: HTML, XML, XHTML, WML. Сценарийли тиллар. "клиент-сервер" технологияси. Web браузерлар ва HTML. Хужжат тузилиши. HTML нинг асосий теглари. Матнларни безаш. Рўйхатлар. Жадваллар. Фреймлар. Графика ва мультимедиа. Формалар. Web саҳифаларга мурожаат. HTML -5 имкониятлари ва эскирган теглар. HTML -5 мультимедиа имкониялари.

**2 – мавзу. Замонавий дастурлаш тизими структураси. PHP, MySQL, JavaScript ва CSS имкониятлари. HTML 5. Apache веб сервери. PHP дастурлаш тили. PHP тузилмаси. Изоҳлар. Асосий синтаксис.**

PHP тилида дастурлаш асослари. MySQL маълумотлар базасида ишлаш. Замонавий CMS лардан фойдаланиш асослари. Мобил тизимлар учун сайтлар қилиш имкониятларини. Мультимедиа технологияларидан фойдаланган холда сайтлар ва порталларни лойихалаш. Сервер веб иловалар ишлаб чиқиш. HTTP протоколини мукамал ўрганиш. Команда сатри утилитлари ёрдамида HTTP билан ишлайдиган инструментлар гуруҳини ўрганиш. Веб серверни созловчи тизимли дастурий таъминотлар.

**3 – мавзу. Массивлар. Ўзгарувчилар номини ўзлаштириш қондаси. Операторлар. Шартли оператор. Switch инструкцияси. MySQLнинг асосий тушунаси.**

PHP да маълумотлар базаси билан ишлаш. PHP да ОЙД га кириш. PHP да колиплар. MySQL маълумотлар базасини бошқаришни такомиллаштириш. Маълумотлар базасини ташкил қилиш ва ундан фойдаланиш. Мобил сайтлар ишлаб чиқиш асослари. Ананавий мобил сайтлар тараққиёти ва фарқлари.

**4 – мавзу. Буйруқлар сатридан MySQL га мурожаат қилиш. MySQL буйруқлари. MySQL маълумотлар базасига PHP дан мурожаат қилиш. JavaScript га кириш. JavaScript ва HTML.**

JavaScript ифодалар, ўзгарувчилар, объектлар. JavaScript объектлар иерархияси. Location объекти. JavaScript да ОЙД асослари. Мультимедиали веб илова ишлаб чиқиш (клиент учун). Ўйин, каркулятор, анимацияли иловалар ишлаб чиқиш. Web саҳифани чоп қилиш. Интернетда ишлайдиган иловалар ишлаб чиқиш (социал сайтлар, почта, излаш тизимлари ).

**5 - мавзу: JavaScript ни файлга жойлаш. Java-Script кодларни отлатка қилиш. HTML5 ва CSS3 ўрганиш. CSS асослари. PHP тилида дастурлаш асослари. MySQL маълумотлар базасида ишлаш. MVC технологиялари**

CSS (Cascading Style Sheets) асослари. Селекторлар ва уларнинг коидалари. Селектор гуруҳлари. Авлод селекторлар (Селекторы потомков). PSD селекторлари ва улардан фойдаланиш. Ссылка селекторлари. Блок элементлари умумий концепциялари. Cross браузерлар.

## **АМАЛИЙ МАШҒУЛОТЛАР МАЗМУНИ**

**1-амалий машғулот. Тизимли дастурий таъминотнинг асосий функциялари, тузилиши ва таркибий қисми. Дастурлаш тизими структураси ва тушунчаси. Дастурлаш тизимини келиб чиқиш тарихи.**

Разметкали тиллар: HTML, XML, XHTML, WML. Сценарийли тиллар. "клиент-сервер" технологияси. Web браузерлар ва HTML. Хужжат тузилиши. HTML нинг асосий теглари. PHP, MySQL, JavaScript ва CSS имкониятлари. Apache веб сервери урнатиш.

**2-амалий машғулот. Замонавий дастурлаш тизими структураси. PHP, MySQL, JavaScript ва CSS имкониятлари. HTML 5. Apache веб сервери. PHP дастурлаш тили. PHP тузилмаси. Изоҳлар. Асосий синтаксис.**

PHP тилида дастурлаш асослари. Мультимедиа технологияларидан фойдаланган холда сайтлар ва порталларни лойихалаш. Сервер веб иловалар ишлаб чиқиш. HTTP протоколини мукамал ўрганиш. Команда сатри утилитлари ёрдамида HTTP билан ишлайдиган инструментлар гуруҳини ўрганиш. Веб серверни созловчи тизимли дастурий таъминотлар.

**3-амалий машғулот. Массивлар. Ўзгарувчилар номини ўзлаштириш  
қоидаси. Операторлар. Шартли оператор. Switch инструкцияси.  
MySQLнинг асосий тушунаси.**

PHP да маълумотлар базаси билан ишлаш. PHP да ОЙД га кириш.  
PHP да қолиплар. MySQL маълумотлар базасини бошқаришни  
такомиллаштириш. Маълумотлар базасини ташкил қилиш ва ундан  
фойдаланиш. Мобил сайтлар ишлаб чиқиш асослари. Ананавий мобил  
сайтлар тараққиёти ва фарқлари.

**4-амалий машғулот. Буйруқлар сатридан MySQL га мурожаат  
қилиш. MySQL буйруқлари. MySQL маълумотлар базасига PHP  
дан мурожаат қилиш. JavaScript га кириш. JavaScript ва HTML.**

JavaScript ифодалар, ўзгарувчилар, объектлар. JavaScript объектлар  
иерархияси. Location объекти. JavaScript да ОЙД асослари. Мультимедиали  
веб илова ишлаб чиқиш (клиент учун). Ўйин, каркульятор, анимацияли  
иловалар ишлаб чиқиш. Web саҳифани чоп қилиш. Интернетда  
ишлайдиган иловалар ишлаб чиқиш (социал сайтлар, почта ,излаш  
тизимлари ).

**5 – амалий машғулот. JavaScript ни файлга жойлаш. Java-Script  
кодларни отлатка қилиш. HTML5 ва CSS3 ўрганиш. CSS асослари.  
PHP тилида дастурлаш асослари. MySQL маълумотлар базасида  
ишлаш. MVC технологиялари**

CSS (Cascading Style Sheets) асослари. Селекторлар ва уларнинг  
қоидалари. Селектор гуруҳлари. Авлод селекторлар(Селекторы потомков).  
PSD селекторлари ва улардан фойдаланиш. Ссылка селекторлари. Блок  
элементлари умумий концепциялари. Cross браузерлар.

## ЎҚИТИШ ШАКЛЛАРИ

Мазкур модул бўйича қуйидаги ўқитиш шаклларида фойдаланилади:

- маърузалар, амалий машғулотлар (маълумотлар ва технологияларни англаб олиш, ақлий қизиқишни ривожлантириш, назарий билимларни мустаҳкамлаш);
- давра суҳбатлари (қўрилаётган лойиҳа ечимлари бўйича таклиф бериш қобилиятини ошириш, эшитиш, идрок қилиш ва мантиқий хулосалар чиқариш);
- баҳс ва мунозаралар (лойиҳалар ечими бўйича далиллар ва асосли аргументларни тақдим қилиш, эшитиш ва муаммолар ечимини топиш қобилиятини ривожлантириш).

## БАҲОЛАШ МЕЗОНИ

№	Баҳолаш турлари	Максимал балл	Баллар
1	Кейс топшириқлари	2.5	1.2 балл
2	Мустақил иш топшириқлари		0.5 балл
3	Амалий топшириқлар		0.8 балл

# II. БЎЛИМ

MODULNI O`QITISHDA  
FOYDALANILADIGAN  
INTERFAOL TA`LIM  
METODLARI



## II. МОДУЛНИ ЎҚИТИШДА ФОЙДАЛАНИЛАДИГАН ИНТЕРФАОЛ ТАЪЛИМ МЕТОДЛАРИ.

### “SWOT-таҳлил” методи.

**Методнинг мақсади:** мавжуд назарий билимлар ва амалий тажрибаларни таҳлил қилиш, таққослаш орқали муаммони ҳал этиш йўлларни топишга, билимларни мустаҳкамлаш, такрорлаш, баҳолашга, мустақил, танқидий фикрлашни, ностандарт тафаккурни шакллантиришга хизмат қилади.



**Намуна:** Веб дастурлашнинг тили имкониятини SWOT таҳлилини ушбу жадвалга туширинг.

<b>S</b>	Веб дастурлашнинг кучли томони	Open source (очиқ кодли), фойдаланувчилар сонининг кўплиги...
<b>W</b>	Веб дастурлашнинг кучсиз томонлари	Ахборот хавфсизлиги муаммолари мавжуд кучсиз, заиф химояланган
<b>O</b>	Веб дастурлашнинг имкониятлари	Beautiful UI (чиройли интерфейс), Connectivity (барча Веб алоқа технологиялари ва Интернет билан боғланиш)...
<b>T</b>	Веб дастурлашнинг тўсиқлар (ташки)	Маълумотлар хавфсизлигининг тўлақонли таъминланмаганлиги...

## Хулосалаш» (Резюме, Веер) методи

**Методнинг мақсади:** Бу метод мураккаб, кўптармоқли, мумкин қадар, муаммоли характеридаги мавзуларни ўрганишга қаратилган. Методнинг моҳияти шундан иборатки, бунда мавзунинг турли тармоқлари бўйича бир хил ахборот берилади ва айни пайтда, уларнинг ҳар бири алоҳида аспектларда муҳокама этилади. Масалан, муаммо ижобий ва салбий томонлари, афзаллик, фазилат ва камчиликлари, фойда ва зарарлари бўйича ўрганилади. Бу интерфаол метод танқидий, таҳлилий, аниқ мантикий фикрлашни муваффақиятли ривожлантиришга ҳамда ўқувчиларнинг мустақил ғоялари, фикрларини ёзма ва оғзаки шаклда тизимли баён этиш, ҳимоя қилишга имконият яратади. “Хулосалаш” методидан маъруза машғулотларида индивидуал ва жуфтликлардаги иш шаклида, амалий ва семинар машғулотларида кичик гуруҳлардаги иш шаклида мавзу юзасидан билимларни мустаҳкамлаш, таҳлили қилиш ва таққослаш мақсадида фойдаланиш мумкин.

### Методни амалга ошириш тартиби:



тренер-ўқитувчи иштирокчиларни 5-6 кишидан иборат кичик гуруҳларга ажратади;



тренинг мақсади, шартлари ва тартиби билан иштирокчиларни таништиргач, ҳар бир гуруҳга умумий муаммони таҳлил қилиниши зарур бўлган қисмлари туширилган тарқатма



ҳар бир гуруҳ ўзига берилган муаммони атрафлича таҳлил қилиб, ўз мулоҳазаларини тавсия этилаётган схема бўйича тарқатмага ёзма баён қилади;



навбатдаги босқичда барча гуруҳлар ўз тақдимотларини ўтказадилар. Шундан сўнг, тренер томонидан таҳлиллар умумлаштирилади, зарурий ахборотлар билан тўлдирилади ва мавзу яқунланади.

### Намуна:

Веб дастурлашнинг					
Android		iOS		Windows Phone	
афзаллиги	камчилиги	афзаллиги	камчилиги	афзаллиги	камчилиги
<b>Хулоса:</b>					

## “Кейс-стади” методи

«Кейс-стади» - англизча сўз бўлиб, («case» – аниқ вазият, ходиса, «stadi» – ўрганмоқ, таҳлил қилмоқ) аниқ вазиятларни ўрганиш, таҳлил қилиш асосида ўқитишни амалга оширишга қаратилган метод ҳисобланади. Мазкур метод дастлаб 1921 йил Гарвард университетиде амалий вазиятлардан иқтисодий бошқарув фанларини ўрганишда фойдаланиш тартибида қўлланилган. Кейсда очик ахборотлардан ёки аниқ воқеа-ходисадан вазият сифатида таҳлил учун фойдаланиш мумкин. Кейс харакатлари ўз ичига қуйидагиларни қамраб олади: Ким (Who), Қачон (When), Қаерда (Where), Нима учун (Why), Қандай/ Қанака (How), Нима-натижа (What).

### “Кейс методи” ни амалга ошириш босқичлари

Иш босқичлари	Фаолият шакли ва мазмуни
<b>1-босқич:</b> Кейс ва унинг ахборот таъминоти билан таништириш	<ul style="list-style-type: none"><li>✓ якка тартибдаги аудио-визуал иш;</li><li>✓ кейс билан танишиш(матнли, аудио ёки медиа шаклда);</li><li>✓ ахборотни умумлаштириш;</li><li>✓ ахборот таҳлили;</li><li>✓ муаммоларни аниқлаш</li></ul>
<b>2-босқич:</b> Кейсни аниқлаштириш ва ўқув топшириғни белгилаш	<ul style="list-style-type: none"><li>✓ индивидуал ва гуруҳда ишлаш;</li><li>✓ муаммоларни долзарблик иерархиясини аниқлаш;</li><li>✓ асосий муаммоли вазиятни белгилаш</li></ul>
<b>3-босқич:</b> Кейсдаги асосий муаммони таҳлил этиш орқали ўқув топшириғининг ечимини излаш, хал этиш йўллариини ишлаб чиқиш	<ul style="list-style-type: none"><li>✓ индивидуал ва гуруҳда ишлаш;</li><li>✓ муқобил ечим йўллариини ишлаб чиқиш;</li><li>✓ ҳар бир ечимнинг имкониятлари ва тўсиқларни таҳлил қилиш;</li><li>✓ муқобил ечимларни танлаш</li></ul>
<b>4-босқич:</b> Кейс ечимини ечимини шакллантириш ва асослаш, тақдимот.	<ul style="list-style-type: none"><li>✓ якка ва гуруҳда ишлаш;</li><li>✓ муқобил вариантларни амалда қўллаш имкониятларини асослаш;</li><li>✓ ижодий-лойиҳа тақдимотини тайёрлаш;</li><li>✓ якуний хулоса ва вазият ечимининг амалий аспектларини ёритиш</li></ul>

**Кейс.** Веб қурилма учун Андроид опреацион тизимининг 5.0 (API Level: 21) версияси учун илова ишлаб чиқилди. Сизнинг телефонингиздаги Андроид опреацион тизимининг версияси 4.3 (API Level: .

Вебиловани телефонингизга ўрнатиб ишга туширмоқчи бўлганингизда хатолик келиб чиқди. Яъни илова ишламади.

### Кейси бажариш босқичлари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- Мобил иловани ишга тушириш учун бажариладагина ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш).

### «ФСМУ» методи

**Технологиянинг мақсади:** Мазкур технология иштирокчилардаги умумий фикрлардан хусусий хулосалар чиқариш, таққослаш, қиёслаш орқали ахборотни ўзлаштириш, хулосалаш, шунингдек, мустақил ижодий фикрлаш кўникмаларини шакллантиришга хизмат қилади. Мазкур технологиядан маъруза машғулотларида, мустақамлашда, ўтилган мавзунини сўрашда, уйга вазифа беришда ҳамда амалий машғулот натижаларини таҳлил этишда фойдаланиш тавсия этилади.

### Технологияни амалга ошириш тартиби:

- қатнашчиларга мавзуга оид бўлган якуний хулоса ёки ғоя таклиф этилади;
- ҳар бир иштирокчига ФСМУ технологиясининг босқичлари ёзилган қоғозларни тарқатилади:

Ф

• фикрингизни баён этинг

С

• фикрингизни баёнига сабаб кўрсатинг

М

• кўрсатган сабабингизни исботлаб мисол келтиринг

У

• фикрингизни умумлаштиринг

- иштирокчиларнинг муносабатлари индивидуал ёки гуруҳий тартибда тақдимот қилинади.

ФСМУ таҳлили қатнашчиларда касбий-назарий билимларни амалий машқлар ва мавжуд тажрибалар асосида тезроқ ва муваффақиятли

ўзлаштирилишига асос бўлади.

**Намуна.**

**Фикр:** “Полимарфизим объектга йўналтирилган дастурлашнинг асосий тамойилларидан биридир”.

**Топшириқ:** Мазкур фикрга нисбатан муносабатингизни ФСМУ орқали таҳлил қилинг.

Ш. БЎЛИМ

НАЗАРИЙ  
МАТЕРИАЛЛАР

### III. НАЗАРИЙ МАТЕРИАЛЛАР

**1-мавзу: Тизимли дастурий таъминотнинг асосий функциялари, тузилиши ва таркибий қисми. Дастурлаш тизими структураси ва тушунчаси. Дастур-лаш тизимини келиб чиқиш тарихи.**

**Режа:**

1. Веб дастурлашга кириш.
2. Процедура «сўров — жавоб».
3. PHP, MySQL, JavaScript ва CSS имкониятлари.
4. HTML 5.
5. Apache веб сервери.

**Таянч иборалар:** html, http, https, url, php, css, mysql.

#### 1.1. Веб дастурлашга кириш

Бутун дунё ўргимчак тўри – бу узлуксиз ривожланаётган тўр бўлиб, дастлабки 1990 йиллардаги концепцияларидан аллақачон узоқлашган бўлиб, олдин у аниқ бир масалани ечишга қаратилган эди.

Европа юқори энергиялар физикаси марказидаги юқори технологик тадқиқотлар қатнашаётган бутун дунёдан йиғилган олимлар орасида тарқатиш учун мумкин бўлмаган даражада катта хажмдаги маълумотларни тақдим этди.

Бу пайтда Интернет мавжуд эди ва унга 100 минглаб компьютерлар боғланган эди, шунинг учун **Тим Бернерс-Ли** гипер мурожаат орқали улар орасида навигацияни ишлаб чиқди, кейинчалик бунинг гипер мурожаатларни узатиш протоколлари ёки (Hyper Text Transfer Protocol (HTTP)) деб аташди )

У махсус разметка тили ишлаб чиқди бу тил гипер текстли разметка тили ёки (Hyper Text Markup Language (HTML)) деб аташди. Бу тушунчаларни умумлаштириш учун биринчи браузер ва веб браузер ишлаб чиқди.

Ўз даври учун бу концепция революцион характерга эга эди. Бунгача

асосан боғланишлар модемлар билан амалга оширилган, бу асосан алохида компьютерлар бир-бири билан махсус хизматлар орқали амалга оширилган.

Албатта касбдошлар билан самарали электрон мулоқот учун жуда кўплаб электрон доскаларга аъзо бўлиш керак бўларди.

Бернерс-Ли битта харакат билан бундай мужмал ишларни йукқа чиқарди, 1990 йилда бир-бири билан рақобатлашувчи учта графикали браузер мавжуд бўлиб ва фойдаланувчилар 5 млн етди.

Аммо маълум бўлдики баъзи камчиликларга йўл қўйилган.

Албатта матнли ва графикли саҳифалар гипер мурожаат орқали бир-бирига мурожаат қиладиган концепция жуда яхши эди, аммо натижада ўша пайтдаги компьютерлар потенциалини, Интернет эса ўсиб келаётган фойдаланувчилар талабларини динамик ўзгариш контекстини ифодалай олмас эди.

Хатто фавқат матн разметкалари ва GIF-расмлар билан ҳам бутун дунё ўргимчак тури хайратомуз таъсурот қолдирган эди.

## **1.2.Процедура «Суров — жавоб»**

Умумий холда «Суров — жавоб» процедураси қуйидагилардан ташкил топади. Браузер сўрови веб браузерга, веб браузер эса веб саҳифага юборади, шундан кейин браузер саҳифани ифодалаш билан шуғулланади (расм. 1.1).

Бунда амаллар кетма кетлиги қуйидагича бўлади:

1. Браузер адрес сатрига қуйидагилар киритилади <http://server.com>.
2. Сизнинг браузер IP-адресни излайди, [server.com](http://server.com). номли доменга мос бўлган.
3. [server.com](http://server.com) нинг бош саҳифасига браузер сўров юборади
4. Сўров интернетдан орқали [server.com](http://server.com) нинг веб серверига келади
5. Сўров ни қабул қилган веб сервер оз қаттик дискидан веб саҳифани излайди
6. Сервер веб саҳифани ишлайди ва қайта маршрути бўйича браузер адресига жўнатади



7. Браузер веб-саҳифани ифодалайди (тасвирлайди).

### **1.3. PHP, MySQL, JavaScript ва CSS имкониятлари**

Маъруза бошида Web 1.0 технологиялар олами ифодаланди, аммо Web 1.1 технологияларини яратиш учун жуда катта сакраш бўлди шу билан бирга бир қатор браузер кенгайтмалар пайдо бўлди, масалан Java, JavaScript, Jscript ва ActiveX.

Сервер томондан жараён шлюз интерфейс (Common Gateway Interface (CGI)) билан таъминланган, бунда сценариялар ёзиш тили сифатида Perl, PHP фойдаланилган ва сценарияларни сервер томонидан бажарилиши битта файлни иккинчи файлга динамик жойлаштиришдан иборат бўлади.

Қачон холат бутунлай ойдинлашса, олдинги холатларда учта асосий технология келтирилди

Сценариялар Perl тили мураккаб бўлса ҳам, баъзи қарама-қаршилиқларга бўлса ҳам оммалашди, содда PHP унинг MySQL маълумотлар базаси билан ишлаш имконияти фойдаланувчилар орасида унга икки марта кўп фойдаланувчилар эга бўлиш имконини берди.

JavaScript эса CSS ва HTML нинг муҳим боғловчи қсми бўлиб хизмат қилади. Бунда AJAX мижоз томондан бажариладиган жараённи ифодалайди.

AJAX технологиялари ёрдамида веб саҳифалар маълумотларни қайта ишлайди ва веб серверга сўров юборади ва бу жараёндан фойдаланувчининг хабари бўлмайди.

Агар сиз JavaScript ва CSS ни бирлаштирсангиз сизда юқори даражадаги динамик ва инрерактив сайт яратиш рецепти пайдо бўлади.

## **PHP**

Кўпгина бошқа дастурлаш тилларидан фарқли равишда, PHP қандайдир ташкилот ёки кучли дастурчи томонидан яратилган эмас. Уни оддий фойдаланувчи Расмус Лердорф 1994 йили ўзининг бош саҳифасини

интерактив услубда кўрсатиш учун яратган. Унга Personal Home Page (PHP – шахсий бош саҳифа) деб ном берган.

1995 йили Расмус РНРни ўзининг HTML формалари билан ишлайдиган бошқа дастур билан умумлаштириб RHP/FI Version 2 (“Form Interpretator“) ҳосил қилди. 1997 йилга бориб РНР дан фойдаланувчи сайтлар 50 мингдан ошди. Шундан сўнг веб технология усталари РНР ғояси асосида мукамал тил яратишга Зива Сураски ва Энди Гутманс асосчилигида киришилди. РНРни самарали деб ҳисобланмагани учун деярли нолдан бошлаб, мавжуд С ва Перл тилларидан ибрат олиб РНР3 талқинини яратилди. 1999 йилга келиб РНР асосида қурилган сайтлар миллиондан ошиб кетди. 2000 йилда эса Zend Technologies ширкати янги кўпгина функцияларни қўшган ҳолда РНР4 шарҳловчисини яратди.

РНР дан фойдаланиш веб саҳифага динамик хусусиятлар бериш имкониятини тубдан ўзгартиради.

Сервер томонидан бажариладиган иловалар Бундай иловалардан фойдаланиш учун РНР-парсер (яъни php-скриптларни қайта ишловчи) ва скриптларни қайта ишловчи web-сервер, скриптларни натижасини кўриш учун браузер ва албатта php-кодини ёзиш учун матн муҳаррири керак бўлади.

Клиент томонидан бажариладиган GUI-иловалар РНР тили ёрдамида бирор-бир компьютерда браузер ва web-серверлардан мустақил равишда ўзи бажариладиган скриптларни ҳам яратиш мумкин.

Буйруқлар сатрида бажариладиган скриптлар. Бу соҳа РНР тилини энди-гина ўрганаётган фойдаланувчилар учун унча муҳим бўлмаган соҳадир. РНР тилини бу соҳага қўллаш учун php кенгайтмали махсус ёрдамчи – РНР-GTK талаб этилади.

Қачонки саҳифага РНР кенгайтма берсак унда тўғридан-тўғри сценария тилига мурожаат қилиш имконияти бўлади. Дастурчи қуйидагича код ёзиши керак бўлади :

```
<?php  
echo " Today is " . date("l") . ". ";
```

?>

Here's the latest news.

Очувчи тег **<?php** веб серверга **?>** теггача интерпретация қилиш имконини беради.

Бу конструкции доирасидаги барча ёзувлар миждога оддий HTML кўринишида жўнатилади

Шунинг учун Here's the latest news матни браузерда кўрилади

PHP-тегнинг ичидаги date функцияси мос тизим серверининг жорий хафта кунини ифодалайди.

Натижада бу иккита қисмдан иборат қуйидаги маълумотларни чиқаради:

Today is Wednesday. Here's the latest news.

PHP етарлича осон ва келишувчан тил хисобланади баъзи дастурчилар PHP конструкцияларга халақит қилишмоқчи бўлишади, масалан, қуйидагича:

Today is <?php echo date("l"); ?>. Here's the latest news.

Бунда қуйидагини билиш муҳим PHP дан фойдаланаётган веб дастурчи сценариялар тилини олади, бунда код тез ишламайди, масалан, C га солиштирсак, аммо шундай булса ҳам тез ишлайди ва HTML разметкага мос тушади.

## MySQL

Дастлабки HTML коднинг динамик ўзгаришини тўла маънода англаш учун фойдаланувчининг сизнинг сайтингизда ишлаш давомида қилган ўзгаришларини кузатиш керак бўлади.

Бутун дунё ўргимақ тўри яратилаётган пайтдаги кўпгина сайтларда парол ва логин маълумотларни сақлаш учун структуралашмаган матнлардан фойдаланишган. Масалага бундай ёндашув, бир вақтда бир қанча фойдаланувчилар мурожаат қилса ва файл ишончли химояланмаган бўлса, бир қатор муаммоларни келтириб чиқаради.

Структуралашмаган файл шундай даражада хажми кўпайиб кетиши мумкинки натижада бундай хажмдаги файл билан ишлаш мумкин бўлмай қолади, бундай хажмдаги файлларни бирлаштириш, мураккаб излаш

амаллари , тезлик масалалари хақида гапиришга хожат қолмайди.

Худди шундай холатлардаги мураккабликни хал қилиш учун реляцион маълумотлар базаси ва структуралашган сўровлар тизими ишлаб чиқилган эди. Худди шунингдек MySQL ҳам юқорида айтилганлар қаторига кирадиган мукамал текин ва кўпгина веб-серверларга ўрнатиган маълумотлар базасини бошқариш тизимидир. Бу тизим маълумотлар базасини ишончли ва тез бошқариш имконини беради, тизим командалари содда инглиз сўзларидан иборатдир.

MySQL нинг юқори даражаси структура хисобланади, ичида бир қатор жадваллардан иборат, бу жадвалларда маълумотлар сақланади.

Фараз қилайлик, сиз users (фойдаланувчилар) жадвали билан ишляяпсиз, унда графлар хосил қилинган фамилия – surname, исм - name, электрон адрес – email, худди шу тартибда кейинги фойдаланувчиларнинг маълумотларини киритиш мумкин.

Қуйида битта командани келтириб ўтамыз:

```
INSERT INTO users VALUES('Smith', 'John', 'jsmith@mysite.com');
```

Олдин эслатиб ўтганимиз каби маълумотлар базасини ва жадвални хосил қилиш учун бир қанча ишларни амалга ошириш керак бўларди, бу ерда эса INSERT командасининг соддалиги ва маълумотлар базасига янги маълумот кўшишнинг осонлиги келтириб ўтилган.

INSERT буйруғи структурали сўровлар тили (Structured Query Language (SQL)) га мисол, бу тил 1970 йилларда ишлаб чиқилган эски дастурлаш тили COBOL ни эслатади.

Бу тил сўровлар ёзиш учун қулай бўлганлиги учун узоқ вақт давомида маълумотлар базасига сўровлар ёзиш учун ишлатилиб келиняпти.

Бунда қидириш масалалари хам жуда осон хал қилинган. Фараз қилайлик электрон адрес мавжуд бўлса шу адрес эгасининг исм ва фамилиясини топиш масаласи қўйилган бўлсин бунинг учун қуйидагича MySQL сўров ёзилади:

```
SELECT surname, firstname FROM users WHERE
```

email='jsmith@mysite.com';

Бундан сўнг MySQL бизга Smith, John маълумотни чиқариб беради, худди шундай давом этишимиз мумкин.

MySQL имкониятлари кенг ва командалари содда бўлиб содда командалар комбинациясидан жуда кўплаб сўровлар ёзиш имконини беради.

## JavaScript

JavaScript - HTML-хужжатнинг барча элементларига сценариялар орқали мурожаат қилиш учун ишлаб чиқилган.

Бошқача айтсак, у фойдаланувчи билан динамик мулоқат қилиш воситасига эга, масалан, электрон адрес ни киритаётганда унинг тўғри ёки нотўғрилигини, формага мос ёки мос эмаслигини текширади ва баъзи маслаҳатлар таклиф қилади.

CSS ва JavaScript имкониятлари асосан динамик веб саҳифаларни ифодалашда намоён бўлади, бошқа саҳифаларга солиштирганда фарқи яққол кўринади.

JavaScript билан ишлаганда баъзи қийинчиликлар пайдо бўлиши мумкин, бунда хар хил браузерлар билан ишлаганда баъзи қўшимча ўзгартиришлар киритишимизга тўғри келиши мумкин.

Асосан фарқлар баъзи браузерлар қўшимча функциялар, имкониятлар қўшишса мавжуд дастурий таъминотлар билан биргаликда ишлашда муаммолар туғилади.

JavaScript-кодни барча браузерлар билан ишлайдиган оддий кўринишини қараймиз:

```
<script type="text/javascript">  
  document.write("Today is " + Date() );  
</script>
```

Бу дастур қисми браузерга script тегнинг ичидаги барчасини интерпретация қилишга юборади, код сифатида JavaScript, ундан кейин браузер ва мос равишда PHP, MySQL, JavaScript ва CSS, жорий хужжатга

«Today is» ва сана ва натижада қуйидагича маълумотлар чиқади

Today is Sun Jan 01 2017 01:23:45

## CSS

CSS нинг 3 стандарти ишлаб чиқилгандан кейин, динамик интерактивликни даражаси кўтарилди, олдин бу фақат JavaScript хисобтга амалга оширилди.

Масалан, энди сиз нафақат HTML ихтиёрий элементига стил беришингиз, унинг размерини, рангини, чегараларини, интервалларини, ўзгартиришингиз бир неча қатор CSS код ёзиш ва уни веб саҳифага қўшиш кифоя қилади. CSS татбиқ қилиш `<style>` и `</style>` ни веб саҳифа сатр бошига қўшиб қўйиш керак бўлади.

:

```
<style>
```

```
  p {
```

```
    text-align:justify;
```

```
    font-family:Helvetica;
```

```
  }
```

```
</style>
```

Бу қоидалар `<p>` тегининг текислаш қоидасини ўзгартиради, бунда абзас тўла текисланади ва унда Helvetica шрифти ишлатилади

## 1.4. HTML 5

Веб – стандарт ларга киритилган қўшимча ўзгартиришлар ҳар қандай шов-шувли бўлмасин бу дастурчилар учун бу етарли бўлмай кўринаверади.

Масалан, халигача графика билан ишлашнинг қўшимча модулларга (Flash) мурожаат қилмасдан ишлайдиган содда усули ишлаб чиқилмаган.

Худди шунингдек аудио ва видео материалларга нисбатан ҳам шу гапни айтиш мумкин.

Бундан ташқари HTML ривожланиш даврида бир қатор камчиликлар қараш мумкин.

Бундай камчиликлар Интернет Web 2.0 технологиясига ўтиш даврида янги HTML стандарт ишлаб чиқилди ва шу аснода бу камчиликлар йўқ қилинди. Бу стандартга HTML5 деб ном берилди бу стандарт ишлаб чиқилиши узоқ 2004 йилга бориб тақалади, қачон дастлабки лойихалар Mozilla Foundation ва Opera Software браузерлар бўлган эди. Аммо унда якуний тугалланган ҳолатда World Wide Web Consortium (W3C) 2013 йилни қайд этиш керак бўлади. Бу стандартни ишлаб чиқиш 9 йил давом этди, аммо бу даврда бу стандартдан тўла бўлмаса ҳам фойдаланишди. Бу даврда қанча сайтлар пайдо бўлди ва тез суратларда ривожланди, баъзилари умуман йўқ бўлиб кетди, аммо таянч дастурий таъминот шошилмасдан керак бўлса секин ва аниқ текширилган қоидалар асосида ишлаб чиқилади . Шунинг учун ҳам HTML5 ҳақидаги тугалланган маълумотлар ёки китоблар 2014 йилнинг охирида пайдо бўлди. Нима учун шундай? Чунки 2015 йилда кейинги версия HTML 5.1 устида иш олиб борила бошланди. Худди шундай чексиз цикл кўринишида стандарт устида иш олиб борилади.

HTML5.1 таркибига жуда кўплаб янгиликлар киритилган аммо асосий HTML 5 қоидалари кўп йиллар давомида ўзгармай қолаверади.

Бу янги ўзгаришлар олдинги асосий қоидаларнинг мантиқини инкор қилмайди балки унга баъзи ўзгариш ва сайсал бериши мумкин.

Бу стандарт HTML га жуда катта ўзгаришлар киритди агар қисқача гапирадиган бўлсак қуйидагиларни санаб ўтишимиз мумкин бўлади:

**Разметка** ларга янги элементлар киритди `<nav>` ва `<footer>`, ва энди қуйидаги тегларни ишлатишга маслаҳат бермайди `<font>` ва `<center>`;

**Янги API.** Масалан, янги элемент `<canvas>` (холст) график холстга ёзиш ва чизиб учун, `<audio>` ва `<video>` элемент автоном веб-илова учун , микро маълумотлар ва локал маълумотлар сақлаш( хранилище);

**Иловалар .** Иккита янги технологиялар ифодаланган: MathML (Math Markup Language — математик разметкалар тили) математик формулаларни экранга чиқариш учун) ва SVG (Scalable Vector Graphics — векторли графикани масштаблаш учун) янги `<canvas>`. элементида график

элементларни ифодалаш учун, бу иккита янги элементни тўла тавсифлаш учун жуда катта хажмдаги маълумотлар керак бўлади, бу иккита MathML ва SVG элемент махсус тушунча учун алоҳида китоблар ёзилган бошқа тўхталиб ўтirmаймиз

### **1.5. Веб сервер Apache**

PHP, MySQL, JavaScript, CSS ва HTML5 ларга қўшимча қилиб динамик веб технологияларга 6 – технологияни веб-серверларни келтиришимиз мумкин. Биз қуйида Apache веб-сервер хақида тўхталамиз.

Биз юқорида веб сервер мижоз ва сервер HTTP протоколи орқали маълумотлар алмашиш жараёнида нима иш қилиши билан қисқача танишиб ўтдик, аммо у биз билганимиздан кўра кўпроқ ишларни бажарар экан.

Масалан Apache нафақат HTML- файлларга балки бошқа файллар билан ҳам ишлайди, булар қаторига Flash-роликлар, MP3 аудио файл форматдаги, RSS-оқимли файллар (Really Simple Syndication) —ларни киритишимиз мумкин.

Веб клиент томонидан HTML-саҳифада изланган хар бир элемент учун, сервердан бажариши керак бўлган хизматни сўрайди.

Аммо бу объектлар статик файл бўлиши мумкин эмас худди GIF-форматдаги каби.

Улар хаммаси генерация қилинган бўлиши керак худди PHP сценариялари каби. Булар қуйидагилар бўлиши мумкин: PHP тасвир хосил қилиши мумкин, файл хосил қилиши мумкин, ҳисоб китобларни кейинги қилиши мумкин. Бунинг учун одатда, олдиндан Apache ёки PHP да компиляция қилинган ёки дастур бажарилиш давомида чақириладиган модул мавжуд бўлади. Худди шундай модуллардан бири GD (Graphics Draw — рисование графики) кутубхонаси айтиш мумкин, бундан PHP график элементларни хосил қилиш ва қайта ишлаш учун фойдаланади.

Apache жуда кўплаб ўз модулларига эга.

PHP модулларига қўшимча қилиб шуни айтиш мумкин хавфсизни таъминлаш учун ишлатиладиган модуллар.



Модуль Rewrite URL-адресларни веб серверга қайта ишлаш ва ички тартиблари ва шартлари билан ёзиб олиш имконини беради.

Тез тез мурожаат қилинадиган саҳифаларни кешдан орқали мурожаат қилиш ҳисобига серверга бўлаётган суровлар сонини камайтиради.

### **Назорат саволлари**

1. Тўла динамик сайтлар яратиш учун зарур бўладиган 4 та компонентани санаб ўтинг?
2. HTML аббревиатура нимани англатади ?
3. Нима учун MySQL номида SQL ном қатнашган ?
4. PHP ва JavaScript веб саҳифа динамик таркибини генерация қилиш учун фойдаланиладиган дастурлаш тили ҳисобланади. Уларнинг асосий фарқлари нималардан иборат ва нима учун сиз бу иккаласидан фойдаланасиз ?
5. CSS аббревиатурасининг маъноси нимадан иборат?
6. HTML5 нинг асосий 3 та элементини санаб ўтинг?.
7. Очиқ кодли дастурий маҳсулотда хатоликларни билиб қолдингиз, қандай қилиб тўғриланган версияни олиш мумкин ?

### **Фойдаланилган адабиётлар:**

1. VG Olifer, N. A.Olifer. Computer networks. Principles, technologies, protocols. St. Petersburg: Publishing House. "Peter," in 2000.
2. R. Darnell, "JavaScript: Handbook." - St. Petersburg: Publishing House. "Peter," in 1998.
3. Duvanov A.A. WEB. Construction of HTML. - St. Petersburg: BHV, 2003.
4. R. Eisenmenger. HTML 3.2/4.0. Handbook. - M: Univ. "Bean", 1998.
5. Stephen Speynaur, Valerie Kuersia. Directory WEB-master. - Ed. BHV
6. Barrett D. JavaScript. Web-professionals. - Kiev: BHV - Kyiv, 2001.
7. Brandenbau D. JavaScript: a collection of recipes. - St. Petersburg.: Peter, 2000.

**2- мавзу: Замонавий дастурлаш тизими структураси. PHP, MySQL, JavaScript ва CSS имкониятлари. HTML 5. Apache веб сервери. PHP дастурлаш тили. PHP тузилмаси. Изоҳлар. Асосий синтаксис.**

### **Режа:**

1. HTML га PHP ни киритиш.

## 2. PHP тузилмаси.

**Таянч иборалар:** *PHP, массив, ўзгарувчилар, операторлар, Switch, цикллр.*

### 2.1. HTML га PHP ни киритиш

Веб-саҳифани яратиш жараёнида ўзи билан PHP, HTML, JavaScript, комбинациясини, MySQL инструкциясини ва CSS ёрдамида форматлашни ва эҳтимол, HTML5 турли элементларидан фойдаланишни тақдим этади.

Бундан ташқари, ҳар бир саҳифа фойдаланувчиларга бошқа саҳифаларни тақдим этиши мумкин манзилни(ссылка) босиш ва формани ўлдириш орқали.

Бу тилларни ўрганишда сиз бундай қийин ҳолатларнинг олдини олишингиз ҳам мумкин. Ушбу босқичда, сиз PHP-кодини ёзишга ва чиқувчи маълумотнинг қандай натижа беришига ёки камида маълумотни таҳлил қилиб тушуниш учун фикрни бир жойга жамлашингиз керак.

PHP ни HTML га жойлаштириш (HTML га PHP ни киритиш)

Одатда, PHP-ҳужжатлар номлари охирида PHP кенгайтмаси қўйилади. Қачонки Web сервер талаб қилинган файлда кенгайтмани учратганда, у автоматик равишда PHP-процессорга файлни юборади.

Буни текшириш учун, сиз, ҳар қандай HTML-ҳужжатни олиб, масалан, index.html файлни олиб index.php номи остида уни сақлаанг, ва у чиқувчи файл билан бир хил тарзда кўрсатилади.

PHP буйруқларини ишга тушириш учун янги тегни ўрганишимиз керак бўлади. Қуйидагича унинг очилиш қисми ҳисобланади:

`<?php` - биринчи бўлиб теги кўз ташланади (тугалланмаган қисми).

Бу бутун PHP код фрагментлари тег ичида жойлаштирилиши мумкин, деб ифодаланган. Булар ёпувчи тег қисми дуч келганда яқунланади `:?>` Унча катта бўлмаган PHP-дастури Hello World, 2.1 расмда кўрсатилган каби бўлиши мумкин.

Мисол 2.1 PHP ни чақириш

```
<?php
echo "Hello world";
?>
```

PHP тузилиши (структураси)

Изоҳлар

PHP кодига изоҳлар киритишнинг икки йўли бор. Биринчидан, сатр бошига жойлаштирувни олдиндан олдини олувчи икки чизик слеш. Бир қатордаги сатрларни изоҳга айлантиради:

```
// Это комментарий Пример Многострочный комментарий
<?php
/* Это область многострочного комментария,
   которая не будет подвергаться интерпретации */
?>
```

### **Асосий синтаксис**

PHP – илдизи билан C ва Perl каби жуда оддий тил, лекин кўпроқ Javaга ўхшаш. Бу жуда мослашувчан, лекин баъзи ўрганиш керак бўлган қоидалар мавжуд синтаксиси ва тузилишига боғлиқ.

Нукта-вергул

Олдинги мисолларда, сиз PHP ни нукта-вергул билан якунланишини кўришингиз мумкин.

```
$x += 10;
```

PHP билан ишлаётган баъзи хатоларнинг сабаби, бу унутувчанликдир. Агар нукта-вергул қўйилмаса, PHP у вазиятни тушуна олмайди ва бир хатони бир вақтнинг ўзида бир неча кўрсатмалар сифатида кўриб чиқиши керак бўлади, бунинг натижасида у жараёни тахлил қила олмайди ва синтаксическ тўплам хатосини чиқаради — Parse error.

Символ \$ (белги)

PHP да \$ белги барча ўзгарувчилар номлари олдидан жойлаштирилган бўлиши керак. Бу PHP-тахлил қилиш дастури дархол ўзгарувчилар билан боғлиқ ишни тез ишлашини таъминлашидан иборат. Ўзгарувчилар қандай тоифага боғлиқ бўлишидан қатъий назар – сатр ёки массивми, сонми улар масалада келтирилгани каби бўлиши керак.

Масала 2.2. Учта хар хил ўзгарувчиларни қўллаймиз

```
<?php
$mycounter = 1;
$mystring = "Hello";
$myarray = array("One", "Two", "Three");
?> Мисол 2.2. Сизнинг биринчи PHP прогаммангиз
<?php // test1.php
$username = "Fred Smith";
echo $username;
echo "<br>";
$current_user = $username;
echo $current_user;
?>
```

Энди дастурни браузерингиз манзил сатрига қуйидагини ёзиб ишга туширишингиз мумкин:

```
http://localhost/test1.php
```

## 2.2.PHP тузилмаси.

### Массивлар

Массивларни гугурт кутиси кўриниши ўрнига қўшилган кўринишида жойлаштириш мумкин. Масалан, бизга битта команда футболчилар исмларини \$team массивида сақлаш керак

```
$team = array('Bill', 'Mary', 'Mike', 'Chris', 'Anne');
```

Агар 4 сони футболчиси кимлигини топиш керак бўлса, сиз қуйидаги буйруқдан фойдаланишингиз мумкин:

```
echo $team[3]; // бу Chris ни кўрсатади
```

### Икки ўлчамли массив

Икки ўлчамли массивга мисол келтириш учун, тасаввур қиламиз сиз маълумотлар структураси керак бўладиган, тўққиз катакдан иборат, 3 X 3 га гуруҳланган бир майдонда крестик-нолик ўйин қадамларини кузатиб боришингиз керак.

Мисол 2.3. Икки ўлчамли массивни аниқлаш

```
<?php
```

```
$oXo = array(array('x', ' ', 'o'), array('o', 'o', 'x'), array('x', 'o', ' '));
```

```
?>
```

Ўзгарувчилар номини ўзлаштириш қоидаси

PHP –ўзгарувчиларни эълон қилишда 4 қоидага амал қилиш керак

- Ўзгарувчилар номлари харф ёки таг чизик \_ (пастки чизик) билан бошланиши керак.

- Ўзгарувчилар исмлари фақат белги бўлиши мумкин: a-zA-Z, 0-9 ва \_ (пастки чизик).

- ўзгарувчилар номларида бўш жой (пробел) бўлиши мумкин эмас. Агар, бирнеча сўз хосил қилиш керак бўлса ажратувчидан (пастки чизик) фойдаланинг, (масалан, \$user\_name).

- ажратувчидан (пастки чизик) фойдаланинг, (масалан, \$user\_name).

- ўзгарувчи номлари харф регистрини хисобга олади. Ўзгарувчи \$High\_score бир ўзгарувчи \$high\_score дан фарқланади.

### Операторлари

PHP кўп кучли операторларга эга, арифметикадан, мантиқий ва сатрли то ўзлаштирувчи операторларгача, таққослаш ва кўплаб бошқа операторларга

(2.1жадвалга қаранг.).

2.1жадвал

Оператор	Тавсиф	Мисол
Арифметик	Элементар математика	$\$a+\$b$
Массивлар билан ишлаш учун	Массивларни кўшиш	$\$a+\$b$
Ўзлаштириш	Қийматни ўзлаштириш	$\$a=23+\$b$
Разрядли амаллар	Файлда Битларни бошқариш	$12^9$
Таққослаш	Иккита қийматни тоққослаш	$\$a<\$b$
Бажариш	Тескари кавсда ёзилган амални бажарилиши	' s-a '
Инкремент  декремент	Битта ошириш ёки камайтириш	$\$a++$
Мантиқий амал	Бул таққослашни бажариш	$\$a$ and $\$b$
Сатрли устида амал	Сатрни бирлаштириш	$\$a.\$b$

Хар хил турдаги операторлари турли миқдордаги операндларни ўзлаштира олади.

- Унар операторлари, масалан, ошириш оператори ( $\$ с ++$ ) сифатида, ёки рақам белгисини ўзгартириш ( $- \$ a$ ) фақат битта операндни олади.

- Бинарй операторлари кўп PHP операторлари томонидан тақдим этиладиган, бўлиш ва кўпайтиришлар ва ўқиш операторларидан ташқари.

- Битта учта компонентали оператор  $x ? y :z$  кўринишга эга. Аслида, бу уч қисмдан иборат битта сатрли `if`, икки сўзлар ўртасида танловни ифодалайди.

### Операторлар бажарилиши устуворлиги

Агар барча операторларда битта ва фақат шу устуворлик бўлса эди, улар интерпретаторга дуч келган бир хил тартибда қайта ишланар эди.

Жадвал 2.2 да PHP операторлари устуворлигига караб энг каттасидан энг паст даражагача рўйхати келтирилган.

2.2-жадвал.

Устувор даражасида жойлашган PHP операторлари (юқоридан пастгача)

Операторлар	Турлар
<code>()</code>	Кавслар
<code>++ --</code>	Инкремент декремент

!	Мантикий амал
+ / *	Арифметик
+ -	Арифметик ва сатрли
<<>>	Битли амаллар
< <= > >= <>	Таққослаш
== != === !==	Таққослаш
&	Разрядлар устида амаллар
^	Разрядлар устида амаллар
	Разрядлар устида амаллар
&&	Мантикий
	Мантикий
?:	Тернар амал
= += -= *= ./= .= %= &= != ^= <<= >>=	Ўзлаштириш

### Операторлар ўзаро муносабати

Биз сўзлар чапдан ўнгга қайта ишлашни кўриб чикдик, операторлар устуворлигига қараб баъзи холлар бундан мустасно. Бироқ, бир неча операторлар ҳам ўнгдан чапга ишлашни талаб қилиши мумкин. Операторлар ўзаромуносабати шарти асосида қайта ишлашни йўналтириш мумкин. Алохида операторлар учун ўзаромуносабат мавжуд емас. Ўзаромуносабат қачонки сиз устуворликни ўзгартирмасангиз ана шунда ўзаромуносабат катта ахамиятга эга. Бунинг учун, сиз операторлар харакатлари хақида билишингиз керак бўлган.

Жадвал. 2.3

Рўйхат операторлари ва уларнинг ички ўзаро боғланишли

Оператор	Тавсиф	Ўзаро боғлиқлиги
CLONE NEW	Янги объектни яратиш	-
< <= > >= <> == != === !==	Таққослаш	-
!	Мантикий инкор	Ўнг
~	Разрядли инкор	Ўнг
++ --	Инкремент декремент	Ўнг
(Int)	Бутун сонга айлантириш	Ўнг
(double) (float) (real)	Хақикий сонга айлантириш	Ўнг
(string)	Сатрга айлантириш	Ўнг
(array)	Массивга айлантириш	Ўнг
(object)	Объектга айлантириш	Ўнг
= += -= *= ./= .= %= &= != ^= <<= >>=	Ўзлаштириш	Ўнг
+	Қўшиш	Чап

-	Айириш	Чап
*	Кўпайтириш	Чап
/	Бўлиш	Чап
%	Модулли бўлиш	Чап
<< >> & ^	Разрядли амаллар	Чап
&&    and or xor	Мантиқий амаллар	Чап
,	Ажратиш амали	Чап

### Тенглик операторлари

Тенглик оператори == (жуфт тенг белгиси) билан биз аллақачон бу бобда учрашдик. Буни (=)ўзлаштириш оператори билан адаштириш мумкин эмас. Мисол 2.4 да биринчи оператор ўзлаштириб олади, иккинчиси эса тенглик учун текшириб кўради.

Мисол 2.4. Ўзлаштириш ва тенглик учун уни текшириш

```
<?php
```

```
$month = "Март";
```

```
if ($month == "Март") echo "Весна наступила";
```

```
?>
```

### Таққослаш Операторлар

Таққослаш операторларидан фойдаланган холда текшириш гуруҳини кенгайтириш мумкин, тенглик ёки тенгсизликда шартларида уни чегаралаб қўймасдан. PHP бунинг учун сизга > (кўп), < (кам), >= (кўп ёки тенг) ва <= (кам ёки тенг) операторларини тақдим этади.

Мисол 2.5. Тўртта таққослаш оператори

```
<?php
```

```
$a = 2; $b = 3;
```

```
if ($a > $b) echo "$a больше $b<br>";
```

```
if ($a < $b) echo "$a меньше $b<br>";
```

```
if ($a >= $b) echo "$a больше или равно $b<br>";
```

```
if ($a <= $b) echo "$a меньше или равно $b<br>";
```

```
?>
```

### Шартли оператор

Кетма-кетлик асосидаги шарт асосида бир неча харакатларни амалга ошириш керак. Керакли натижага эришишда elseif шартларини ишлатиш мумкин. Бу else шартига ўхшаш фақат шарт кодига яна шарт ибора қўйилади.

Бу бир неча харакатларни амалга ошириш учун зарур шартни бир бўлади. Баёнотда, агар бошқа фойдаланиб, сиз керакли натиёга еришиш учун. Бу холати коди бошқа шартли баёнот кўшиш учун ташқари, бошқа баёнотда ўхшайди деб хисоблаш мумкин. елсеиф ... яна тузилиши, ўрнак 2.6 кўрсатилган ... агар тўлиқ.

Мисол 2.6. ... else if ... else конструкцияси фигурали қавсдан фойдаланади

```
<?php
if ($bank_balance < 100) {
    $money      = 1000;
    $bank_balance += $money;
}
elseif ($bank_balance > 200) {
    $savings    += 100;
    $bank_balance -= 100; }
else {
    $savings    += 50;
    $bank_balance -= 50; }?>
```

### **Switch инструкцияси**

Switch инструкцияси бир ўзгарувчи ёки иборани аниқлаш натиёсида бир неча аниқликлар мавжуд бўлиши мумкин бўлганда фойдаланиш мумкин, хар бири аниқ функцияни чақирити лозим. PHP коди бошқаруви остида меню тизимини кўриб чиқайлик. Home, About, News, Login и Links – ўзгарувчи \$page эса куйидагилардан бирини маълумоти билан қабул қилади. Код реализацияси if...elseif...else конструкциясидан фойдаланган холда 4.22 мисолда келтирилган кўринишга келиши мумкин.

Мисол 2.7. Кўпсатрли инструкция if...elseif

```
<?php
if ($page == "Home") echo "Вы выбрали Home";
elseif ($page == "About") echo "Вы выбрали About";
elseif ($page == "News") echo "Вы выбрали News";
elseif ($page == "Login") echo "Вы выбрали Login";
```



```
elseif ($page == "Links") echo "Вы выбрали Links";  
?>
```

### Циклларни ташкил қилиш

Компьютерлар тез такрорлаш ва тез хисоблаш қобилияти билан шуғулланади. Кўпинча дастур кодидан яна ва яна такрорлаш талаб қилинади дастур коди кетма-кетлигида, токи қандайдир воқеа содир бўлмагунча, масалан фойдаланувчи томонидан қиймат киритилиши ёки дастур охири натижасига етмагунга қадар. PHP да циклларни ташкил қилишда хар хил тузилмалар масалани ечишда ажойиб усулларни тақдим этади.

While цикли

```
<?php  
$fuel = 10;  
while ($fuel > 1)  
{  
    // Продолжение поездки...  
    echo "Топлива еще достаточно";  
}  
?>
```

### do...while цикли

do...while цикли унча катта бўлмаган while цикли модификациясини тақдим этади, бу яна бир марта ишлаши керак бўлганда фойдаланилади, шарти шундан сўнг текширилади.

do...while циклига мисол 12 га кўпайтириш

```
<?php  
$count = 1;  
do  
    echo "Число $count, умноженное на 12, равно " . $count * 12 . "<br>";  
while (++$count <= 12);  
?>
```

for цикли

```
<?php  
for ($count = 1 ; $count <= 12 ; ++$count)  
    echo "Число $count, умноженное на 12, равно " . $count * 12 . "<br>";  
?>
```

## Цикл ишининг тугаши

Switch инструкциясида кўриб чиққанимиздек for циклини break буйруғи йордамида тўхтатиш мумкин. Бу қачонки инструкцияларнинг бирида хатолик содир бўлиши мумкин бўлганда циклнинг давом этиши хавфсиз хисобланмайди шундай вазиятларда керак бўлади.

Файлни ёзишда дискда жой етмаслиги натижасида шундай хатоликлар содир бўлиши мумкин(мисол 2.8).

```
<?php
    $fp = fopen("text.txt", 'wb');
    for ($j = 0 ; $j < 100 ; ++$j) {
        $written = fwrite($fp, "data");
        if ($written == FALSE) break; }
    fclose($fp);
```

?> Перехват ошибки деления на нуль с помощью инструкции continue

```
<?php
    $j = 10;
    while ($j > -10) {
        $j--;
        if ($j == 0) continue;
        echo (10 / $j) . "<br>";
    }
```

## Назорат саволлари

1. TRUE ва FALSE калит сўзлар қандай маънони англатади.
2. Энг содда иккита ифода формаси нимани ифодалайди?
3. Унар, бинар ва уч компонентали операторлар фарқларини баён қилинг?
4. Оператор устиворлигини ўрнатиш учун энг қулай усул қайси?
5. Ўзаро муносабат операторлар дегани нимани англатади?
6. Қачон (===) операторни ишлатиш керак бўлади ?
7. Учта шартли инструкцияни санаб ўтинг?
8. Жорий циклни сакраб ўтказиб юбориш ва кейинги итерациядан бошлаб кетиш учун қандай оператор ёки команда керак бўлади?
9. Нима учун for оператори while операторидан кучли дейилади?
10. if ва while инструкциялари хар хил турдаги ифодаларни қандай талқин қилади ?

**Фойдаланилган адабиётлар.**

1. VG Olifer, N. A.Olifer. Computer networks. Principles, technologies, protocols. St. Petersburg: Publishing House. "Peter," in 2000.
2. R. Darnell, "JavaScript: Handbook." - St. Petersburg: Publishing House. "Peter," in 1998.
3. Duvanov A.A. WEB. Construction of HTML. - St. Petersburg: BHV, 2003.
4. R. Eisenmenger. HTML 3.2/4.0. Handbook. - M: Univ. "Bean", 1998.
5. Stephen Speynaur, Valerie Kuersia. Directory WEB-master. - Ed. BHV
6. Barrett D. JavaScript. Web-professionals. - Kiev: BHV - Kyiv, 2001.
7. Brandenbau D. JavaScript: a collection of recipes. - St. Petersburg.: Peter, 2000.

**3- мавзу: Массивлар. Ўзгарувчилар номини ўзлаштириш қондаси. Операторлар. Шартли оператор. Switch инструкцияси. MySQLнинг асосий тушунаси.**

**Режа:**

1. MySQL га кириш.
2. MySQL асослари.

**Таянч иборалар:** *MySQL, маълумотлар базаси, сўровлар, жадвал.*

### **3.1. MySQL га кириш.**

Компьютерларга ўрнатилган 10 млн дан ортиқ MySQL нусхалари веб серверлар-даги маълумотлар базалари учун хизмат қилаётгани унинг кенг тарқалганлигидан далолат беради. MySQL 1990-йилларнинг ўрталарида ишлаб чиқилган, ҳозирда интернет ресурслар орасида тўлақонли кенг тарқаладиган замонавий технологиялар қаторига киради. MySQL ва PHP ларнинг кенг тарқалишига яна битта сабаб бу дастурларнинг Open Sours эканлигидир. Бундан ташқари MySQL маълумотлар базасини бошқариш тизими жуда тез ишловчи, кам ресурс талаб қилувчи, замонавий технологиядир. MySQL маълумотлар базасини бошқариш тизими масштабланишга тизимдир, яъни сайт хажмига қараб ошиб боради, охириги ҳолатни билмоқчи бўлсак, <http://mysql.com/whymysql/benchmarks> адресдан билиб олиш мумкин бўлади.

## **2.2. MySQL асослари**

Маълумотлар базаси – бу маълумотларнинг компьютерда сақлаш учун мўлжалланган, тартибланган, структуралашган, керакли маълумотларни тез излаш ва топишга учун мўлжалланган ёзувлар тўпламидир. MySQL номидан кўриниб турибдики, бу Structured Query Language асосида вужудга келган структуралашган сўровлар тилидир. Умумий ҳолда бу тил бошқа тизимларда ҳам ишлатилади, масалан, Oracle ва Microsoft SQL Server. У содда сўровлар ёзиш учун мўлжалланган бўлиб масалан, қўйидагича: `SELECT title FROM publications WHERE author = 'Charles Dickens';`

MySQL маълумотлар базасида бир ёки бир нечта жадваллар бўлиши мумкин, ҳар бир жадвал сатрлар ёки ёзувлардан иборат бўлади. Сатрларда ҳар хил устунлар ёки майдонлар бўлиши мумкин. Бу устунлар ёки майдонларда маълумотлар сақланади. Қўйидаги жадвалда, 5 та китоб ҳақидаги маълумотлар структура билан берилган, автори, номи, категорияси ва чоп қилинган йили.

<b>Author (автор)</b>	<b>Title (название)</b>	<b>Type (категория)</b>	<b>Year (год)</b>
Mark Twain (Марк Твен)	The Adventures of Tom Sawyer («Приключения Тома Сойера»)	Fiction (Художественная)	1876
Jane Austen (Джейн Остен)	Pride and Prejudice («Гордость и предубеждение»)	Fiction (Художественная)	1811
Charles Darwin (Чарльз Дарвин)	The Origin of Species («Происхождение видов»)	Non-Fiction (Научная)	1856
Charles Dickens (Чарльз Диккенс)	The Old Curiosity Shop («Лавка древностей»)	Fiction (Художественная)	1841
William Shakespeare (Вильям Шекспир)	Romeo and Juliet («Ромео и Джульетта»)	Play (Пьеса)	1594

Бу маълумотлар базасини бир қийматли аниқлаш учун publications (издания) маълумотлар базасига мурожаат қиламиз. Агар эътибор қилган бўлсангиз, бу адабиётлар барчаси классик адабиётларга тегишли, шунинг бу базани classics деб атаймиз.

### **MySQL маълумотлар базасининг асосий тушунчалари:**

- маълумотлар базаси – MySQL маълумотлар йиғиндисининг контейнери;
- жадвал— базага киритилган контейнер бўлиб, маълумотларни сақлашга мўлжалланган;
- сатр —алохида ёзув, бир нечта устундан иборат бўлиши мумкин;
- устун — майдон номи сатр ичида жойлашган бўлади.

Юқорида келтирилган қоидалар қатъий адабиётларда келтирилган таърифлар билан устма-уст тушмаслиги мумкин, аммо бу содда тушуниш учун келтирилган маълумот деб қабул қилинг.

### **2.2.2. MySQL га командалар сатридан мурожаат**

MySQL билан ишлашнинг 2 хил билан танишамиз: командалар сатридан, веб-интерфейс дан масалан phpMyAdmin ва PHP дан фойдаланиш.

### **2.2.3 MySQL масофадаги серверда**

Масофада турган сервердаги MySQL маълумотлар базасига мурожаат

қилиш учун масофадаги турган машинага Telnet воситасида чиқилади, бу машина Linux/FreeBSD/UNIX бошқарувида бўлиши керак.

Масофадаги машинага боғлангандан кейин унча катта бўлмаган сервердаги вариацияларни кўриш мумкин.

```
You may also use sysinstall(8) to re-enter the installation and
configuration utility. Edit /etc/motd to change this login announcement.

robnix# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4377812
Server version: mysql-server-5.0.51a

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> show databases;
+-----+
| Database                |
+-----+
| information_schema      |
| mysql                   |
| test                    |
+-----+
3 rows in set (0.02 sec)

mysql> █
```

Агар бу сервер жамоавий фойдаланишга мўлжалланган бўлса, ишлаш тартибида ўзгаришлар бўлиши мумкин. Шунинг учун MySQL га мурожаатда сизда фойдаланувчи номи ва пароли мавжудми аниқлаб олинг. Бу маълумотларни эга бўлсангиз, қўйидаги командани ёзишингиз мумкин, username ўрнига фойдаланувчи номини, parol ўрнига фойдалувчи паролни киритишингиз мумкин. Экранда таклиф пайдо бўлгандан кейин паролни киритиш керак, ундан кейин эса, юқорида кўрсатилгандек тахминан қўйидагича командани киритиш керак : SHOW databases;

Маълумотлар базасида олдинда хосил қилинган маълумотлар ҳам бўлиши мумкин, улар орасида балки test базаси бўлмаслиги мумкин.

Бунда шунга эътибор қилишимиз керак бўлади, барча амаллар тизим администратори бошқаруви остида бўлади, баъзи кутилмаган муаммоларга дуч келишимиз мумкин. Масалан, сиз хосил қилган маълумотлар базаси номига уникал идентификацион сатр жойлаштиришингизни талаб қилиши мумкин, бошқа фойдаланувчилар базасидан фарқлаш учун ва конфликтли

муаммоларнинг олдини олиш учун ва хавфсизлик маъносида шунга ўхшаш талаблар қўйиши мумкин. Муаммо туғилган пайтда тизим администратори билан хал қилинади. Ундан битта фойдаланувчи ва парол олиш керак бўлади маълумотлар базаси яратиш учун ёки сиз учун битта тайёр маълумотлар базаси яратиш ни сўраш мумкин. У холда барча керакли жадвалларни шу базада ташкил қилиш мумкин.

#### **2.2.4. Команда сатри интерфейсидан фойдаланиш**

Ихтиёрий операцион тизим учун (Windows, Mac OS X или Linux) MySQL маълумотлар базаси билан ишлаш командалари деярли фарк қилмайди.

##### **Нуктали вергул**

Энг соддасидан бошлаймиз. Эътибор қилган бўлсангиз, хар бир командадан кейин нуктали вергул қўйилади, масалан, SHOW databases; ?

Бу нуктали вергул битта команда тугагани ёки командаларнинг биридан иккинчисини фарқлаш учун ишлатилади. Агар бу белгини киритишни эсдан чиқарсак хатолик бўлади, сиздан киритишни талаб қилади, узун командаларни бир нечта қисмларга ажратиш имконини беради. Бу белги ёрдамида бир нечта командаларни бир пакет қилиб интерпретаторга узатиш мумкин бўлади. Enter тугмасини босишимиз билан интерпретатор бу командаларни берилиш кетма кетлигида бажаради. Экранда MySQL нинг 6 хил таклифини кўришингиз мумкин бўлади. Сиз кўп сатрли киритишнинг қайси этапида турганлигингизни кўришингиз мумкин бўлади.

#### **2.2.5. MySQL командалари**

Қуйидаги жадвалда энг кўп ишлатиладиган **MySQL** командалари келтирилган

Кўпгина командаларни дарсимиз давомида ўрганамиз, олдин иккита мухим MySQL командаларига тегишли маълумотни эслатиб ўтаимиз.

- SQL командалари катта кичик харфларни фарқламайди, яъни регистрнинг фарқи йўқ, масалан, CREATE, create и CrEaTe лар бир хил

маънода тушунади. Аммо тушунарли бўлиши учун барча командалар учун катта харфларлардан фойдаланишни таклиф қиламиз.

- Жадвал номлари Windows да регистрни фарқламайди, аммо Linux и Mac OS X да фарқлайди, шунинг учун жадвал номларида, фақат битта регистрдан фойдаланилса мақсадга мувофиқ бўларди.

### MySQL буйруқлари

Буйруқлар	Бажарадиган амали
ALTER	МБ ёки жадвалга ўзгартириш киритиш
BACKUP	Жадвалнинг резерв нусхасини олиш
\с	Киритишни инкор қилиш
CREATE	МБ хосил қилиш
DELETE	Жадвалдан сатрни ўчириш
DESCRIBE	Жадвал устинини тавсифлаш
DROP	МБ ёки жадвални ўчириш
EXIT	Чиқиш
GRANT	Фойдаланувчи маълумотларини ўзгартириш
HELP	Ёрдамни ифодалаш
INSERT	Маълумотларни жойлаш
LOCK	Жадвални блоклаш
QUIT	Чиқиш
RENAME	Жадвални қайта номлаш
SHOW	Объектлар хақидаги маълумотлар рўйхати
SOURS	Файлдаги буйруқни бажариш
STATUS	Жорий ҳолатни ифодалаш
TRUNCATE	Жадвални бўшатиш
UNLOCK	Жадвални блоклашни инкор қилиш
UPDATE	Обновление қилиш
USE	МБ фойдаланиш

#### 2.2.6. Маълумотлар базасини ва жадвалларини қуриш

Агар сиз масофадаги серверда ишлаётган бўлсангиз ва сизда битта фойдаланувчи ёзуви ва битта рухсатга эга бўлсангиз битта маълумотлар базаси кўриш имконига эга бўласиз, у ҳолда сиз «Создание таблицы» пунктига ўтишингиз мумкин бўлади. Агар бундай бўлмаса сиз навбатдаги командани киритишни бошлайсиз, янги маълумотлар базасини қуриш учун қуйидаги командани киритасиз:

publications:



CREATE DATABASE publications;

Команда мувоффақиятли бажарилса, қуйидагича хабар пайдо бўлади —  
Query OK, 1 row affected (0.38 sec)(Сўров 1 сатри за 0,38 с да қайта ишланди)

Маълумотлар базаси бажарилгандан кейин, у билан ишлаш учун қуйидаги командани берамиз:

USE publications;

Энди бу командадан кейин маълумотлар базаси ўзгаргани хақида маълумот хабари пайдо бўлиши керак (Database changed), ундан кейин база кейинги командаларни бажаришга тайёр бўлади.

### **Фойдаланувчиларга рухсатни ташкил қилиш**

Энди сиз MySQL фойдаланиш қанчалик соддалигига ишонч хосил қилган бўлсангиз, ўзингизнинг дастлабки маълумотлар базасини қургандан кейинги масалалардан бири бу фойдаланувчиларга мурожаатга рухсат бериш бўлади, акс холда сайтингиз хакерлик ҳаракатлари содир бўлиши мумкин бўлади. Янги фойдаланувчига рухсат бериш учун қуйидагича командалардан фойдаланилади: GRANT бу команда қуйидагича ишлайди: GRANT ПРАВА ON база\_данных.объект TO 'имя\_пользователя@имя\_хоста' IDENTIFIED BY 'пароль';

Бу форма қийинчилик туғдирмаслиги керак, қуйида баъзи параметр ва унинг қийматлари келтирилган:

Энди биз publications даги янги маълумотлар базасига фойдаланувчига рухсат берувчи командани келтириб ўтамиз фойдаланувчи номи jim пароль эса mypasswd бўлган:

GRANT ALL ON publications.\* TO 'jim'@'localhost'

IDENTIFIED BY 'mypasswd';

Бу команда jim@localhost фойдаланувчига publications маълумотлар базаси mypasswd парол билан барча рухсат ва мурожаат лар учун имкон беради. Буни quit командасини бериш билан текшириб кўриш мумкин

бўлади. Ундан кейин MySQL ни қайта ишлатиб юбориб юқорида айтиб ўтилган фойдаланувчи парол ва логинидан фойдаланиш керак бўлади.

### **Жадвални хосил қилиш**

Жорий ҳолатда сиз publications маълумотлар базаси учун барча рухсатлар (ALL правами) билан MySQL тизимида жадвал хосил қилиш учун бўлишингиз керак. Шунинг учун базага қуйидаги командани киритишингиз керак бўлади

```
USE publications;
```

3.1. Мисол classics номли жадвал хосил қилиш

```
CREATE TABLE classics (  
  author VARCHAR(128),  
  title VARCHAR(128),  
  type VARCHAR(16),  
  year CHAR(4)) ENGINE MyISAM;
```

### **Жадвалга маълумотларни киритиш**

Жадвалга маълумотларни киритиш учун INSERT командасидан фойдаланилади

Мисол classics жадвалга маълумот киритиш:

```
INSERT INTO classics(author, title, type, year)  
VALUES('Mark Twain','The Adventures of Tom Sawyer','Fiction','1876');  
INSERT INTO classics(author, title, type, year)  
VALUES('Jane Austen','Pride and Prejudice','Fiction','1811');  
INSERT INTO classics(author, title, type, year)  
VALUES('Charles Darwin','The Origin of Species','Non-Fiction','1856');  
INSERT INTO classics(author, title, type, year)  
VALUES('Charles Dickens','The Old Curiosity Shop','Fiction','1841');  
INSERT INTO classics(author, title, type, year)  
VALUES('William Shakespeare','Romeo and Juliet','Play','1594');
```

Хар иккита сатрдан кейин сўровни мувоффақиятли тугаганлигини текширишингиз керак бўлади— Query OK.

Хаммаси ижобий бўлса, барча сатрларни чиқариши учун қуйидаги командани ёзинг:

```
SELECT * FROM classics;
```

### **Жадвални ўчириш**

Жадвални ўчириш жуда содда. Бунинг учун аввало янги жадвал ҳосил қиламиз ва ўша жадвал мавжудлигини текшираемиз ва ундан кейин ўчираемиз ва уни экранда кўраемиз:

```
CREATE TABLE disposable(trash INT);
```

```
DESCRIBE disposable;
```

```
DROP TABLE disposable;
```

```
SHOW tables;
```

#### **2.2.7. PHP воситасида MySQL маълумотлар базасига сўровлар ёзиш**

PHP дан MySQL да интерфейс сифатида фойдаланишдан мақсад SQL сўровлар натижаларини форматлаш ва уларни веб саҳифада ифодалашдан иборат.

Ўрнатилган MySQL тизимга парол ва логин билан кириб қилинадиган ишларни PHP билан ҳам амалга ошириш мумкин бўлади. Бунда MySQL командалар сатрдан фойдаланиш ўрнига, сўровлар ёзиш ва уни MySQL га узатиш керак бўлади.

MySQL жавоби маълумотлар структураси кўринишида бўлади, бу ҳолда PHP уни таниб олади ва қайта ишлаши учун қулай бўлади. Ундан кейин PHP қайта ишлайди ва веб саҳифа форматига айлантиради.

MySQL дан PHP ёрдамида фойдаланиш жараёни қуйидагилардан иборат бўлади:

1. MySQL га боғланиш.
2. Фойдаланмоқчи бўлган маълумотлар базасини танлаш.

3. Сўровлар сатрини хосил қилиш.
4. Сўровларни бажариш.
5. Натжаларни олиш ва веб саҳифага чиқазиш.
6. 3 ва 5 пунктларни барча зарурий маълумотларни олгунча такрорлаш
7. MySQL ни бекитиш.

Ундан кейинги жараёнлар босқичма босқич амалга оширилади, энг аввало тизимга хавфсиз кириш қуриб олиш керак бўлади.

### **Регистрация файлини хосил қилиш**

Кўпгина сайтлар PHP да ишланган к MySQL маълумотлар базаси билан ишлайди, унга мурожаат қилиш учун парол ва логин керак бўлади. Шунинг учун алоҳида файл керак бўлади, бу файл номи одатда login.php деб аташ мумкин. Қуйида шу файл тексти берилган.

Файл login.php

```
<?php // login.php
$db_hostname = 'localhost';
$db_database = 'publications';
$db_username = 'имя_пользователя';
$db_password = 'пароль';
?>
```

Бу файлда асосий ролни қамраб олувчи теглар ўйнайди <?php и ?>, бу теглар орасидаги сатрлар PHP кодлар деб интерпретация қилинади.

Агар қамраб олувчи теглар қўйилмаса сайтда файл чақирилганда барча кодларининг текст кўринишида сайтда пайдо бўлиши мумкин. Агар хаммаси жойида бўлса, сайтда бўш сатр пайдо бўлади. Бу файлларни ҳеч қандай муаммосиз бошқа PHP-файлларга қўшиш мумкин бўлади.

`$db_hostname` ўзгарувчи PHP га кайси компьютер маълумотлар базаси билан ишлаши ҳақидаги хабарни жўнатади. Бундай ўзгарувчининг мавжудлиги сиз барча компьютерлардаги MySQL маълумотлар базасига

рухсат олишингиз мумкинлигини билдиради, PHP ўрнатилган, у ихтиёрий хостни бириктириб бериш имкони мавжуд бўлади.

### **MySQL га боғланиш**

```
<?php  
  
require_once 'login.php';  
  
$db_server = mysql_connect($db_hostname, $db_username, $db_password);  
  
if (!$db_server) die("Невозможно подключиться к MySQL: " . mysql_error());  
  
?>
```

### **Назорат саволлари**

1. Маълумотлар базаси деганда нимани тушунаси?
2. Қанақа маълумотлар базаси дастурларини биласиз ва уларни имкониятлари?
3. PHP да MySQL билан боғланиш функциясини тушунтириб беринг?
4. PHP да MySQL сўровларини амалга оширишга мисол келтиринг?

### **Фойдаланилган адабиётлар.**

1. VG Olifer, N. A.Olifer. Computer networks. Principles, technologies, protocols. St. Petersburg: Publishing House. "Peter," in 2000.
2. R. Darnell, "JavaScript: Handbook." - St. Petersburg: Publishing House. "Peter," in 1998.
3. Duvanov A.A. WEB. Construction of HTML. - St. Petersburg: BHV, 2003.
4. R. Eisenmenger. HTML 3.2/4.0. Handbook. - M: Univ. "Bean", 1998.

**4- мавзу: Буйруқлар сатридан MySQL га мурожаат қилиш. MySQL буйруқлари. MySQL маълумотлар базасига PHP дан мурожаат қилиш. JavaScript га кириш. JavaScript ва HTML. Режа:**

1. JavaScript га кириш.
2. JavaScript ва HTML матнлари.
3. JavaScript кодларни отладка қилиш.
4. Изохлар(comment) дан фойдаланиш.
5. Массивлар.
6. Арифметик операторлар.

**Таянч иборалар:** *JavaScript тили функцияси, Global, Math, Date синфлари, объектлар.*

#### **4.1. JavaScript га кириш**

JavaScript сизнинг сайтингизга динамик функционалик беради. Сиз сайтни кўраётганингизда сичқонча кўрсатгичи браузернинг қандайдир элементида кўрсатганда содир бўладиган ходисалар (янги матн пайдо бўлиши, ранг ва расмларнинг ўзгариши ва ҳақозо эффектлар ) JavaScript ёрдамида амалга оширилади. Бу тил шундай эффектларни таклиф ва тавсия қила оладики, бошқа воситалар билан амалга ошириш мумкин эмас ва шу билан бирга бевосита веб браузер ичида веб ҳужжатнинг барча элементларига мурожаат қила олади.

1995 йилда Netscape компанияси мутахассислари клиент томонидаги саҳифа-ларни генерация қилиш учун махсус дастурлаш тили яратишди ва уни JavaScript деб номлашди.

Дастлаб бут тил тўғрисида нотўғри тасаввурлар пайдо бўлди, гўёки бу тил JAVA тили ортиқча иловаси деб тушунишди, улар орасида боғланиш бор деб ўйлашди. Тилнинг бундай номланиши фақатгина маркетинг маъносида омадли усул эди, чунки JAVA дастурлаш тили жуда машхур, шунга кўра янги сценариялар тили JavaScript тезда дастурчилар назарига тушди ва машхур бўлиб кетди. JavaScript тилига веб саҳифанинг HTML элементи

хужжатнинг объект модели DOM (Document Object Model) қўшилгандан кейин янада машхур бўлиб кетди. Объект модели хужжатга янги абцас ёки матннинг бирор қисмини ўзгартириш имконини беради. JavaScript ва PHP ҳам структурали дастуралаш элеменларини қўллаб қувватлайди худди C тилидаги каби бу иккала тил ҳам бир бирига ўхшаб кетади.

Бу иккала тил ҳам юқори даражали тиллар гурухига киради. Масалан, иккаласида ҳам кучсиз турлаш бир ўзгарувчи турини янги контекст бошқасига алмаштириш жуда осон. PHP билан танишгандан кейин JavaScript ўрганиш жуда осон кечади. Бу тилни ўрганиш сизга жуда катта фойда келтиради, чунки бу тил янги технологиялар Web 2.0 AJAX асоси хисобланади. Булар асосида мослашувчан HTML5 ва унинг фойдаланувчи интерфейсини ўзлаштишингиз осон бўлади.

## 4.2. JavaScript ва HTML матнлар

JavaScript сценариялар ёзишга мўлжалланган бўлиб, браузер да клиент томонидан веб саҳифалар сценарияларини бошқарувчи тилдир.

Бу тилга мурожаат қилиш учун очилаган ва ёпилган HTML теглар ичига script сўзи ёзилади, қуйидагича:

`<script>` ва `</script>`. Одатий хужжат Hello World, HTML 4.01 да ифодаланган бўлса унга

JavaScript ни татбиқ қилсак у қуйидагича бўлади, Масалан 4.1. Hello World матни JavaScript воситасида қуйидагича ифодаланади:

```
<html>
  <head><title>Hello World</title></head>
  <body>
    <script type="text/javascript">
      document.write("Hello World")
    </script>
  </noscript>
```

Ваш браузер не поддерживает JavaScript, или его поддержка

отключена

```
</noscript>
```

```
</body>
```

```
</html>
```

Бунда фақат битта сатр JavaScript тегишли, `<script>` теги остидаги `document.write` команда бу команда PHP даги `echo` ёки `print` командаларига эквивалентдир. Бу сатр ундаги сатр маълумотни экранга чиқаради. Бунда шунга эътибор қилиш мумкин командалардан кейин нуқтали вергул ишлатилмади, чунки бунда нуқта вергул янги сатр ўтишни тушунади. Агар битта сатрга бир нечта инструкцияларни жойлаштириш керак бўлса нуқта вергул (охиргисидан ташқари) қўйилади.

Хохласангиз хар бир инструкциядан кейин битта сатрда бўлмаса ҳам нуқтали вергул қуйиш мумкин. Бу JavaScript ишига халақит қилмайди.

Бу мисол `<noscript>` ва `</noscript>` теглар жуфтлигига эътибор қаратишимиз керак, бу бошқа фойдаланувчилар учун қайсики JavaScript ни қўллаб қувватламайдиган браузерларни ишлатаётганда ёки браузер ишламай қолганда хар хил хабарлар чиқариш ва ҳақозолар учун ишлатилади. Кейинги мисолларда `<noscript>` тегдан фойдаланмасдан туриб ҳам юқорида амалга оширилган амалларни бажариш мумкинлигини кўрамиз.

4.1 мисолни ишга туширсак қуйидагича матн пайдо бўлади:

Hello World

Агарда JavaScript ажратилган бўлса ухолда қуйидагича хабар чиқади 4.1 расмдаги каби хабар чиқади:



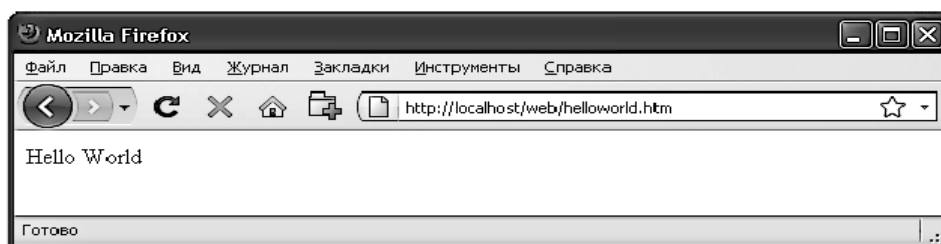


Рис. 14.1. Включенный и работающий JavaScript

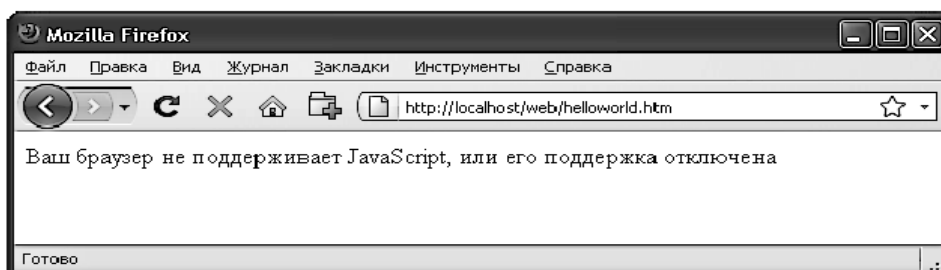


Рис. 14.2. JavaScript отключен

### 4.3. JavaScript кодларни отладка қилиш

JavaScript кодни отладка қилиш JavaScript ни ўрганишда жуда муҳим саналади, бунда коднинг қаерида хатолик бўлганини аниқлаш ва шунга ўхшаган ишларни ўрганиш мумкин бўлади. PHP дан фарқли равишда JavaScript хатоликларни браузерга чиқармайди, хаттоликлар браузерларга қараб ҳар хил бўлиши мумкин, Шунинг учун қуйида 5 та браузерда JavaScript да қўзатиладиган хатоликларни рўйхати келтирилган

Таблица 14.1. Доступ к сообщениям об ошибках JavaScript в различных браузерах

Браузер	Способ доступа к сообщениям об ошибках JavaScript
Apple Safari	В Safari нет консоли ошибок, включенной по умолчанию, но вы можете включить эту функцию, выбрав в меню пункты Safari ▶ Настройки ▶ Дополнения и установив флажок Показывать меню "Разработка" в строке меню. Кроме того, предпочтение можно отдать JavaScript-модулю Firebug Lite, который многие считают более простым в использовании
Google Chrome	Щелкните на значке меню, который похож на страницу с загнутым углом, и выберите пункт Разработчикам ▶ Консоль JavaScript. Можно также нажать сочетание клавиш Ctrl+Shift+J на PC или Command+Shift+J на Mac
Microsoft Internet Explorer	Выберите команду Сервис ▶ Свойства обозревателя ▶ Дополнительно, снимите флажок Отключить отладку сценариев и установите флажок Показывать уведомление о каждой ошибке сценария
Mozilla Firefox	Выберите команду Инструменты ▶ Консоль ошибок или воспользуйтесь сочетанием клавиш: Ctrl+Shift+J на PC или Command+Shift+J на Mac
Opera	Выберите команду Инструменты ▶ Дополнительно ▶ Консоль ошибок

Хатоликларни фарқлаш учун, одатда кўп қўзатиладиган хатоликлари мавжуд сценарияларни қараймиз. Бекилиши керак бўлган қавслар

беркилмаган "Hello World" сатрида, бундай синтактик хатоликлар жуда кўп учраб туради.

Масалан 4.2. да JavaScript-сценарияда Hello World сатридаги хатоликлар

```
<html>
  <head><title>Hello World</title></head>
  <body>
JavaScript и текст HTML342
  <script type="text/javascript">
    document.write("Hello World)
  </script>
</body>
</html>
/html>
```

Бу кодни test.html файлига сақлаб браузердан чақириб ишга туширсангиз. У фақат саҳифа номини чиқариб бера олади, асосий ойна бўш бўлади. Энди консол хатоликларни чақириб кўрсангиз ёпилмаган қавслар ҳақидаги хабарларни кўрасиз, худди 14.4 мисолдаги каби.

Ўнг томонда манбага кўрсаткич турган бўлади, сичқончани боссангиз хатолиги мавжуд сатр пайдо бўлади.

Масалан 4.2. Mozilla Firefox хатоликлар консол хабарида қуйидагича:

```
SyntaxError: unterminated string literal
```

Unterminated string constant да хатоликлар ҳақидаги хабарда бунда ёрдамчи маълумотлар бўлмайди аммо хатоликнинг қайси сатрда ва позиция кўрсатилади.

Google Chrome ва Opera да хатоликлар ҳақидаги хабарда худди 4.4. мисолдаги каби бўлади аммо сатр кўрсатилади, масалан, Chrome/Opera Uncaught SyntaxError: Unexpected token ILLEGAL каби.

Apple Safari да эса хатоликлар хақидаги хабарда худди 4.57 мисолдаги каби бўлади, кўрсаткич(ссылка)дан ўнг томонда туради, бунда сатр номери ва хатолик кўрсатилган бўлади.

Сатрни ажратиш учун кўрсаткич(ссылка)ни сичқонча билан активлаштирамиз, аммо хатоликни жойини кўрсатмайди.

4.5. масала хатолиги хақидаги хабар консолда қуйидагича: Safari SyntaxError: Unexpected EOFЕ маълумот чиқади.

Бундай хабар етарли эмас, Firefox учун эса ва Chrome учун қўшимча модул мавжуд Firebug номли, бу модул купгина муаммоларни хал қилади. Бу модул JavaScript дастурчилар орасида анча машхурдир. Бу модул катта эътибор қаратиш керак бўлади.

#### **4.4. Изохлардан фойдаланиш**

Меросхўрлик асосида бу PHP ва JavaScript тил хам C тилидан кўплаб дастурлаш элементларини қабул қилиб олган масалан изохларни, арифметик амалларни, инкремент декримент ва хакозоларни, худди шунингдек бу иккала тил умумий томонлари мавжуд буни мисоллар орқали кўрдик.

#### **JavaScript кодда изохларни ўрганиш.**

Худди C++ тилидаги каби битта сатрда изох ёзмоқчи бўлсак :

```
// this is comment
```

Каби ёзилади.

Агар кўп сатр изох ёзмоқчи бўлсак :

```
/* this
```

```
is
```

```
other
```

```
comment
```

```
*/
```

Каби ёзилади.

Ичма ич жойлашган изохлар мумкин эмас.

Бунда битта сатрда битта ифода ёзилса нуқтали вергул талаб қилинмайди, масалан, `x += 10` битта сатр ёзиш мумкин.

Агар битта сатрда бир нечта ифода ёзишга зарур бўлса, масалан, `x += 10; y -= 5; z = 0` у холда улар орасида нуқтали вергул зарур бўлади.

Сатрли ўзгарувчилар JavaScript да битталиқ ёки иккиталиқ қўштирноқ остидаги қийматларни олиши мумкин:

```
greeting = "Привет! "  
warning = 'Осторожно!'
```

Иккиталиқ қўштирноқ ичига битталиқ қўштирноқ жойлаштириш мумкин ёки аксинча битталиқ қўштирноқ ичига иккиталиқ қўштирноқ жойлаштириш мумкин.

Бир хил қўштирноқларни тескари слеш воситасида бекор қилиш мумкин бўлади:

```
greeting = "\"Привет!\" является приветствием"  
warning = '\'Oсторожно!\' является предупреждением'
```

Сатр ўзгарувчиларининг қийматини бошқа ўзгарувчидан ўқиб олиш учун бошқа ўзгарувчига ўзлаштириш мумкин бўлади, масалан:

```
newstring = oldstring
```

Функциядан ўзлаштириш мумкин бўлади:

```
status = "Все системы работают успешно"  
document.write(status)
```

#### 4.5. Массивлар

JavaScript массивлар PHP га жуда ўхшаб кетади, бунда ўзгарувчилар сатрли ёки сонли қийматларни ва бошқа массивларни ўзлаштиришлари мумкин бўлади. Массивларни ўзлаштиришлари учун қуйидаги синтаксисдан фойдаланилади, масалан,

```
toys = ['bat', 'ball', 'whistle', 'puzzle', 'doll']
```

Катта ўлчамдаги массивни хосил қилиш учун кичик ўлчамдаги массив йиғилади.

Рангли квадратнинг икки ўлчовли массивини хосил қилиш учун эса қуйидагича ишлар амалга оширилади: Red (R), Green (G), Orange (O), Yellow (Y), Botl (B) ва White (W), —ёки буни қуйидаги код билан ифодалаш мумкин:

```
face =  
[  
  ['R', 'G', 'Y'],  
  ['W', 'R', 'O'],  
  ['Y', 'W', 'G']  
]
```

Олдинги мисолни қуйидагича ёзиш мумкин бўлади:

```
face = [['R', 'G', 'Y'], ['W', 'R', 'O'], ['Y', 'W', 'G']]
```

ёки янада бошқача кўринишда ёзиш мумкин:

```
top = ['R', 'G', 'Y']  
mid = ['W', 'R', 'O']  
bot = ['Y', 'W', 'G']  
face = [top, mid, bot]
```

Бу массивдаги элементга мурожаат қилиш учун масалан, тепадан иккинчи, чапдан учинчи элементга мурожаат қуйидагича амалга оширилади (эслатиб ўтамиз массив элементлари 0 дан бошланади):

```
document.write(face[1][2])
```

Бу инструкция O харфни ёки Orange рангини билдиради

#### 4.6. Арифметик операторлар

Арифметик операторлар математик амалларни бажариш учун ишлатилади. Бунда асосан қўшиш, айириш, кўпайтириш, бўлиш, инкремент, декремент ва 4.1 жадвал да келтирилган:

Оператор	Тавсифи	Мисол
+	Қўшиш	J+12
-	Айириш	J-12
*	Кўпайтириш	J*12
/	Бўлиш	J/3.13
%	Модулли бўлиш	J% 12

++	Инкремент	++j
--	Декремент	--j

### Ўзгартириш оператори

Бу операторлар ўзгарувчиларга қийматларни ўзлаштириш учун ишлатилади .

Булар қуйидагилар:

$$1) S += n \Leftrightarrow S = S + n$$

$$2) S -= m \Leftrightarrow S = S - m$$

$$3) S *= p \Leftrightarrow S = S * p$$

$$4) S /= q \Leftrightarrow S = S / q$$

Агар S нинг дастлабки қиймати маълум бўлса, масалан  $S=3, n=2, m=1, p=3, q=1$ ;

У холда

$$1) S += n \Leftrightarrow S = S + n \Rightarrow S = 5$$

$$2) S -= m \Leftrightarrow S = S - m \Rightarrow S = 2$$

$$3) S *= p \Leftrightarrow S = S * p \Rightarrow S = 9$$

$$4) S /= q \Leftrightarrow S = S / q \Rightarrow S = 3$$

Қуйида JavaScript даги таққослаш амаллари келтирилган

Оператор	Тавсифи	Мисол
==	Тенг	J==42
!=	Тенг эмас	J!=17
>	Катта	j>0
<	Кичик	J<100
>=	Катта ёки тенг	j>=23
<=	Кичик ёки тенг	J<=23
===	Тенг (турига кўра тенг)	J===56
!==	Тенг эмас(турига кўра )	J!=='1'

## Назорат саволлар

1. JavaScript кодни якунлаш учун қайси теглар ишлатилади?
2. Хужжатнинг қайси қисми одатга кўра маълумот киритади JavaScript кодга?
3. Бошқа манбадан сизнинг хужжатингиз қандай қилиб JavaScript кодга жойлаша олади?
4. JavaScript қандай функциялари PHP-нинг echo ёки print командаларига эквивалент бўла олади?
5. JavaScript изоҳлар қандай ёзилади?
6. JavaScript да сатрларни қандай команда бирлаштиради?
7. Қандай калит сўзлар татбиқ қилинади ўзгарувчиларни локал ифодалаш учун?
8. Барча браузерларда ишловчи метод кўрсатинг қайсики URL-адрес олинган ссылкани id орқали ифодалай олсин?
9. JavaScript нинг қандай иккита командаси олдинги history массивидаги саҳифани браузерга юклайди ?
10. oreilly.com сайтининг жорий бош саҳифасини ўзгартириш учун JavaScript қандай командасидан фойдаланасиз?

## Фойдаланилган адабиётлар

1. VG Olifer, N. A.Olifer. Computer networks. Principles, technologies, protocols. St. Petersburg: Publishing House. "Peter," in 2000.
2. R. Darnell, "JavaScript: Handbook." - St. Petersburg: Publishing House. "Peter," in 1998.
3. Duvanov A.A. WEB. Construction of HTML. - St. Petersburg: BHV, 2003.
4. R. Eisenmenger. HTML 3.2/4.0. Handbook. - M: Univ. "Bean", 1998.
5. Stephen Speynaur, Valerie Kuersia. Directory WEB-master. - Ed. BHV
6. Barrett D. JavaScript. Web-professionals. - Kiev: BHV - Kyiv, 2001.
7. Brandenbau D. JavaScript: a collection of recipes. - St. Petersburg.: Peter, 2000.

**5- Мавзу: JavaScript ни файлга жойлаш. Java-Script кодларни отлатка қилиш. HTML5 ва CSS3 ўрганиш. CSS асослари. PHP тилида дастурлаш асослари. MySQL маълумотлар базасида ишлаш. MVC технологиялари**

**Режа:**

1. CSS нинг CSS3 ҳисобига кенгайтиши .
2. Атрибутлар селектори.
3. Background -clip хусусиятлари.
4. Background –origin хусусиятлари.
5. HTML5 га кириш.
6. Холст.
7. Геолокация.

**Таянч иборалар:** *OpenGL ES, Source code, Universal app, GUI, IDE, OS (Operating System), Resources, Service, Surface, SurfaceView, View, Viewgroup Widget*

### **5.1. CSS нинг CSS3 ҳисобига кенгайтиши**

CSS нинг биринчи версиялари 1996 йилда ишлаб чиқилган, 1999 ва 2001 йилда ишлаб чиқилгани барча браузерларга мос келган. Бу версия (CSS1) стандарти 2008 йилда қайта кўрилган. Иккинчи версия дастлаб 1998 йилда бошланган, стандарти 2007 йилда ва 2009 йилда қайта кўрилган ва яқунланган. 2001 йилда CSS3 версия ишлаб чиқилган ва унинг характеристикалари 2009 йилда шаклланган ҳозиргача ривожлантирилиб бориляпти.

Хали CSS3 стандарти охирига етмаган бўлса ҳам CSS4 устида иш бошлаб юборилди. CSS4 ни кўплаб браузерлар қабул қила бошлади. CSS4 нинг баъзи характеристикаси JavaScript нинг баъзи функционалликни ўрнини боса олади.

JavaScript ва CSS3 биргаликда динамик хусусиятлари ишлаб чиқишга ҳаракат қилишни маслаҳат бераман.

Тасаввуримиздаги CSS хусусиятлари ҳужжат атрибутларини шу ҳужжатнинг қисмига айлантиради, улар JavaScript ёрдамисиз ҳужжатга бирлаш-тирилади. Қачон у ҳужжат қисмига айланса бу тушунарли ҳолат.



## 5.2.Атрибутлар Селектори

Селектор бу форматлаш безаш параметрларини ўз ичига олувчи стилдир. Селекторларга теглар, синфлар, идентификаторлар киради. Ўнлаб селекторлар мавжуд бўлиб, CSS3 ишлаб чиқган дастурчилар бу селекторлар яхши ишлаши учун кўплаб янгиланишлар киритишда давом этиптилар. Масалан керакли селекторни уни атрибутига қараб топадиган янгиланишлар қилишди.Бунга мисол қилиб, CSS2 даги href-атрибути 'info.htm' сатри ўрнига энди a[href='info.htm'] ишлатиладиган бўлди. CSS3 да яна бошқа ўзгаришлар 3 та янги операторлар : ^, \$ ва \* пайдо бўлди.

### Оператор ^

Бу оператор сатр бошидан излайди, масалан, қуйидаги href-атрибут қайси сатр <http://website> билан бошланса ўшани излайди.

a[href^='http://website'] Шунга кўра унга мос келувчи сатр қуйидагича бўлиши мумкин :<a href='http://website.com'>

Қуйидаги элемент эса мос келмайди:

<a href='http://mywebsite.com'>

### Оператор \$

Сатр охиридаги мослик билан излаш эса қуйидаги селектор билан мос келади, ихтиёрий img-тег нинг src-атрибути .png билан тугаса қуйидагича код билан амалга оширилади:

img[src\$='.png']

Масалан бунга қуйидаги тег мос келади:

<img src='photo.png' /> Қуйидаги тег мос келмайди:

<img src='snapshot.jpg' />

### Box –sizing хусусияти

W3C модел блокида аниқланган объектнинг эни ва буйи ундаги элементнинг ўлчамига боғлиқ бошқа чегара ва чекланишларни тан олмайди.

Аммо баъзи веб дастурчилар барча элементлар учун ўлчовларни кўрсатиш хатто чегара чекловларни аниқлаштиришни хошлашади.

Бундай хохишни амалга ошириш учун CSS3 сизга шундай блокни

такклиф қилиши мумкинки, сиз хоҳлаган ўлчамдаги блок билан ишлаш имконини беради бу `box-sizing` деб аталади.

Масалан объектнинг умумий буйи ва энини чегара ва чеклашларни ҳисобга олган қуйидаги эълон билан амалга ошириш мумкин:

```
box-sizing:border-box;
```

Ёки объектнинг буйи ва эни фақат ундаги элементники бўлса уҳолда қуйидаги эълон билан амалга ошириш мумкин

```
box-sizing:content-box;
```

### **5.3. background-clip хусусиятлари**

Бу хусусият фон ишлатиш ёки йўқлигини аниқлайди, фон чегараланган соҳада ичида бўлсинми ёки ташқарисидами, шуни аниқлаш учун ишлатилади. Масалан қуйидаги эълон фон элементнинг барча қисмларида чегараланган соҳагача ифодалайди:

```
background-clip:border-box;
```

Агар эълон фон элементнинг чегараларида ифодаласа:

```
background-clip:padding-box;
```

Агар эълон фон элементнинг ички қисмини ифодаласа:

```
background-clip:content-box;
```

5.1. расм браузерда ифодаланган учта элемент кўрсатилган:

биринчида Safari браузерда `background-clip` хусусият `border-box` қиймат билан;

иккинчидан Safari браузерда `background-clip` хусусият `padding-box` қиймат билан;

иккинчидан Safari браузерда `background-clip` хусусият `content-box` қиймат билан;

Ички блокнинг биринчи қаторида элементни ихтиёрий ерда ифодалашга рухсат берилади.

Бундан шуни айтиш мумкинки у биринчи блок доирасида ифодаланади, бунда стил чегараси пунктир билан аниқланади.

Иккинчи қаторда соҳа доирасида фон ва унинг тасвирлари ифодаланмайди, улар background-clip хусусиятнинг padding-box қийматига кўра аниқланади.

Учинчи қаторда хусусиятлар элемент хусусиятига background-clip га content-box қийматни бериш билан ҳар бир элементнинг ички мазмунини аниқлаш учун ишлатилади.

#### 5.4. background-origin хусусияти

Бу хусусият билан фон тасвир қаерда туришини ва қаердан бошлашини аниқлаштириш (кўрсатиш) мумкин бўлади.

Масалан кейинги эълон фон тасвири ташқи чегаранинг чап юқори қисмидан бошланишини кўрсатади.

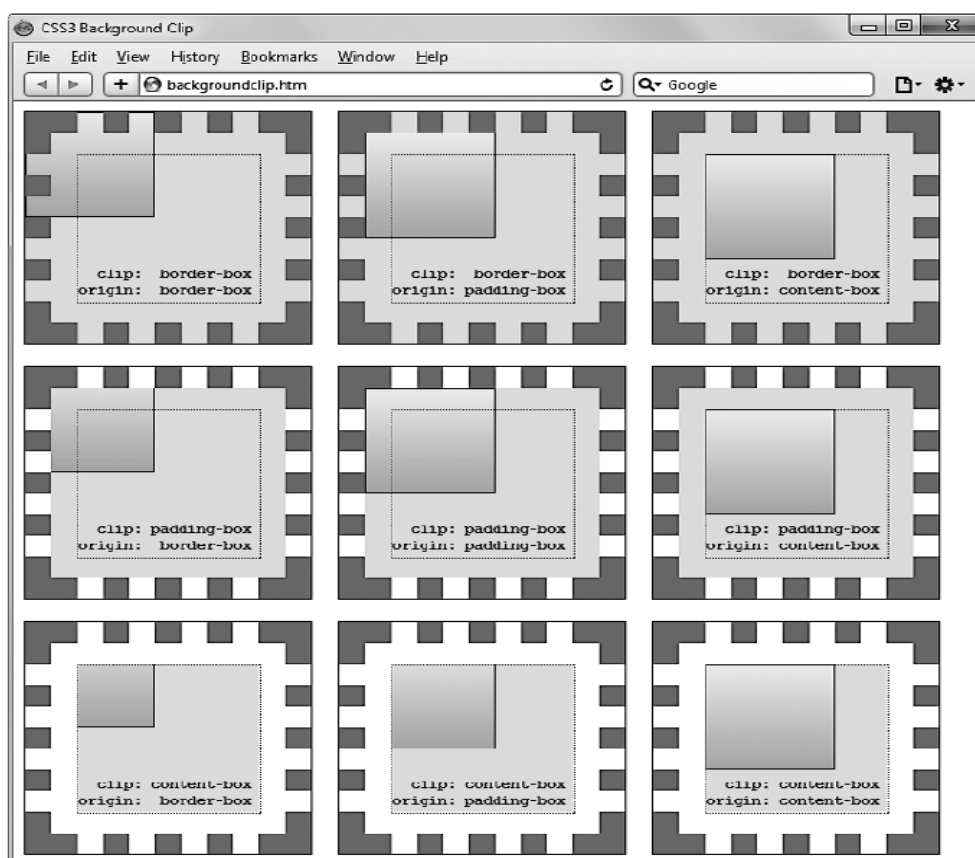


Рис. 20.1. Разные способы сочетания свойств фона CSS3

5.1.-расм. юююююююю

Тасвир ўрнатишни чап юқори бурчакдан бошланиши учун қуйидаги эълондан бошлаш керак бўлади:

```
background-origin:padding-box;
```

Агар тасвирни ўрнатишни чап юқори бурчак элемент ичкарасидан бошлаш учун эса қуйидаги эълондан бошлаш керак бўлади:

```
background-origin:content-box;
```

5.1 расмга қаранг бунда биринчи блок учун элемент хусусияти учун background-origin унинг қиймати учун эса border-box ,иккинчи блок учун элемент хусусияти учун background-origin унинг қиймати учун эса padding-box, учинчи блок учун элемент хусусияти учун background-origin унинг қиймати учун эса content-box ишлатилипти.

### **background-size хусусияти**

Худди шундай <img> тегидаги ўлчовларни аниқлаш усули билан бир хил фон тасвири улчамларини аниқлаш. Кейинги барча браузерлар фонни худди шундай ифодалайди.

Хусусият қуйидагича аниқланиши мумкин масалан, ww —эни, hh — буйи

```
background-size:ww px hh px;
```

Зарурат бўлса битта аргументдан фойдаланиш мумкин, агар иккита ўлчов бир хил бўлса.

Агар бу хусусиятни блок элементларига татбиқ қилсак масалан <div> ёки <span> булар га нисбатан эни ва буйи улчамларини процентда кўрсатиш керак бўлади.

### **auto қийматидан фойдаланиш**

Фон тасвирларидан битта ўлчамини ўзгартириш керак бўлса, бошқа улчамлар учун auto қиймати ишлатилади, масалан,

```
background-size:100px auto;
```

Бу эълон билан эни 100 пиксел ўрнатилади, баландлик шунга мос пропорционал ўзгаради.

## **5.5. HTML5 га кириш**

HTML5 тили веб-дизайнда анча илдамлаб кетди, татбиқ қилиш қулайлиги билан машхур бўлди.

У графика билан браузерда қўшимча модулларсиз қулай ишлаш имконини берди, масалан Flash ва шунга ўхшаган аудио ва видео файлларни веб саҳифага модулларсиз жойлаш имконини берди.

Худди шунингдек бошқа янгиликлар, масалан, фойдаланувчининг ўрнини аниқлаш, ишчи веб жараёни, формани қайта ишлаш яхшиланган, локал маълумот сақлаш воситаларига мурожаат яхшиланган, веб саҳифани мобил веб иловага айлантириш имконияти пайдо бўлган.

Шундай пайт келадиги бу тил авторлари томонидан мукамал ўзгаришлар яқунлангандан кейин фойдаланиш имкони бўлади.

Улар ҳар қандай бўлмасин биз иккинчи динамик веб интерактивлик ривожланиш палласида турганимиз бизга аён, чунки буни Web 2.0 ҳам билса бўлади.

Аммо буни биз Web 3.0 деб айтишга хали эрта деб ўйлайман, шунинг учун ҳам HTML5 билан чегараланишяпти.

Нима бўлса ҳам қизиқ Web 3.0. қандай янгиликлар олиб келади.

Агар тахмин қиладиган бўлсак бу технология сунъий интеллект татбиқлари

Siri, Apple, Cortana, Microsoft , Watson , IBM компанияларидаги каби технологияларни, масалан товушли излаш, товушли маълумотлар киритиш, визуал клавиатура каби худди Google Glass, Galaxy Gear соатларидаги каби янгиликларни кутиш мумкин.

HTML5 даги барча янги имкониятлар барча браузерларда фойдаланиш мумкин ва ишлайди

## **5.5. Холст**

Дастлаб Apple компанияси ишлаб чиққан визуаллаштирувчи дастур WebKit (Safari браузер ) учун энди кўпгина iOS, Android, Kindle, Chrome, BlackBerry, Opera и Tizen браузерлар учун вебсаҳифага график элементлар чизиш имконини беради қўшимча модулларсиз.

Canvas стандартлаштиргандан кейин барча браузерлар учун машхур веб дастурлаш воситасига айланди.

Бошқа HTML-элементлар каби canvas – бу оддий веб саҳифа элементи олдиндан ўлчами аниқланган, ичида JavaScript дан фойдаланиб графика билан ишлаш мумкин.

Холст <canvas> теги ёрдамида хосил қилинади, уни идентификатор билан аниқлаш керак, чунки JavaScript кодда қайси холстга мурожаат

5.1 мисолда mycanvas номли <canvas> элементи хосил қилинган. Япон байроғи расми солинган.

Мисол 5.1. HTML5да canvas элементи

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Холст HTML5</title>
```

```
<script src='OSC.js'></script>
```

```
</head>
```

```
<body>
```

```
<canvas id='mycanvas' width='320' height='240'>
```

```
Это элемент canvas с идентификатором <i>mycanvas</i>
```

```
Этот текст виден только в браузерах, не поддерживающих HTML5
```

```
</canvas>
```

```
<script>
```

```
canvas = O('mycanvas')
```

```
context = canvas.getContext('2d')
```

```
context.fillStyle = 'red'
```

```
S(canvas).border = '1px solid black'
```

```
context.beginPath() context.moveTo(160, 120)
```

```
context.arc(160, 120, 70, 0, Math.PI * 2, false)
```

```
context.closePath()
```

```
context.fill()
```

```
</script>  
</body>  
</html>
```

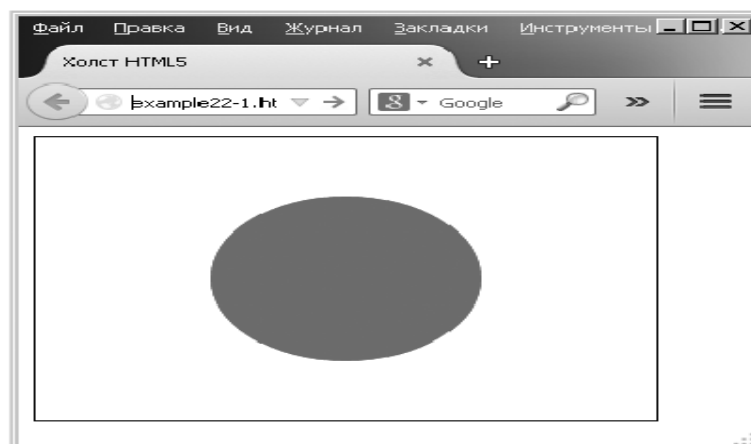


Рис. 22.1. Рисование японского флага на холсте HTML5

Юқоридаги кодни тахлил қилиш маънога эга эмас деб ўйлаймиз, хозирда холстан фойдаланиш соддалигини билишимиз бу ишни амалга ошириш учун бир нечта JavaScript нинг функциясини билишимиз кифоя қилади .

### 5.6. Геолокация

Геолокациядан фойдаланганда сизнинг браузерингиз веб серверга сизнинг жойлашган ўрнингиз хақидаги маълумотни қайтариши мумкин.

Бу маълумотлар GPS микросхемаларидан олинган бўлиши мумкин, қайсики сиз фойдаланаётган компьютер ёки мобил қурилмадаги, ёки Wi-Fi даги IP-адресга кўра аниқланган бўлиши мумкин.

Хавфсизлик маъносида, фойдаланувчи хар доим вазиятни назорат қилади, ва хохлаسا бу хизматдан фойдаланади ёки бутунлай бу хизматни блоклаб қуйиши мумкин бўлади.

Бу технология жуда кўп татбиқларига эгадир, масалан қадамли навигация, маҳаллий хариталар, энг яқин ресторан хақидаги маълумотлар, Wi-Fi борлиги хақидаги, дўстлар хақидаги ва хакозалар.

5.2 мисолда Google харитаси келтирилган фойдаланувчи жойлашган ўрнини кўрсатяпти, бунда фойдаланувчи браузерни геолокацию рухсат берилган ва фойдаланувчи ўзи турган жойни аниқлаган. Акс холда хатолик

хақида хабар берилади.

Мисол 5.2. Фойдаланувчи ўзи турган жойни аниқлаш

```
<!DOCTYPE html>
<html>
  <head>
    <title>Пример геолокации</title>
    <script src='OSC.js'></script>
    <script
src="https://maps.googleapis.com/maps/api/js?sensor=false"></script>
  </head>
  <body>
    <div id='status'></div>
    <div id='map'></div>
    <script>
      if (typeof navigator.geolocation == 'undefined')
        alert("Геолокация не поддерживается.")
      else
        navigator.geolocation.getCurrentPosition(granted, denied)
      function granted(position)
      {
        O('status').innerHTML = 'Разрешение дано'
        S('map').border = '1px solid black'
        S('map').width = '640px'
        S('map').height = '320px'
        var lat = position.coords.latitude
        var long = position.coords.longitude
        var gmap = O('map')
        var gopts =
        {
          center: new google.maps.LatLng(lat, long),
```



```
    zoom: 9, mapTypeId: google.maps.MapTypeId.ROADMAP
  }
  var map = new google.maps.Map(gmap, gopts)
}
function denied(error)
{
  var message
  switch(error.code)
  {
    case 1: message = 'Доступ запрещен'; break; case 2: message =
'Позиция недоступна'; break;
    case 3: message = 'Время ожидания операции истекло'; break;
    case 4: message = 'Неизвестная ошибка'; break;
  }
  O('status').innerHTML = message
} </script>
</body>
</html>
```

### **Назорат саволлари**

1. HTML хужжатни нечта турга ажратиш мумкин ?
2. HTML ва Cookies орасидаги боғланишни тушунтиринг?
3. Бошқарувчи тегларнинг вазифасини тушунтиринг?
4. Маълумот турини характерловчи теглар вазифасини тушунтиринг?
5. Рўйхатлар.
6. Ссилкалар.
7. Жадваллар.
8. Фреймлар.
9. Маълумотларни алмашиш формалари.
- 10.CSS селекторлар вазифалари
- 11.CSS синфлар вазифаси.

### **Фойдаланилган адабиётлар.**

1. VG Olifer, N. A.Olifer. Computer networks. Principles, technologies, protocols. St. Petersburg: Publishing House. "Peter," in 2000.
2. R. Darnell, "JavaScript: Handbook." - St. Petersburg: Publishing House. "Peter," in 1998.
3. Duvanov A.A. WEB. Construction of HTML. - St. Petersburg: BHV, 2003.
4. R. Eisenmenger. HTML 3.2/4.0. Handbook. - M: Univ. "Bean", 1998.
5. Stephen Speynaur, Valerie Kuersia. Directory WEB-master. - Ed. BHV
6. Barrett D. JavaScript. Web-professionals. - Kiev: BHV - Kyiv, 2001.
7. Brandenbau D. JavaScript: a collection of recipes. - St. Petersburg.: Peter, 2000.

# IV. БЎЛИМ

АМАЛИЙ МАШҒУЛОТ  
МАТЕРИАЛЛАРИ

## IV. АМАЛИЙ МАШҒУЛОТ МАТЕРИАЛЛАРИ

### 1 – амалий машғулот. HTML асослари.

**Ишдан мақсад:** HTML асосларини ўрганиш, унинг сарлавҳа ва тана қисмига тегишли асосий теглар билан танишиш.

Ушбу амалий иш давомида қуйидагиларни **бажариш лозим:**

- гиперматнли хужжат ташкил этиш асослари, манбаларни, HTML-хужжат ҳақида тушунчаларни;
- HTML-хужжат яратиш.

#### Ишни бажариш учун намуна

HTML (Hyper Text Markup Language) – белгили тил бўлиб, яъни бу тилда ёзилган код ўз ичига махсус рамзларни мужассамлаштиради. Бундай рамзлар хужжат кўринишини фақатгина бошқариб, ўзи эса кўринмайди. HTMLда бу рамзларни тэг (тэг – ёрлик, белги) деб аталади. HTMLда ҳамма теглар рамз-чегараловчилар (< , >) билан белгиланади. Улар орасига тэг идентификатори (номи, масалан B) ёки унинг атрибутлари ёзилади. Ягона истисно бу мураккаб чегараловчилар (<!--ва -->) ёрдамида белгиланувчи шархловчи теглардир. Аксарият теглар жуфти билан ишлатилади. Очувчи тегнинг жуфти ёпувчи тэг. Иккала жуфт тэг фақатгина ёпувчи тэг олдида «слэш» (“/”) белгиси қўйилишини ҳисобга олмаганда, деярли бир хил ёзилади. Жуфт тегларнинг асосий фарқи шундаки, ёпувчи тэг параметрлардан фойдаланмайди. Жуфт тэг яна контейнер деб ҳам аталади. Жуфт теглар орасига кирувчи барча элементлар тэг контейнери таркиби дейилади. Ёпувчи тегда зарур булмаган бир қатор теглар мавжуд. Баъзида ёпувчи теглар тушириб қолдирилса ҳам замонавий браузерлар аксарият ҳолларда хужжатни тўғри форматлайди, бироқ буни амалда қўллаш тавсия этилмайди. Масалан, расм қўйиш тэги <IMG>, кейинги қаторга ўтиш <BR>, база шрифтини кўрсатиш <BASEFONT> ва бошқалар ўзининг </IMG>, </BR> ва ҳоказо ёпувчи жуфтларисиз ёзилиши мумкин. Нотўғри ёзилган тегни ёки унинг параметри браузер томонидан рад килинади. (бу браузер танимайдиган тегларга ҳам тааллуқли). Масалан, <NOFRAME> тег-контейнери фақатгина фреймларни танийдиган браузер томонидан ҳисобга олинади. Уни танимайдиган браузер <NOFRAME> тегини тушунмайди.

Теглар параметр ва атрибутларга эга бўлиши мумкин. Параметрлар йиғиндиси ҳар-бир тегда индивидуалдир. Параметрлар қуйидаги коида асосида ёзилади:- Тэг номидан сўнг пробеллар билан ажратилган параметрлар келиши мумкин;- Параметрлар ихтиёрий тартибда келади;- Параметрлар ўзининг номидан кейин келувчи «=» белгиси орқали берилувчи қийматларга эга бўлиши мумкин. - Одатда параметрлар қиймати « » -

«кўштирноқ» ичида берилади. - Параметр қийматида баъзан ёзув регистри муҳим. Шунинг учун эса тутиш лозимки, ҳамма тэглар ўзининг индивидуал параметрига эга бўлишига карамай, шундай бир қатор параметрлар мавжудки, уларни <BODY> бўлимининг барча тэгларида ишлатиш мумкин. Бу параметрлар CLASS, ID, LANG, LANGUAGE, STYLE ва TITLE лардир.

HTML-хужжати ёзишни бошлашда ишлатиладиган биринчи тэг бу <HTML> тэгидир. У ҳар доим хужжат ёзувининг бошида бўлиши лозим. Яқунловчи тэг эса </HTML> шаклига эга бўлиши керак. Бу тэглар, улар орасида жойлашган ёзувнинг ҳаммаси бутун бир HTML-хужжати англатиши билдиради. Аслида эса хужжат оддий матнли ASCII-файлидир. Бу

```
<HTML>
  <HEAD>
  *
  * сарлавҳа қисми
  *
  </HEAD>
  <BODY>
  *
  * тана қисми
  *
  </BODY>
</HTML>
```

тэглрсиз браузер хужжати форматини аниқлаб, таржима қила олмайди. Кўпинча бу тэг параметрга эга эмас. HTML 4.0 версиясига қадар VERSION параметри мавжуд эди. HTML 4.0да эса VERSION ўрнига <!DOCTYPE> параметри пайдо бўлди.

<HTML> ва </HTML> орасида 2 бўлимдан ташкил топиши мумкин бўлган хужжатнинг ўзи жойлашади. Мазкур хужжатнинг биринчи бўлими сарлавҳалар бўлими (<HEAD> ва </HEAD>), иккинчи бўлим эса хужжат тана қисмидир (<BODY> ва </BODY>), уни хужжат танаси ҳам деб юритамиз. Фрейм тузилиши хужжатлар учун <BODY> бўлимининг ўрнига <FRAMESET> бўлимидан фойдаланилади. **Хужжатнинг HEAD бўлими.** HEAD бўлими сарлавҳа ҳисобланади ва у мажбурий тэг эмас, бироқ мукамал тузилган сарлавҳа жуда ҳам фойдали бўлиши мумкин. Сарлавҳа қисмининг мақсади хужжатни таржима қилаётган дастур учун мос ахборотни етказиб беришдан иборат. Хужжат номини кўрсатувчи <TITLE> тэгидан ташқари бу бўлимнинг қолган барча тэглари экранда акс эттирилмайди. Одатда <HEAD> тэги дарҳол <HTML> тэгидан кейин келади. <TITLE> тэги сарлавҳанинг тэгидир, ва хужжатга ном бериш учун хизмат қилади. Хужжат номи <TITLE> ва </TITLE> тэглр орасидаги матн қаторидан иборат. Бу ном браузер ойнасининг сарлавҳасида пайдо бўлади (бунда сарлавҳа номи 60 белгидан кўп бўлмаслиги лозим). Ўзгартирилмаган ҳолда бу матн хужжатга «закладка» (bookmark) берилганда ишлатилади. Хужжат номи унинг таркибини қисқача таърифлаши лозим. Бунда умумий маънога эга бўлган номлар (масалан, No page, Index ва бошқалар)ни ишлатмаслик лозим. Хужжат очилаётганда биринчи бўлиб унинг номи акс эттирилиши, сўнгра эса хужжат асосий таркиби кўп вақт олиб, кенгайиб кетиши мумкин бўлган форматлаш билан бирга юкланишини ҳисобга олган ҳолда, фойдаланувчи ҳеч булмаганда ушбу ахборот қаторини ўқий олиши учун хужжатнинг номи берилиши лозим.

**Хужжатнинг BODY бўлими** Ушбу бўлинма хужжатнинг таркибий қисмини ўз ичига олади. Бўлинма <BODY> тэгидан бошланиб </BODY> тэгида тугайди. Бироқ ушбу тэглр катъий мавжуд бўлиши шарт эмас, чунки браузерлар матнга қараб хужжат таркибий қисмининг ибтидосини аниқлаши мумкин. <BODY> тэгининг бир қатор параметрлари

мавжуд бўлиб, уларнинг бирортаси ҳам мажбурий эмас.

<BODY> тэги параметрлари:ALINK – фаол мурожаат (ссылка)нинг рангини белгилайди.BACKGROUND – фондаги тасвир сифатида фойдаланилувчи тасвирнинг URL-манзилини белгилайди. BOTTOMMARGIN – ҳужжатнинг куйи чегараларини пикселларда белгилайди.BGCOLOR – ҳужжат фонининг ранглари белгилайди.BGPROPERTIES – агар FIXED қиймати ўрнатилмаган бўлса, фон тасвири айланттирилмайди. LEFTMARGIN – чап чегараларни пикселларда белгилайди.LINK – хали кўриб чиқилмаган ссылканинг рангини белгилайди.RIGHTMARGIN – ҳужжат ўнг чегарасини пикселларда ўрнатади. SCROLL – браузер дарчалари харакатлантириш (прокрутка) йўлакларини ўрнатади. TEXT – матн рангини аниқлайди.

TOPMARGIN – юқори чегарасини пикселларда ўрнатади.VLINK – ишлатилган мурожаат рангини белгилайди. BOTTOMMARGIN, LEFTMARGIN, RIGHTMARGIN ва TOPMARGIN паметрлари матн чегараси ва дарча четлари орасидаги масофани пикселларда белгилайди. (Фақат HTML 4.0 версиясидан бошлаб IE браузерлари бу параметрларни таний олади).BGPROPERTIES параметри фақатгина битта FIXED қийматига эга. HTML даги ранглар ўн олтилик санок тизимида (RGB), ёки ранглар номи ёрдамида берилиши мумкин. Ранглар базаси 3 та рангга – қизил (R) , яшил (G) ва кўк (B) рангларга асосланган бўлиб, у RGB деб белгиланади. Ҳар-бир ранг учун 00 дан FF гача бўлган ўн олтилик санок тизимидаги қиймат берилади, бу эса 0 дан 255 гача бўлган диапазонга тўғри келади. Сўнгра бу қийматлар бир сонга бирлаштирилади ва уларнинг олдида “#” белгиси қуйилади. Масалан, #800080 сиёҳрангни билдиради.

**Мисоллар:**<BODY TEXT = “#000000”> ёки <BODY TEXT = black><BODY BGCOLOR = “#ffffff”> ёки <BODY BGCOLOR = WHITE><BODY LINK = “#ff0000”> ёки <BODY LINK = RED><BODY LINK = “#00FFFF” ALINK = “#800080”> ёки <BODY VLINK = Aqua ALINK = PURPLE>Ҳамма бараузлар ўн олтилик санок тизимидаги стандарт ранларни танийди. Булар куйидагилардир: Black = #000000 Maroon = #800000

Silver = #C0C0C0	Red = #FF0000	Grey = #808080
Purple = #800080	White = #FFFFFF	
Fuchsia = #FF00FF	Green = #008000	Navy = #000080
Lime = #00FF00	Blue = #0000FF	Olive = #808000
Teal = #008080	Yellow = #FFFF00	
Aqua = #00FFFF		

**Мисол:**<BODY BGCOLOR = “AQUA” TEXT = “#848484” LINK = RED VLINK = PURPLE ALINK = GREEN>Агар BGCOLOR параметри рангни номи ёки унинг таркибий қисмларини ўн олтилик санок тизимидаги кодда келтириш вазифаси ёрдамида фон рангини чиқариш учун ишлатилса, BACKGROUND тасвир ёрдамида саҳифага фон беришда фойдаланилади. Тасвир сифатида GIF ёки JPG форматдаги график файллар ишлатилади. HTML-ҳужжат фондаги тасвир доимо бутун саҳифани тўлдириб туради. Агар тасвир ўлчами дарча ўлчамидан кичик бўлса, у мозайка тамойилига асосан куйлаштирилади. Одатда фон тасвири сифатида тармоқ орқали юклаш

учун унча кўп вақт кетмайдиган кичик тасвир танлаб олинади, ёки фон сифатида шаффоф рельеф логотипи тасвиридан фойдаланилади. Мисол:

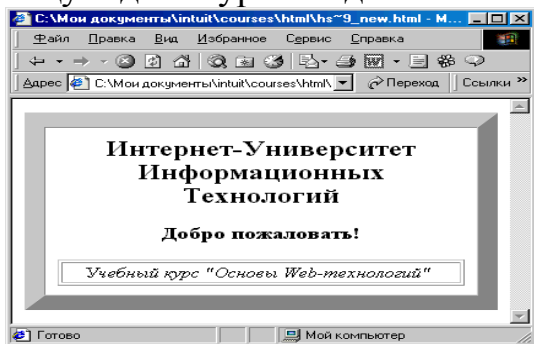
<BODY BACKGROUND = texture.gif BGCOLOR = gray>. Саҳифа яратилишида доимо фон рангини бериш тавсия қилинади. Агар фон тасвири ҳам берилаётган бўлса, фон ва тасвир ранглари бир-бирига яқин бўлгани маъқул.

**Мисол:**<BODY TEXT = BLUE LINK = RED VLINK = BLUE ALINK = PINKBACKGROUND= HYPERLINK "http://www.foo.com/jkorpela/HTML3.2/wave.gif">

**Мисол:**<BODY>  
<H1 ALIGN=center>Таблица</H1>  
<CENTER>  
<TABLE BORDER>  
<TR> <TD COLSPAN=3>Агарда таблицада иккита TR теги эълон қилинса, у ҳолда иккита қатор ҳосил бўлади.</TD> </TR>  
<TR> <TD>Агарда қаторда 2 та бўлса,</TD>  
<TD>у ҳолда унда</TD>  
<TD>3 та устун пайдо булади</TD>  
</TR>  
</TABLE>  
</CENTER>  
</BODY>  
</HTML>

## Назорат саволлари

### 1. қуйидаги кўринишдаги web-саҳифа яратиш



2. Турли ўлчамдаги шрифтлардан фойдаланган ҳолда матн ёзинг. Матн иккала чеккаси бўйича текисланган бўлиши керак.
3. 2 мисолдаги масалани қуйидаги тартибда ёзинг: биринчи сатр чап томонга текислансин, иккинчи сатр ўнг томонга ва ҳ.к.
4. Ўнта рўйхатдан иборат бўлган гиперсилкалар тузинг. Силкалар Интернетнинг турли дресларига мурожаат қилсин, жумладан қидирув тизимига.
5. Ўзингизнинг почта қутингизга ссылка яратинг.
6. 2 вариантдаги мисолни ҳар бир сўзини қандайдир хужжатга мурожаат қилувчи гиперсилкалардан ташкил этинг.
7. ихтиёрий расим олинг ва html-хужжат яратинг, расим устида матн мавжуд

бўлсин.

8. Учта элементдан иборат бетартиб рўйхат ташкил қилинг. Биринчи элемент айлана билан, иккинчиси квадрат учунчиси доира коринишидаги рўйхатлар бўлсин.

9. Қуйидаги жадвални ҳосил қилувчи HTML-код ёзинг.

1_1		1_2	1_3	1_4	
		2_1	2_2		
3_1	3_2	3_3		3_4	3_5
4_1	4_2			4_3	4_4
5_1		5_2	5_3	5_4	
		6_1	6_2		

10. HTML нинг асосий тэгларида фойдаланган ҳолда WEB саҳифа ташкил этинг.

#### **Фойдаланилган адабиётлар.**

1. VG Olifer, N. A.Olifer. Computer networks. Principles, technologies, protocols. St. Petersburg: Publishing House. "Peter," in 2000.
2. R. Darnell, "JavaScript: Handbook." - St. Petersburg: Publishing House. "Peter," in 1998.
3. Duvanov A.A. WEB. Construction of HTML. - St. Petersburg: BHV, 2003.
4. R. Eisenmenger. HTML 3.2/4.0. Handbook. - M: Univ. "Bean", 1998.
5. Stephen Speynaur, Valerie Kuersia. Directory WEB-master. - Ed. BHV
6. Barrett D. JavaScript. Web-professionals. - Kiev: BHV - Kyiv, 2001.
7. Brandenbau D. JavaScript: a collection of recipes. - St. Petersburg.: Peter, 2000.



## 2- амалий машғулот. HTML-хужжат ташкил қилишда объектлар, форма ва фреймлардан фойдаланиш.

**Ишдан мақсад:** HTML-хужжат ташкил қилишда объектлар, форма ва фреймлардан фойдаланиш.

Масаланинг кўйилиши: объект асослари, форма ва фреймлар.

**Қила олиши керак:** HTML-хужжат яратиш.

### Ишни бажариш учун намуна

#### HTML-формалар

**Формалар** WWW да фойдаланувчи томонидан киритилаётган маълумотларни тартибга солиш мақсадида қўлланилган. Форма элементлари тўлдирилиб бўлгач улардаги маълумотлар сервердаги маълумотларни қайта ишловчи дастурга юборилади. Кўп сонли жўнатилаётган маълумотлар жунатиш тугмаси босилгандан сўнг серверда жойлашган Common Gateway Interface (CGI) ёрдамида қайта ишланади. Шу тариқа фойдаланувчи Internet орқали Web-сервер билан биргаликда ишлайди.

#### Форманинг берилиши —FORM элементи

FORM элементи хужжатни маълум бир *формага* солади ва *форма* элементлари тэглари бошқа тэглardan ажратиб туради. <FORM> бир нечта <INPUT> тэглари кетма кетлигидан ташкил топади. Улар <FORM> ва </FORM> тэглари орасига жойлаштирилади. *Формада* усулдан (method), *формага* киритилган маълумотларни қайта ишлаш учун ҳолатлар (action) мавжуд. Усул (GET ёки POST) формага киритилган маълумотлар қай тарзда серверга жўнатилиш усулини белгиласа, ҳолат эса сервердаги қайси дастурга юборилиш URI (Uniform Resource Identifier) адресини ифодалайди.

<FORM METHOD=post ACTION=mailto:yourname@your.email.address>

#### Форманинг бошқарув элементлари —<INPUT> теги

Ушбу тэг *форманинг* қайси нуқтасига маълумот киритилишини белгилайди. У фойдаланувчи томонидан киритилаётган маълумотларни формага келтиради. Булар матн киритиш майдони, рўйхатлар, расмлар ёки тугмалар бўлиши мумкин. Майдон типи TYPE атрибути ёрдамида аниқланади.

#### TYPE=text атрибути

Агар фойдаланувчи унча катта бўлмаган матн киритса (бир ёки бир нечта сатр), <INPUT> тэгидан фойдаланади ва TYPE атрибутига text қиймати ўзлаштирилади. Стандарт ҳолат учун бу қийматни бериш муҳим эмас. Бундан ташқари *майдонни* номлаш ва унга мурожаат қилиш учун NAME атрибути ҳам берилади.

Сизнинг исмингиз <INPUT NAME=Name SIZE=35>

Фойдаланиш мумкин бўлган яна учта кўшимча атрибутлар мавжуд. Биринчиси MAXLENGTH деб аталади, у фойдаланувчи киритаётган матн майдони максимум узунлигини белгилайди. Стандарт бўйича бу қиймат чегараланмаган. Иккинчи атрибут SIZE ҳисобланади, у эса матн майдонини кўришиб турувчи қисмини белгилайди. Стандарт бўйича унинг қиймати браузерга боғлиқ бўлади. Агар MAXLENGTH қиймати SIZE қийматидан катта бўлса, браузер маълумотни ойнага мослаштиради. Сўнга кўшимча атрибут матн майдонини бошланғич қийматини белгиловчи VALUE дир.

### **TYPE=checkbox атрибути**

HTML формада мустақил белгилагич (байроқча) дан фойдаланиш учун <INPUT> тэгининг атрибутига TYPE=checkbox ни ўзлаштириш керак. Формага боғлиқ равишда фойдаланувчи бир ёки бир нечта белгилагичларни белгилаши мумкин. Агар <INPUT> тэги атрибути билан CHECKBOX қиймати қўлланилса, у билан бирга NAME ва VALUE атрибутлари ҳам қўлланилиши керак. NAME атрибути ушбу маълумот киритиш объектининг номини ифодалайди. VALUE атрибутида ушбу майдоннинг қиймати кўрсатилади.

<BR>Россия<INPUT NAME="Давлат" TYPE=checkbox  
VALUE="Россия">

Страны СНГ<INPUT NAME="Давлат" TYPE=checkbox  
VALUE="СНГ">

Баъзи ҳолларда ушбу майдон белгиланган ҳолда қўлланилиши ҳам мумкин. Бундай ҳолларда <INPUT> тэгида CHECKED атрибути қўлланилиши керак.

### **TYPE=radio атрибути**

Баъзан бир нечта қийматлар орасидан бирини танлашга тўғри келади. Бундай ҳолларда формада <INPUT> тэги билан бирга TYPE=radio атрибути қўлланилади. Агар <INPUT> тэги атрибути билан ушбу қиймати қўлланилса, у билан бирга NAME ва VALUE атрибутлари ҳам қўлланилиши керак. NAME атрибути ушбу маълумот киритиш объектининг (тугма) номини ифодалайди. VALUE атрибутида ушбу майдоннинг қиймати кўрсатилади..

<BR>Эркак жинси

<INPUT NAME="Жинс" TYPE=radio VALUE="Эркак">

Аёл жинси <INPUT NAME="Жинс" TYPE=radio VALUE="Аёл">

### **TYPE=image атрибути**

Форманинг таркибига қараб баъзан унда жойлашган расмнинг устига сичқончани босиш билан ундаги маълумотларни жўнатишга тўғри келиб қолади. Бунинг учун <INPUT> тэги TYPE=image атрибути билан қўлланилади. Фойдаланувчи расм устига сичқонча курсорини босса, айнан

шу ердаги экран координаталарини браузер сақлаб қолади. Сўнг формага киритилган маълумотларни “қайта ишлайди”. Агар `<INPUT>` тэги `image` атрибути билан қўлланилса, у билан бирга `NAME` ва `SRC` атрибутлари ҳам қўлланилиши керак. `NAME` майдоннинг номини белгилайди. `SRC` атрибути эса расм жойлашган манбанинг `URI` манзилени беради. `ALIGN` атрибути кўшимча ҳисобланади ва у ҳам баъзан `<IMG>` теги билан қўлланилади.

`<BR>Нуқтани танланг <INPUT TYPE=image NAME=point SRC=image.gif>`

### **TYPE=password атрибути**

Агар *формада* пароллардан фойдаланиш керак бўлиб қолса, `TYPE` атрибути қийматига `password` (`TYPE=password`) ни ўзлаштирилади. Ушбу типдан фойдаланиш киритилаётган маълумотни ошкор бўлмаган ҳолда кўрсатишни таъмин этади. Шу сабаб, киритилган маълумот очиқ канал орқали жўнатилади ва ушбу маълумот тутиб олиниши мумкин.

`<BR>Номингиз<INPUT NAME=login>Парол  
<INPUT TYPE=password NAME="Сўз">`

### **TYPE=reset атрибути**

Базан фойдаланувчи *формани* тўлдириш вақтида, уларни бошдан тўлдиришга тўғри келади. Ушбу ҳолда `Reset` тугмаси мавжуд бўлиб, бу тугманинг босилиши *формани* дастлабки, кириш ҳолатиги олиб келади (*формани* “тозалайди”). `Reset` иугмасини ташкил қилиш учун `<INPUT>` тэги атрибутига `TYPE=reset` ўзлаштирилади. Агар *формада* `reset` атрибути қўлланилса, `<INPUT>` тэгига `VALUE` атрибутини кўшимча қилиш мумкин. Ушбу атрибут тугмадаги ёзувни ифодалайди.

`<INPUT TYPE=reset VALUE="Формани тозалаш ">`

### **TYPE=submit атрибути**

`HTML форма` да фойдаланувчи маълумот киритиш жараёнини яқунлаш жараёни мавжуд. Бунинг учун `<INPUT>` тэгининг атрибутига `TYPE=submit` қиймат ўзлаштирилади. Агар *формада* `<INPUT>` тэги `submit` атрибути билан қўлланилса, унга кўшимча равишда иккита атрибутдан фойдаланиш мумкин: `NAME` ва `VALUE`. `NAME` атрибути *майдоннинг* номини ифодалайди. `VALUE` атрибути — `Submit` тугмаси матнини кўрсатади.

`<BR><INPUT TYPE=submit VALUE="Хабарни жўнатиш ">`

### **TYPE=hidden атрибути**

Яширин *майдон*. `INPUT` тэгини `TYPE=hidden` атрибути билан қўлланилиши фойдаланувчига маълум бўлмаган `NAME` ва `VALUE` атрибутларидаги қийматларни жўнатишга имкон беради.

**<TEXTAREA>** – кўп сатрли матн киритишни ташкил этиш тэги

Баъзан *формада* кўп сатрли матнларни киритиш талаб этилади. Бунинг

учун **<TEXTAREA>** тэги ёрдамида бир неча сатрдан иборат бўлган матн майдони ташкил этиш мумкин. Ушбу тэг учта атрибут билан ишлатилади: COLS, NAME ва ROWS.

### **Атрибут COLS**

Майдоннинг устунлари (белгилар сони) сонини белгилайди.

### **Атрибут NAME**

Майдоннинг номини белгилайди.

### **Атрибут ROWS**

Майдоннинг кўринувчи сатрлари сонини белгилайди.

**<BR><TEXTAREA NAME=мавзу COLS=38 ROWS=3> </TEXTAREA>**

### **<SELECT>- формада рўйхатдан фойдаланиш тэги**

Агарда *форма* мукамал бўлса, гоҳида унда ҳаракатланувчи рўйхат ҳам қўлланилади. Бунинг учун **SELECT** тэгидан фойдаланилади. Рўйхат бўлимларини аниқлаш учун **<OPTION>** тэгидан фойдаланилади. **<SELECT>** тэги муҳим бўлмаган учта атрибутни қўлаб қувватлайди: MULTIPLE, NAME ва SIZE.

### **MULTIPLE атрибути**

Бир вақтнинг ўзида бир неча вариантни танлаш имконини беради.

### **NAME атрибути**

Объект номини ифодалайди.

### **SIZE атрибути**

Рўйхатни кўринувчи сатрлари сонини ифодалайди. SIZE > 1 бўлган ҳолда браузер оддий рўйхатни кўрсатади.

*Формада <OPTION>* теги фақат **<SELECT>** тэглари орасида қўлланилади. Бу тэглар кўшимча иккита атрибутни қўлаб қувватлайди: SELECTED ва VALUE.

### **SELECTED атрибути**

Дастлаки ҳолатда ушбу элемент танланган эканлигини билдиради.

### **VALUE атрибути**

Рўйхатга ўзлаштирилиши мумкин бўлган қийматни ифодалайди.

**<BR>Танлаш**

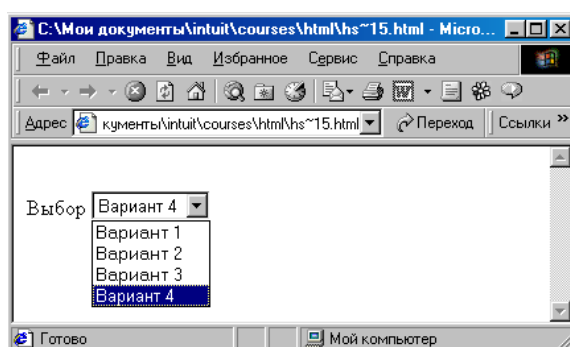
**<SELECT NAME="Танлаш">**

**<OPTION>Вариант 1**

**<OPTION>Вариант 2**

**<OPTION VALUE="Вариант 3">Вариант 3**

<OPTION SELECTED>Вариант 4  
</SELECT>



Расм 2.1. Қалқиб чикувчи меню

## Фреймлар

Фреймлар барузерни кузатув ойнасини ёнма-ён жойлашган бир нечта тўғри бурчакли соҳаларга бўлиш имконини беради. Мазкур бўлаклардан ҳар бирига алоҳида HTML-файл, яъни бошқалардан мустақил равишда кўздан кечирилувчи файлларни юклаш мумкин. Зарурият туғулганда фреймлар орасида ўзаро боғлиқликни ташкил этиш мумкин. Ўзаро боғлиқлик ташкил этилганда фреймлардан бирида ссылка танланса, бошқа фрейм ойнасида керакли ҳужжатнинг юкланишига олиб келади.

Ҳарчи HTML-ҳужжатларда фойдаланувчига ахборот акс эттирилишининг турли усуллари ҳавола этилсада, ахборотни ифодалашнинг фрейм тизими ҳам ўзининг афзалликларига эга. Қуйидаги ҳолларда айнан фрейм тизими қўл келади:

- Бир соҳада ишлаётганда бошқа бир соҳага ҳужжатларни юклаш орқали бошқаришни ташкил этиш зарурати туғулганда;
- Экраннинг бошқа ҳудудларида нима бўлишидан қатъий назар экранда доимо кўриниб туриши керак бўлган ахборотни кўздан кечириш дарчасининг маълум қисмига жойлаштириш лозим бўлганда;
- Дарчанинг ҳар бири мустақил равишда кўриб чиқилиши мумкин бўлган ёнма-ён бир неча соҳаларида жойлаштириш қўлай бўлган ахборотни тақдим этиш зарурати туғулганда.

Фреймлар тизимини тасвирлаш учун <FRAMESET>, <FRAME> ёки <NOFRAME> тэгларидан фойдаланилади.

<FRAMESET> тэги фреймларни белгилайди.

Фреймлардан ташкил топган Web-саҳифалар <BODY> бўлинмасига эга бўлиши мумкин эмас.

<FRAMESET> ва </FRAMESET> контейнерлари ҳар бир фреймни белгилаш блокини ўраб туради. Бундай контейнернинг ичида фақат <FRAME> тэглари ёки киритилган <FRAMESET> тэглари мавжуд бўлади.

<FRAMESET> тэгининг атрибутлари:

- ROWS;
- COLS.

Ушбу параметрлар қийматлари пикселларда, фоизларда ёки нисбий бирликларда берилиши мумкин. Қатор ёки устунлар сони мос рўйхатдаги қийматлар сони билан аниқланади. **Масалан:**

<FRAMESET ROWS = “100, 240, 140”> - учта фреймдан иборат тўпламни белгилайди. Қийматлар пикселларда берилган. Биринчи фрейм 100 пиксел, иккинчиси 240 пиксел ва ниҳоят сўнгги фрейм 140 пиксел баландликка эга.

<FRAMESET ROWS = “25%, 50%, 25%”> -

экраннынг мақбул баландлигидан юқори қаторнинг қиймати 25 фоиз, ўрта қаторники 50 фоиз, қуйи қаторники 25 фоиз эканлигини билдиради.

<FRAMESET COLS = “\*, 2\*, 3\*”> - қийматлар нисбий бирликларда.

“Юлдузча” – “\*” фазони пропорционал тақсимлаш учун ишлатилади. Хар бир юлдузча бутуннинг бир қисмини билдиради. Ҳисоблаб топиш учун юлдузчалар олдидаги сонларни қўшиш ва ҳосил бўлган сондан касрнинг махражи сифатида фойдаланилади. Юқоридаги мисолда биринчи устун дарча умумий кенглигининг 1/6, иккинчи устун 2/6, учинчи устун 3/6 қисмини эгаллайди.

<FRAMESET COLS = “100, 25%, \*, 2\*”>.

Ушбу мисолда қийматларни белгилашнинг 3 та усулидан ҳам фойдаланилган. Ҳам ROWS, ҳам COLS атрибутлари биргаликда ишлатилганда фреймлар сеткаси яратилади:

<FRAMESET COLS = “2\*,\*”, ROWS = “\*, 2\*”>.

<FRAME> тэги алоҳида файлларни белгилайди, бу тэг <FRAMESET> ва </FRAMESET> тэглари жуфтлигининг ичида жойлашиши лозим. Масалан:

<FRAMESET ROWS = “\*, 2\*”>

<FRAME>

</FRAME>

</FRAMESET>

<FRAMESET> тэги берилганида қанча алоҳида фреймлар белгиланган бўлса, шунча фрейм тэглари ёзиш лозим.

<FRAME> тэги атрибутлари:

SRC

NAME

MARGINWIDTH

MARGINHEIGHT

SCROLLING

NORESIZE

FRAMEBORDER=YES/NO (Фақат IE лар учун)

SRC атрибути бошидан бошлаб мазкур фреймга юкланувчи ҳужжатнинг URL-манзилини белгилайди. Одатда бундай манзил сифатида асосий ҳужжат қайси каталогда бўлса, уша ерда жойлашган HTML-файлнинг номидан фойдаланилади. Масалан:

<FRAMESET SRC=“sample.html”>

Зеро, фреймни тасвирлашда берилган HTML-файл тўлиқ HTML-

хужжат бўлиши керак, яъни у HTML, HEAD, BODY ва бошқаларга эга бўлиши лозим. Агар фреймдан тасвирни акс эттиришда фойдаланилса, унда:  
<FRAME SRC="http://www.bhv.ru/exampl.gif">

NAME параметри берилган фреймга ссылка сифатида ишлатиш мумкин бўлган фреймнинг номи белгилайди. Масалан:

<FRAME SRC="sample.html" NAME="frame1">

frame1 деб номланган ушбу фреймга ссылка қилиниши мумкин.

**Масалан:** <A HREF="other.html" TARGET="frame1">frame1 </A> фреймига other.html файлини юклаш учун шу ерга сичқонча курсори босилади.

MARGINWIDTH ва MARGINHEIGHT атрибути фрейм хошия(чегара) кенглигини белгилайди.

Атрибутлар қийматлари пикселларда берилади.

**Масалан:** <FRAME MARGINWIDTH = "5" MARGINHEIGHT = "7">

Бу ерда фрейм юқори ва пастда 5 пиксел, ўнг ва чап томонларидан эса 7 пиксел чегарага эга. Ишлатилиши мумкин бўлган энг кичик қиймат 1 пикселдир.

SCROLLING атрибутидан **прокрутка** йўлакларини акс эттиришни бошқаришда фойдаланилади. Унинг синтаксиси

<FRAME SCROLLING= "YES/ NO/ AVTO"> кўринишга эга.

NORESIZE атрибути фойдаланувчи томонидан фрейм ўлчами ўзгартирилишининг олдини олишда ишлатилади. **Масалан:** <FRAME NORESIZE>.

Табиийки NORESIZE атрибутининг битта фреймга нисбатан қўлланилиши бошқа фреймлар ўлчами ўзгартирилишининг ҳам олди олинишига сабаб бўлади.

Гарчи фреймлар тизими HTML 4.0да стандарт билан мустахкамланган бўлсада, <NOFRAMES> тэги фреймларни қўллаб-қувватламайдиган браузерлар ёрдамида кўздан кечиришда асқотади. Демак, фреймларга боғланмаган браузерлар учун <NOFRAMES>ва </NOFRAMES> тэглари жуфтлигидан фойдаланилади. Масалан:

<NOFRAMES> бутун HTML-хужжат </NOFRAMES>

Мазкур тэглар орасига жойлаштирилган барча маълумотлар фреймларни қўллаб-қувватлаш имкониятига эга бўлмаган браузерлар ёрдамида акс эттирилади. Фреймларга боғланган браузерлар эса <NOFRAMES> ва </NOFRAMES> орасидаги барча ахборотга боғлиқ емас. Юқорида келтирилган параметрлар ишлатилган мисолларни кўриб чикамиз.

### **Фреймлар орасидаги ўзаро таъсир**

Фреймлар билан ишлаётганда фойдаланувчи учун қўлай бўлган хужжат юклаш схемасини яратиш мумкин. Фреймлар орасидаги ўзаро алоқа хужжатларни бошқа фреймдаги буйруқлар ёрдамида айнан танланган фреймга юклаш имконини беришидир. Бу мақсадда <A> тэгининг TARGET атрибутидан фойдаланилади. TARGET атрибути ушбу ссылка кўрсатаётган хужжат юкланувчи фрейм ёки браузер ойнаси номи белгилайди. Ўзгартирилмаган ҳолда ушбу параметр йўқ бўлганда хужжат жорий фрейм ёки ойнада юкланади. Фрейм номи сифатида мавжуд дарча ёки

фрейм номи берилиши ёки бўлмаса янги ойна очиш учун янги ном берилиши мумкин. 4 та захирадаги номлар бор. Улар:

\_blank, \_self, \_top, \_parent. Булардан ташқари “\_” белгиси билан бошланувчи хар қандай номдан фойдаланиш мақсадга мувофиқ эмас.

TARGET=“\_blank” – ҳужжатнинг янги ойнага юкланишини таъминлайди. Бу ойна номга эга бўлмаслиги туфайли унга бошқа ҳужжатни юклашнинг иложи бўлмайди.

TARGET=“\_self” дан фойдаланилганда ҳужжат жорий фреймга юкланади.

TARGET=“\_top” ҳужжатнинг бутун дарчага юкланишига сабаб бўлади.

TARGET=“\_parent” ҳужжатнинг жорий фреймнинг фрейм-ота-онаси томонидан эгалланган соҳасига юкланишига олиб келади.

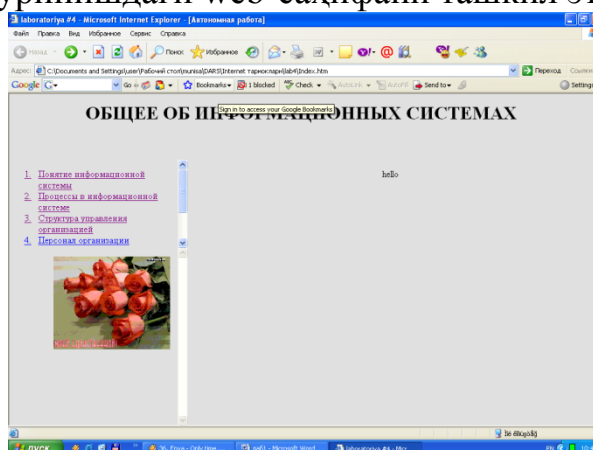
Таркиби “А” фреймига юкланган frame\_a.html файлининг биттагина test.html файлига TARGET параметрининг турли қийматларига эга 6 та ссылкаси мавжуд.

## Назорат саволлари

### 1. Қуйидаги кўринишдаги web-саҳифани ташкил этинг

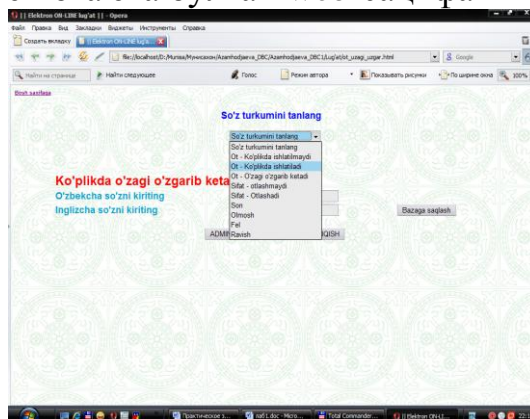


### 2. Қуйидаги кўринишдаги web-саҳифани ташкил этинг





### 3. Қуйидагича менюга эга бўлган web-саҳифани ташкил этинг



### 4. Қуйидаги кўринишдаги формани ташкил этинг

Введите персональные данные:

Имя

Фамилия

День рождения

Ваш пол  Мужской  Женский

Ваш e-mail

### Фойдаланилган адабиётлар.

1. VG Olifer, N. A.Olifer. Computer networks. Principles, technologies, protocols. St. Petersburg: Publishing House. "Peter," in 2000.
2. R. Darnell, "JavaScript: Handbook." - St. Petersburg: Publishing House. "Peter," in 1998.
3. Duvanov A.A. WEB. Construction of HTML. - St. Petersburg: BHV, 2003.
4. R. Eisenmenger. HTML 3.2/4.0. Handbook. - M: Univ. "Bean", 1998.
5. Stephen Speynaur, Valerie Kuersia. Directory WEB-master. - Ed. BHV
6. Barrett D. JavaScript. Web-professionals. - Kiev: BHV - Kyiv, 2001.
7. Brandenbau D. JavaScript: a collection of recipes. - St. Petersburg.: Peter, 2000.

### **3 – амалий машғулот. JavaScript. Клиент томонида дастурлаш. JavaScript ни HTML-хужжатларга жойлаштириш. JavaScript да маълумотлар типлари, ўзгарувчилар, ифодалар ва арифметик ифодалардан фойдаланиш.**

**Ишдан мақсад:** JavaScript ни ўрганиш, клиент томонидаги дастурлар яратиш, JavaScript ни HTML-хужжатга жойлаштириш, JavaScript да маълумотлар типларидан фойдаланиш, ўзгарувчилар, амаллар ва арифметик операторлар билан ишлаш.

Масаланинг қўйилиши:

**Билиши керак:** JavaScript нинг асосий катталиклари;

**Қила олиши керак:** HTML-хужжатда JavaScript ёрдамида клиент томонидаги сценарийлар яратиш.

#### **Ишни бажариш учун намуна**

JavaScript тилида ўзгарувчиларни ишлатиш мумкин ва уларни номлари билан адреслаш мумкин. Ўзгарувчилар глобалли ва локалли бўлиши мумкин. Глобалли ўзгарувчилар сценарийнинг хохлаган жойида рухсати бўлиши мумкин. Локалли ўзгарувчиларнинг ҳаракати эса эълон қилинган ўзгарувчилар ичидаги функциялар билан чегараланган. Basic дастурлаш тили сингари JavaScript сценарийсини яратаётган вақтда аввалдан эълон қилинмаган ўзгарувчиларни ишлатиш мумкин.

#### **Ўзгарувчилар эълони**

JavaScript да ҳамма ўзгарувчилар var калит сўзи орқали эълон қилинади ва қуйидагича кўрсатилган:

```
var MyHelloMsg;
```

Ўзгарувчи типи ўзлаштириладиги қачонки, унга бирор бир қиймат ўзлаштирилса, қуйида аввалдан эълон қилинмаган матнли қатор ўзгарувчига ёзилмоқда:

```
MyMsg = "Салом!";
```

MyMsg ўзгарувчи номи ўзлаштирилгандан сўнг рухсат берилади.

Ўзгарувчи номини танлаганда, ыуйидаги оддий ыоидаларни ушлаб ыщйиш керак:

- Ўзгарувчи номи ҳарфлардан ёки "\_", "\$" белгилардан бошланиш керак ва фақат ҳарфлардан, сонлардан ва "\_", "\$" белгилардан иборат бўлиши керак;
- Ўзгарувчилар номи JavaScript нинг захираланган калит сўзлари билар мос келмаслиги керак.

Қуйида JavaScript нинг захираланган калит сўзлар келтирилган:

```
break case catch class const continue  
debugger default delete do else enum
```

export extends false finally for function  
if import in new null return  
super switch this throw true try  
typeof var void while with

Бу сўзлар орасида JavaScript тилида ва унинг ривожланишида ўзлаштириш режалаштирилмоқда.

### **Ўзгарувчининг қийматини ўзлаштириш**

"=" ўзлаштириш оператори ёрдамида ўзгарувчилар қиймати ўзлаштирилади. Мисол қилиб ўуйидаги ўзгарувчи келтирилган ва унда матнли қатор ёзилган:

```
var MyHelloMsg;  
MyHelloMsg = "Hello, world!";
```

MyHelloMsg сонли ўзгарувчини дастурнинг хохлаган жойида ўзлаштириш мумкин, мисол учун:

```
MyHelloMsg = 4;
```

Бу оператор бажарилгандан сўнг ўзгарувчи типи ўзгаради, шунингдек интерпретация жараёнида браузер ҳеч қандай огохлантирувчи хабарларни юбормайди.

Ўзгарувчини махсус null қиймати орқали ўзлаштириш мумкин:

```
MyHelloMsg = null;
```

Бундай ўзлаштириш ҳеч қандай типда ўзгарувчини белгиламайди.

### **JavaScript да маълумотлар типи**

JavaScript тилида бир нечта маълумотлар типи мавжуж. Булар сонлар, матнли қаторлар, мантиқий маълумотлар, объектлар, аниқланмаган типли маълумотлар, ҳамда махсус тип null.

#### **Сонлар**

JavaScript тили ҳар хил форматдаги сонларни ишлатишга рухсат беради, булар бутун сонлар, сузувчи нуқтали ўнли форматдаги сонлар ва илмий нотация сонлар. Бутун сонлар 8, 10, 16 асосида берилиши мумкин. мисол учун:

```
25      10 асосидаги бутун сон  
0137    8 асосидаги бутун сон  
0xFF    16 асосидаги бутун сон  
386.7   Сузувчи ўнли нуқтали сон  
25e5
```

или 25E5 Илмий нотациядаги сон, 2500000 га тенг.

Айрим ҳолларда "сон бўлмаган" арифметик функциялар келиб чиқиши мумки. JavaScript да айтилганидек NaN (Not a Number). "Сон бўлмаган" – бу ҳеч қандай сонга лойиқ бўлмаган махсус қиймат. Бу сонлар устида операция бажарилаётган вақтда, ва натижа сон кўринишида тақдим этилмаган ҳолларда пайдо бўлади. "Сон бўлмаган" қийматга тўғри келишини isNaN функцияси ёрдамида текшириш мумкин.

## Матнли қатор

Матнли қатор – бу бир ёки қўштирноқ кетма кетлик белгиси, мисол учун:

```
"Hello, world!"
```

```
""
```

```
"12345"
```

```
'Бу матнли қатор'
```

"" қатори –бўшдир. Қуйидаги 2 та ўзлаштириш эквивалент эмаслигини аниқлаймиз:

```
MyStr=""
```

```
MyStr1=null
```

Биринчи холда MyStr ўзгарувчисида матнли қатор сақланмоқда (бўш бўлса ҳам), иккинчисида эса ҳеч нарса.

## Мантиқий маълумотлар

Мантиқий маълумотлар фақат 2 та қийматни, яъни True ва False ни ўз ичига олади. Бу қийматлар 0 ва 1 сонлар билан боғлиқ эмас. Бу қийматларнинг асосий образи солиштириш операцияси бажарилаётган вақтга қаратилган, ҳамда шартли операциялар ишлатилганда ҳам.

## Аниқланмаган типли маълумотлар.

Агар ўзгарувчи эълон қилинган бўлса, аммо унга ҳали қиймат ўзлаштирилмаган бўлса, у холда у аниқланмаган типга бўлади. Мисол учун қуйидаги қаторда аниқланмаган типга эга бўлган MyVariable ўзгарувчиси эълон қилинган:

```
var MyVariable;
```

Агарда бу ўзгарувчини null қиймати билан ўзгартирсак, у холда ўзгарувчи типи ўзгаради ва null қийматга эга бўлган ўзгарувчига айланади:

```
MyVariable = null;
```

## JavaScript тили операторлари

### Унар оператори

Унар оператори белгининг ўзгариши учун тўлдириш операциясини бажаришда, инкрементда ҳамда декрементда ишлатилади:

- тескари холатда белгининг ўзгариши

! Қушимча. Мантиқий ўзгарувчиларнинг қийматини реверсирования қилиш учун ишлатилади.

++ Ўзгарувчи қийматини ошириш. Ўзгарувчи префикси ёки унинг суффикси бўлиб қўлланиши мумкин.

-- Ўзгарувчи қийматини камайтириш. Ўзгарувчи префикси ёки унинг суффикси бўлиб қўлланиши мумкин.

Унар операторини ишлатишга доир мисоллар:

```
i=0; // i тенг 0 даги ўзгарувчининг бошланғич қиймати
```

```
i++; // i тенг 1 даги қиймат
--i; // i тенг 0 даги қиймати
```

```
var j=3; // j тенг 3 даги ўзгарувчининг қиймати
i = -j; // i тенг -3 даги ўзгарувчининг қиймати
```

```
var fYes = true; // fYes тенг true даги ўзгарувчининг қиймати
testFlag(!fYes); // testFlag функциясига false қиймати узатилмоқда
```

### **Бинар оператори**

Бинар оператори 2 та операндни бирлаштиради. JavaScript тилида бинар операторлари айириш, бўлиш, кўшиш, кўпайтириш ҳамда бўлинмани қолдиғини ҳисоблаш учун ишлатилади (кўрилади):

- Айириш
- + Кўшиш
- \* Кўпайтириш
- / Бўлиш
- % Бўлинмани қолдиғини ҳисоблаш

Бу операторлар C тилида ишлатилганидек JavaScript да ҳам худди шундай ишлатилади, мисол учун:

```
i=0; // i тенг 0 даги ўзгарувчининг қиймати
i = i + 1; // i тенг 1 даги қиймат
```

```
var j=9; // j тенг 9 даги ўзгарувчининг қиймати
i = j / 2; // i тенг 4 даги ўзгарувчининг қиймати
k = j % 2; // i тенг 1 даги ўзгарувчининг қиймати
```

### **Алохида битлар билан ишлаш оператори**

Сценарияларда шундай операторлар ишлатиладики, улар алохида битлар билан ишлаш операторлари ҳисобланади, улар қуйидагилар: И, ИЛИ, ИСКЛЮЧАЮЩЕЕ ИЛИ, НЕ:

- & И
- | ИЛИ
- ^ ИСКЛЮЧАЮЩЕЕ ИЛИ
- ~ НЕ

### **Силжувчи операторлари**

JavaScript да силжиш операциясини бажариш учун 3 та оператор қўрилган:

- >> Силжиш ўнг томонга
- << Силжиш чап томонга
- >>> Бўшатиладиган разрядларни ноллар билан тўлдириб ўнгга силжиш

### **Мунособат операторлари**

Мунособат операторлари ўзгарувчиларнинг қийматини солиштириш

учун ишлатилади. Бу операторлар солиштириш натижаларига боғлиқлик true ёки false мантиқий қийматларни қайтаради ва шартли операторларда асосий бўлиб ишлатилади. True қийматини қайтарадиган JavaScript тилининг мунособат операторлари кўрсатилган:

- > Чап операнд ўнг операнддан катта
- >= Чап операнд ўнг операнддан катта ёки тенг
- < Чап операнд ўнг операнддан кичик
- <= Чап операнд ўнг операнддан кичик ёки тенг
- == Чап операнд ўнг операндга тенг
- != Чап операнд ўнг операндга тенг эмас

### Мантиқий операторлар

|| ИЛИ оператори. True қиймат қайтаради, қачонки операндлардан бири true бўлса.

&& И оператори. True қиймат қайтаради, қачонки икки операнд true бўлса

### Ўзлаштириш оператори

Ўзлаштириш оператори ўзгарувчиларнинг қийматини ўзлаштириш учун ишлатилади. JavaScript тилида ва C дастурлаш тилидаги каби бу оператор бошқа операторлар билан комбинациясига рухсат этилади. Қуйида ўзлаштириш операторини бошқа операторлар билан комбинацияси берилган:

- = Оддий ўзлаштириш
- += Сонли қийматни катталаштириш ёки қаторларни қўшилиши
- = Сонли қийматни кичиклаштириш
- \*= Кўпайтириш
- /= Бўлиш
- %= Бўлишдан қолган қолдиқни ҳисоблаш
- >>= Ўнгга силжиш
- >>>= Бўшатиладиган разрядларни ноллар билан тўлдириб ўнгга силжиш
- <<= Чапга силжиш
- |= ИЛИ
- &= И
- ^= ИСКЛЮЧАЮЩЕЕ ИЛИ

C тили билан таниш бўлмаганлар учун ўзлаштириш операторини бошқа операторлар билан биргаликда ишлатилиши қийинроқ ва ғайриоддий туйилиш мумкин, лекин аслида сценарийни осонлаштирадибошланғич текстни соддалаштиради.

Масалан сонли ўзгарувчилар қийматини ошириш учун += оператори ишлатилади. Аввал бу вазифани ечимини += операторини ишлатмаган ҳолатда кўриб чиқамиз. Қуйида nCounter ўзгарувчиси эълон қилинди ва унга бошланғич 1 қиймати ўзлаштирилди, сўнг бу қиймат 5 га оширилди:

- ```
var nCounter = 1;  
nCounter = nCounter + 5;  
Энди буни += оператори ёрдамида бажарамиз:
```

```
var nCounter = 1;
```

```
nCounter += 5;
```

Кўриниб турибдики 2-усул 1-усулга нисбатан қисқа.

Ўзгарувчи қийматини 3 разрядга ўнгга силжитиш учун `>>=` операторидан фойдаланиш мумкин ва у қуйидаги матнда кўрсатилган:

```
nCounter >>= 3;
```

Натижа эса қуйидаги матнда кўрсатилганидек бўлади:

```
nCounter = nCounter >> 3;
```

### JavaScript тилида функция

Бошланғич матн бўлагини функция кўринишида ёзиш мумкин ва уларни **JavaScript** сценарийсининг турли жойларидан мурожаат қилиш мумкин. Одатда функциялар HTML документини сарлавха бўлимида аниқланади. Функциялар чақирилишидан аввал эълон қилиниши керак ва барча функция эълони HTML документ сарлавхасида жойлаштирилган бўлиши керак.

Функциянинг умумий эълони қуйида келтирилган:

```
function имя([параметр 1] [,параметр 2] [...,параметр N])
```

```
{ ...
```

```
  Функция матни қаторлари
```

```
  ...
```

```
  [return қиймат]
```

```
}
```

Барча параметрлар функцияга қийматига берилади. Шунинг учун функция унга параметр сифатида бериладиган ўзгарувчилар қийматини ўзгартира олмайди.

Return калит сўзи ёрдамида функция қиймати қайтарилади.

### Назорат саволлари

1. Қуйидаги кўринишда шарт оператори таркибини тузинг:  
(шарт)? амал1:амал2

Бунда қуйидаги операторлардан фойдаланинг

A) Арифметик амаллар

B) Ўзлаштириш

C) Инкремент

D) Декремент

2. Ҳисобланг

a)  $12 \& 9 =$  ; b)  $13 \& 14 =$ ; c)  $10 \& 4 =$

d)  $14 | 10 =$  e)  $16 \wedge 3 =$  f)  $14 \& 56 | 11 =$

### **Фойдаланилган адабиётлар.**

1. VG Olifer, N. A.Olifer. Computer networks. Principles, technologies, protocols. St. Petersburg: Publishing House. "Peter," in 2000.
2. R. Darnell, "JavaScript: Handbook." - St. Petersburg: Publishing House. "Peter," in 1998.
3. Duvanov A.A. WEB. Construction of HTML. - St. Petersburg: BHV, 2003.
4. R. Eisenmenger. HTML 3.2/4.0. Handbook. - M: Univ. "Bean", 1998.
5. Stephen Speynaur, Valerie Kuersia. Directory WEB-master. - Ed. BHV
6. Barrett D. JavaScript. Web-professionals. - Kiev: BHV - Kyiv, 2001.
7. Brandenbau D. JavaScript: a collection of recipes. - St. Petersburg.: Peter, 2000.



#### 4 – амалий машғулот. JavaScript да дастурни бошқариш элементларидан фойдаланиш. Шарт, цикл, танлаш ва бошқа операторлар. Функция ва усуллар яратиш. Объектлар.

**Ишдан мақсад:** JavaScript нинг бошқариш элементларини билиш. Шарт, цикл, танлаш ва бошқа операторлар билан ишлаш. Функция ва усуллар яратиш. Объектларни ўрганиш.

Масаланинг қўйилиши:

**Билиши керак:** JavaScript нинг бошқариш элементларини;

**Қила олиши керак:** HTML-хужжатда JavaScript ёрдамида бошқариш элементларидан фойдаланиб илова яратиш. Функция, усуллар ва объектлар яратиш

#### Ишни бажариш учун намуна

JavaScript сценарийли тили объектга-мўлжалланган тилдир. JavaScript объектлари хусусиятлар ва усуллар тўпламини ифодалайди. Объект хусусияти – бу, объектга боғлиқ бўлган маълумотлардир, усуллар эса - объект маълумотларини қайта ишловчи функциялардир. JavaScript сценарийда хусусиятларни адреслаш уларнинг номлари билан ёки уларнинг номерлари билан амалга ошири мумкин. Кейинги вариант бўйича, ҳар бир хусусият массивнинг бир элементи сифатида олинади ва улар ўзларининг уникал номерларига эга бўладилар.

JavaScript тилида C ва Java дастурлаш тилларидаги каби процедура ва функциялар мавжуд бўлиб, улар қуйидагича эълон қилинади:

- function калит сўзи;
- функция номи;
- вергул ва кавс билан ажратилган аргументлар рўйхати;
- фигурали кавс ичига олинган функция танаси.

```
function myFunction(arg1, arg2, ...)
```

```
{
```

```
...
```

```
Операторлар кетма-кетлиги
```

```
...
```

```
}
```

Бу ерда myFunction – функция номи, arg1, arg2 – параметрлар.

**Мисол:**

```
function Factorial(n) {  
  if((n<0)||((round(n)!=n)) {  
    alert("Factorial функцияси ушбу аргументда аниқланмади "+n);  
    return NaN;  
  } else {  
    result=(n*Factorial(n-1));  
    return result;  
  }  
}
```

```
}}
```

Функцияда return калит сўзи орқали қиймат қайтарилмаслиги ҳам мумкин.

**Мисол:**

```
function Greeting(s){  
    document.write("Hello,"+s+"!");  
    return ;  
}
```

Функцияни чақириш аниқ параметрлар билан чақирилади:

Мисол:

**Factorial(3);**

- бу функция натижаси 6 га тенг,

**Greeting("world");**

- бу функция экранга "Hello, world!" стрини чиқаради.

Ҳар бир функция, масалан, **myFunction** функцияси myFunction номли объект хисобланади, агарда аргументлар arguments номи билан берилса, унга мурожаат қуйидагича:

**myFunction.arguments[i]**, бу ерда **i** — аргумента номери (рақамлаш 0 дан бошланади).

Функция эълонида аниқ параметрлар формал параметрларга тенг еки кўп сонда бўлиши лозим. Бунда функция ишга туширилганда жунатилаётган аргументлар миқдори myFunction.arguments.length майдони ёрдамида аниқланади ва ушбу майдондаги қийматни қайта ўзлаштиришни динамик ўзгартириш мумкин.

**Мисол:**

Экранга HTML форматидаги рўёхатни чиқариш.

Бу ерда (ListType) нинг биринчи аргументи тартибланмаган рўйхат учун "o" еки "O", тартибланмаган рўйхат учун "u" еки "U" булиши мумкин.

```
function myList(ListType) {  
    document.write("<" + ListType + "L");  
    for(var i=1; i < myList.arguments.length; i=i+1) {  
        document.write("<LI>" + myList.arguments[i]);  
    }  
    document.write("</" + ListType + "L>");  
}
```

HTML хужжатида функцияга мурожаат қуйидагича:

```
<script> myList("o", "матн", 2, "3") </script>
```

**Натижа:**

матн

2

3

**Global класи**

Ушбу класс JavaScript нинг функционал қисми бўлиб, бу класс бир объектда бир нечта усул ва хоссаларни бирлаштириш вазифасини бажаради.

Усулга мурожаат қилинганда объект кўрсатилмайди, аниқроғи бу усул конструкторга эга бўлмайди. Бундай хосса ва усулларга қуйидагиларни келтириш мумкин:

| Хосса    | Мазмуни                                          |
|----------|--------------------------------------------------|
| Nan      | NaN (Not A Number)                               |
| Infinity | Number.POSITIVE_INFINITY қийматни ўз ичига олади |

| Усул       | Мазмуни                                              |
|------------|------------------------------------------------------|
| escape     | Қаторни барча платформаларга мос ҳолда тасвирлаш     |
| eval       | JavaScript тили функцияси еки усулларини узатиш      |
| isFinite   | Аргументнинг охири рақамлилигини аниқлаш             |
| isNaN      | Аргументнинг рақам ёки рақам эмаслигини аниқлаш      |
| parseFloat | Қаторни кўчиб юрвчи нуқтали сон кўринишида тасвирлаш |
| parseInt   | Қаторни бутун сонга айлантириш                       |
| unescape   | Escape функцияси натижасини қайтариш                 |

eval(s) функцияси - s қаторни JavaScript операторлари кетма-кетлиги кўринишида тасвирлаш.

getClass(Jobj) функцияси – JavaScript типдаги аргумент учун JavaScript объектини қайтаради.

**Мисол:**

```
var myJavaRClass=new java.awt.Rectangle()
var myJavaRClass=getClass(myJavaRect)
getClass() Java-методи билан адаштирманг:
var myJavaRObject=myJavaRect.getClass() - бу java.awt.Rectangle
класининг Java тилидаги реализация ҳолати.
```

isNaN(x) функцияси – x “Not a Number”, яъни сон эмаслигини текшириш.

parseFloat(s) функцияси – Float типдаги s рақамни аниқлаш. Агар сон топилмаса у ҳолда NaN (“Not a Number”) қиймати қайтарилади.

parseInt(s) – Integer тип учун юқоридаги ҳолат.

**eval(s) функцияси**

eval(s) функцияси – JavaScript нинг ички функцияси ҳисобланади. Ушбу функция бир ёки бир нечта JavaScript операторларидан иборат бўлган s сатрни аргумент томонидан узатилган кодни бажаради. Бунда s сатридаги операторлар нуқтали вергул ёрдамида ажратилади. Бу функция нафақат операторни бажариш, балки бирор амалларни ҳисоблаш имконини ҳам беради. Бунда у кодда келтирилган амал ҳисобининг охири қийматини қайтаради.

**isNaN(x) функцияси**

Бу функция x аргументнинг “сон эмас” лигини текширади. Натижа NaN қийматга эга эмаслигини, яъни мумкин булмаган сон (масалан, нолни нолга

бўлиш натижаси) ни текширади. Ушбу функция JavaScript да литерал кўринишда NaN қийматни бериш мумкин эмаслиги учун муҳимдир. Бундан ташқари parseFloat(s) ва parseInt(s) функциялар натижаларини текшириш (мумкин бўлган сон эканлигини) ва арифметик хатолар мавжудлиги, масалан, нол сонига бўлиш мавжудлигини текширади.

### **parseFloat(s) функцияси**

s сатрини синтактик анализ қилиш ва дастлаб рақамни қайтариш (сатрни рақамна айлантиради). parseFloat(s) да s сатрида рухсат этилмаган рақам элементлари (масалан, белгилар, рақам, ўнли вергуллар, даража кўрсаткичи ва ҳоказо) мавжуд бўлса анализ тўхтатилади ва қиймат қайтарилади. Агарда s сатрда сон билан бошланмаса, у ҳолда parseFloat(s) функция NaN қийматни қайтаради.

### **parseInt(s) функцияси**

Бу функция сатрни бутун сонга айлантиради. parseInt(s) функциядаги s сатрда ҳисоблаш тизимида кўрсатилмаган қийматларга эга бўлганда синтактик анализ тўхтатилади ва қиймат қайтарилади. Одатда, parseFloat ва parseInt функциялар s сатр сон билан бошланмаганда NaN қиймат қайтаради.

**parseInt(s,n)** ҳолатида n асос ихсобида, агарда n=10 бўлса, parseInt(s) функция сатрдаги 10 лик санок системасидаги сонларни текширади. n=8 бўлса, 8 лик санок тизимидаги сонлар мавжудлигини (бунда n 0 дан 7 гача бўлган сонлар қийматига эга бўлиши мумкин). n=16 бўлса, 16 лик санок тизимидаги сонлар мавжудлигини (бунда 0 дан 9 гача бўлган сонлар ва A дан F гача бўлган ҳарфлар қийматига эга бўлинади). Агарда n=0 бўлса ёки қиймат берилмаса, у ҳолда parseInt(s) функция сатрнинг ўзидан асосни аниқлайди. Бу ҳолатда агарда сатр 0x билан бошланса, унда функция сатрнинг қолган қисмини 16 лик санок тизимидаги сон сифатида анализ қилади, агарда сатр 0 дан бошланса, сатр 8 лик санок тизимидаги қиймат сифатида анализ қилинади.

## **Math классси**

Math – константалар ва методлардан иборат классдир. Улар объект учун одатдагидек мурожаат қилинади:

Math.константа

Math.функция(i..)

### **Math классси константалари**

E – e сони (натурал лагориғм асосли)

LN10 — 10 ли натурал лагориғм (ln10 сони)

LN2 — 2 ли натурал лагориғм (ln2 сони)

LOG10E — 10 асосли e лагориғм (log10e сони)

LOG2E — 2 асосли e лагориғм (log2e сони)

PI — p константаси ("пи" сони)

SQRT1\_2 — 2 нинг тескари квадрат илдизи ( $1/\sqrt{2}$ )

SQRT2 — 2 нинг квадрат илдизи ( $\sqrt{2}$ )

## Math классы методлари

$\text{abs}(x)$  ( $x$ -сон ёки ифода) – абсолют қийматни ҳисоблаш;

$\text{acos}(x)$  ( $x$  бу ерда  $[-1.0;1.0]$  радиан интервалдаги сон ёки ифода) – арккосинусни ҳисоблаш. Қайтариладиган қиймат 0 дан  $\pi$  радиан оралиғида бўлади.

$\text{asin}(x)$  ( $x$  бу ерда  $[-1.0;1.0]$  радиан интервалдаги сон ёки ифода) – арксинусни ҳисоблаш. Қайтариладиган қиймат  $-\pi/2$  дан  $\pi/2$  радиан оралиғида бўлади.

$\text{atan}(x)$  ( $x$  – сон ёки ифода) – арктангенсни радианларда ҳисоблаш. Қайтариладиган қиймат  $-\pi/2$  дан  $\pi/2$  радиан оралиғида бўлади.

$\text{atan2}(x,y)$  ( $x,y$  — тўғри бурчакли координата системаси координата нуқталари) – кутб координатасида  $(x,y)$  нуқталар бурчагини ҳисоблайди. Қиймати 0 дан  $2\pi$  радиан оралиғида бўлади.

$\text{ceil}(x)$  ( $x$  — сон ёки сонли ифода) – сонни бутун сонга йўналтирилган ҳолда яхлитлаш. Манфий сонлар 0 сони йўналишига қараб яхлитланади.

$\text{cos}(x)$  ( $x$  – радиандаги бурчак) – косинусни ҳисоблаш, қайтариладиган қиймат  $-1.0$  дан  $1.0$  радиан оралиғида бўлади.

$\text{sin}(x)$  ( $x$  – радиандаги бурчак) – косинусни ҳисоблаш, қайтариладиган қиймат  $-1.0$  дан  $1.0$  радиан оралиғида бўлади.

$\text{Exp}(x)$  ( $x$  — сон ёки сонли ифода) —  $e$  экспонентсини ҳисоблаш.

$\text{Floor}(x)$  ( $x$  — сон ёки сонли ифода) – сонни бутун қисмига йўналтириб яхлитлаш, масалан,  $\text{floor}(-1,1)$  тенг  $(-2)$ ;  $\text{floor}(1,1)$  тенг  $1$ .

$\text{Log}(x)$  ( $x$  — мусбат сон ёки ифода) – натурал лагориғини ҳисоблаш.

$\text{max}(a,b)$  ( $a,b$  — сон ёки ифода) – икки қийматдан каттасини қайтаради.

$\text{min}(a,b)$  ( $a,b$  — сон ёки ифода) – икки қийматдан кичигини қайтаради.

$\text{pow}(x,y)$  —  $x$  ни ҳисоблаш (биринчи аргументни даражага кўтариш).

$\text{random}$  — 0 дан 1 гача интервалдаги тасодиқий сонларни ҳисоблаш.

$\text{round}$  — сонни бутун қисмига қараб яхлитлаш (масалан,  $\text{round}(15.5)$  натижаси 16 ни беради,  $\text{round}(-15.5)$  даёт  $-15$ ).

$\text{Math.round}(x)$  ( $x$  — сон ёки ифода)

$\text{Math.sin}(x)$  ( $x$  — радианда берилган бурчак)

$\text{Math.sqrt}(x)$  ( $x$  — 0 га тенг ёки катта бўлган сон ёки ифода)

$\text{tan}$  — тангенсни ҳисоблаш.

$\text{Math.tan}(x)$  ( $x$  — радианда берилган бурчак)

## Класс Date

$\text{Date}()$  методи аргументсиз берилганда қиймати жорий сана ва вақтга эга  $\text{Date}$  объекти яратилади.  $\text{Date}()$  методида янги объект учун аргументи сифатида сана ва зарур ҳолларда вақт кўрсатилади.  $\text{Date}$  методи JavaScript тили объекти ҳисобланиб, HTML тилида ҳеч қандай аналогга эга эмас. Кўп ҳолларда  $\text{Date}$  объекти методлари унинг экземплярлари ёрдамида чақирилади, масалан:

```
d=new Date(); // бугунги сана ва вақтни олиш
```

```
system.write("Today is: "+d.toLocaleString()); // ва уни тасвирлаш
```

$\text{Date}$  объектини яратишнинг юқоридаги синтактикасида кўрсатилгани

бўйича, сана ва вақт ҳудудий вақт бўйича берилди. Агарда тузилаётган дастур фойдаланувчи жойлашган часовой поясга боғлиқ бўлмаган ҳолда ишлаши зарур бўлса, у ҳолда Гринвич (GMT) еки универсал координация вақти (UTC) бўйича санани кўрсатиш керак бўлади.

Date объектини яратишда қуйидаги 5 та синтактик вариантдан фойдаланиш мумкин. 3-5 вариантларда вақт ҳудудий тарзда интерпретация қилинади (Гринвич да эмас):

1. new Date();
2. new Date(миллисекунд) – бу ерда миллисекунд жорий сана билан 01.01.1970 сана яримкуни орасидаги сон;
3. new Date(сана сатри) – бунда сана сатри = ой номи, дд, гг [чч:мм[:сс]])
4. new Date(йил, ой, кун) – бунда, йил 2011; ой 0-11; кун 1-31;
5. new Date(йил, ой, кун, соат, минут, секунд) – 24 соатликтизимда.

#### **Date классининг методлари**

getDate() - Date объектининг 1 дан 31 гача ораликдаги қийматини беради;

getDay() - Date объектининг 0 [якшанба] дан 6 [шанба] гача ораликдаги ҳафта кунлари беради;

getHours() - Date объектининг 0 [ярим тун] дан 23 гача ораликдаги соат майдони қийматини беради;

getMinutes() - Date объектининг 0 дан 59 гача ораликдаги минут майдони қийматини беради;

getSeconds() - Date объектининг 0 дан 59 гача ораликдаги секунд майдони қийматини беради;

getMonth() - Date объектининг 0 [январ] дан 11 [декабр] гача ораликдаги ойларни беради;

getTime() - Date объекти вақт кўрсаткичининг миллисекундлардаги қийматини беради;

getFullYear() - Date объекти вақт кўрсаткичининг йиллар майдони қийматини беради; бунда 2011 йил 11 кўринишида берилди;

parse() – сананинг сатр кўринишидаги ҳолатини синтактик анализ қилади ва натижани миллисекунд форматда беради;

setDate() - Date объекти вақт кўрсаткичини ўрнатади;

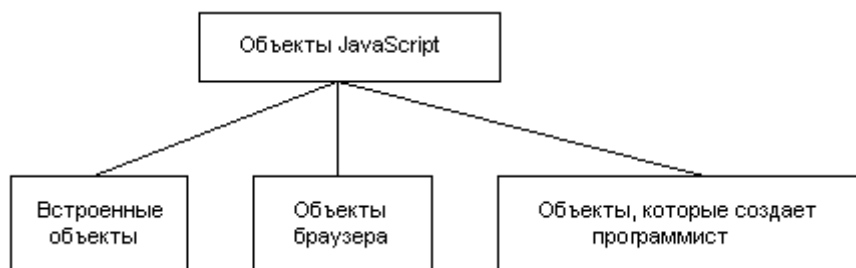
*data*.setDate(ой сони) //ой сони 1-31 ораликда.

toLocaleString() – жорий ҳудудий вақт майдонини асосида Date форматини матнли (String) кўринишга келтиради;

UTC() – сана ва вақтнинг рақамли кўринишини миллисекундли форматга айлантиради.

#### **JavaScript нинг уч турдаги объектлари**

JavaScript тилида уч турдаги объектлар мавжуд: стандарт объектлар, браузер объектлари ва дастурчи томонидан яратилувчи объектлар. Уларнинг ҳар бири ўзларининг таснифи ва хусусиятларига эга.



## Стандарт объектлар

Куйида JavaScript да қўлланилувчи стандарт объектлар, хусусиятлар ва усуллар келтирилган. Уларни ишлатишда олдиндан эълон қилиш талаб этилмайди.

| Объект   | Таснифи                           |
|----------|-----------------------------------|
| Array    | Массив                            |
| Boolean  | Мантиқий маълумотлар              |
| Date     | Календарли вақт                   |
| Function | Функция                           |
| Global   | Глобал усуллар                    |
| Math     | Математик константа ва функциялар |
| Number   | Сон                               |
| Object   | Объект                            |
| String   | Сатр                              |

Стандарт объектлар билан қандай ишлаш мумкин? Анча оддий. Объектни реализация қилувчи дастур ёзилади ва унинг хусусият ва усулларига мурожаат қилинади. Мисол сифатида жорий вақтни кўрсатувчи HTML хужжатни кўрамиз.

```

<HTML> <HEAD> <TITLE>Жорий кун ва вақт </TITLE> </HEAD>
<BODY BGCOLOR=WHITE>
  <H1> Жорий кун ва вақт </H1>
  <SCRIPT LANGUAGE="JavaScript">
  <!--
    var dt;
    var MyDate="";
    dt = new Date();
    MyDate = "Date: " + dt.getDate() + "." + dt.getMonth() + "." + dt.getYear();
    document.write(MyDate);
    document.write("<BR>");
    document.write("Time: " + dt.getHours()
      + ":" + dt.getMinutes() + ":" + dt.getSeconds());
  // -->
  </SCRIPT> </BODY></HTML>
  
```

Бу ерда JavaScript сценарий new калит сўзи ёрдамида Date объектини яратади. Бунда Date конструктори параметрларсиз келтирилади:

```
var dt;  
dt = new Date();  
MyDate = "Date: " + dt.getDate() + "  
+ dt.getMonth() + "." + dt.getYear();
```

getDate, getMonth ва getYear усуллар ёрдамида жорий сана олинади. Ушбу усуллар dt объекти учун чақирилади.

Матн сатри эса HTML хужжатга write усули ёрдамида босмага чиқарилади. Бу усул document объектининг усули ҳисобланади:

```
document.write(MyDate);
```

Date объекти жорий вақтни ҳам ўз ичига олади. Бу маълумотлар getHours, getMinutes ва getSeconds (соат, минут ва секунд) усуллари ёрдамида кўрилади:

```
document.write("Time: " + dt.getHours()  
+ ":" + dt.getMinutes() + ":" + dt.getSeconds());
```

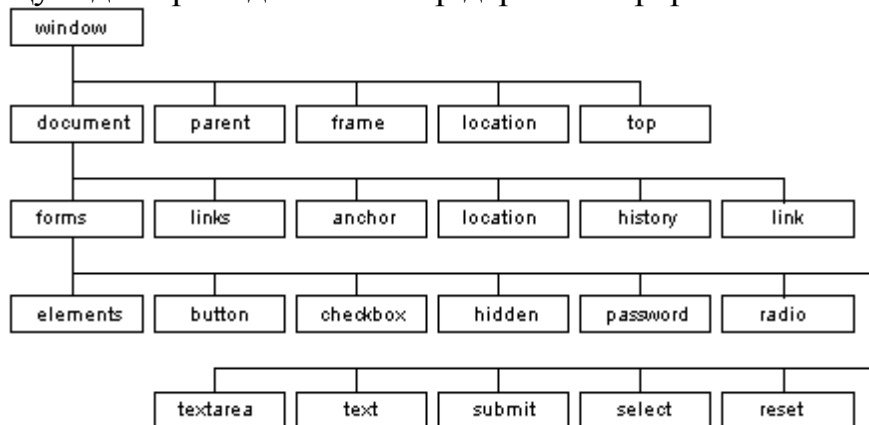
### Браузер объектлари

JavaScript сценарий нуқтаи назари бўйича объектлар иерархик дарахт кўринишда ташкил этилади.

Браузер объектлари фойдаланувчи учун яратилган, браузер ойнасида жойлашган объектлар ҳисобланади. JavaScript сценарида браузер объектлари, хусусият ва усулларида фойдаланиб бир класс асосида бошқа класс яратиб бўлмайди.

### Браузер объектлари иерархияси

Куйидаги расмда объектлар дарахти иерархияси келтирилган.



**window** объекти бу иерархиянинг илдизи ҳисобланади. Қачонки HTML хужжат юкланса унинг ичида **document**, **parent**, **frame**, **location** ва **top** бошқа объектлар ҳосил бўлади.

### Объектлар билан боғлиқ ҳолатлар

Браузернинг ҳар бир объекти билан аниқ бир ҳолатлар тўпламидан ташкил топади.

Масалан, **window** объекти **onLoad** ва **onUnload** ҳолатлари билан боғлиқ ҳолда ишлайди. Биринчи ҳолат браузер ойнаси юклаб бўлгач ишга тушади.



Иккинчиси эса браузер ойнани ёпиш вақтида ишга тушади.

### **Дастурчи томонидан яратилувчи объектлар**

Аввало myRecord номли класс яратамиз. Ҳозирча унда усуллар мавжуд эмас, уларни кейинчалик қўшамиз. Бу класс қуйидагича яратилади:

```
function myRecord(name, family, phone, address) {  
  this.name = name;  
  this.family = family;  
  this.phone = phone;  
  this.address = address;  
  this.secure = false;  
}
```

Яратилаётган объектни хусусиятларини кўрсатиш учун махсус this калит сўзидан фойдаланилади. Бу калит сўз объектнинг хусусиятларига бўлган муурожаатини кўрсатади.

Келтирилган классдан қандай фойдаланиш мумкин? Яратилган класс асосида исталган сондаги объектлар яратиш мумкин. Қуйида берилган myRecord классида иккита rec1 ва rec2 объектлари яратилган:

```
var rec1;  
var rec2;  
rec1 = new myRecord("Иван", "Иванов", "000-322-223", "А. Темур кўча,  
д. 225, кв. 226");  
rec2 = new myRecord("Петр", "Петров", "001-223-3334", "Бобур кўча, д.  
552, кв. 662");  
rec2.secure = true;
```

Объектлар **new** оператори ёрдамида яратилади.

## **Назорат саволлари**

Масаланинг қўйилиши

JavaScript объектларини ўрганиш.

JavaScript нинг танланган объекти асосида мисол келтириш.

1. Объект Global (ошкормас ҳолда бериш)
2. Объект Math
3. Объект Window
4. Коллекция фреймов (window.frames)
5. Объект Document (window.document)
6. Объект Location (window.location)

7. Обьект Link (document.link)
8. Обьект History
9. Обьект MimeType
10. Обьект Navigator

### **Фойдаланилган адабиётлар.**

1. VG Olifer, N. A.Olifer. Computer networks. Principles, technologies, protocols. St. Petersburg: Publishing House. "Peter," in 2000.
2. R. Darnell, "JavaScript: Handbook." - St. Petersburg: Publishing House. "Peter," in 1998.
3. Duvanov A.A. WEB. Construction of HTML. - St. Petersburg: BHV, 2003.
4. R. Eisenmenger. HTML 3.2/4.0. Handbook. - M: Univ. "Bean", 1998.
5. Stephen Speynaur, Valerie Kuersia. Directory WEB-master. - Ed. BHV
6. Barrett D. JavaScript. Web-professionals. - Kiev: BHV - Kyiv, 2001.
7. Brandenbau D. JavaScript: a collection of recipes. - St. Petersburg.: Peter, 2000.

## 5 – амалий машғулот.PHP га кириш. Маълумотлар типлари билан ишлаш. Ўзгарувчилар. Амаллар. Операциялар

**Ишдан мақсад:** PHP ни ўрганиш ва маълумотлар типларидан фойдаланиш, ўзгарувчилар, амаллар операциялар.

**Масаланинг қўйилиши:**

**Билиши керак** маълумотлар типлари. ўзгарувчилар, амаллар ва операциялар.

**Қила олиши керак** PHPда ишлаш;

### Ишни бажариш учун намуна

Кўп ҳолларда *PHP* тилини интерпретатори ишлаётганлигини текшириб кўриш учун тузиладиган дастур энг содда дастур деб аталади. Ҳозир биз *PHP* тилидаги ушбу дастурни чуқур ўрганамиз ҳамда уни бошқа дастурлаш тиллари Си, Perl ва JavaScript лардан фарқли томонини текширамиз. Ушбу мисолни кўрамиз:

**Мисол. PHP кодида тузилган содда html-файл:**

```
<html>
  <head> <title> Мисол </title> </head>
  <body>
    <?php echo "<p> Салом, бу мен – PHP скрипт! </p>"; ?>
  </body></html>
```

Бу *PHP* дастурлаш тилининг махсус кодли теглари ёрдамида тузилган содда html-файлдир.

Юқорида айтиб ўтганимиздек, *PHP* дастурлаш тили Си ва Perl дастурлаш тилига ўхшаш. Бироқ келтирилган дастур Си ва Perl дастурлаш тилидаги дастурдан анча катта фарқ қилади. Бу ерда HTML саҳифага чиқариш учун бир қатор махсус буйруқларни ёзиш шарт эмас. Бевосита *PHP*-код асосида қурилган бирор вазифани бажарадиган HTML-скрипт ёзилади (бизни мисолда экранда чиқарилган матн). *PHP* дастурлаш тилининг Си ва Perl дастурлаш тилларидан камчилиги шуки, мураккаб скриптларни *PHP* дастурлаш тили анча секин бажаради.

**PHP-скриптлар** – бу серверда бажариладиган ва қайта ишланадиган дастурлардир. Бу скриптларни JavaScript типдаги скриптлар билан таққослаш мумкин эмас, чунки JavaScript тилидаги скриптларда ёзилган буйруқлар фақат клиент компьютеридагина бажарилади. Клиент компьютерида ва сервер компьютерида бажариладиган скриптларнинг фарқи нимада? Агарда скрипт серверда қайта ишланса, мижоз компьютерига фақатгина натижа юборилади. Масалан, агарда серверда скрипт бажарилаётган бўлса, юқорида келтирилганга ўхшаб мижоз HTML-саҳифа кўринишдаги натижани олади:

```
<html>
  <head> <title> Мисол </title> </head>
  <body> <p> Салом, бу мен – PHP скрипт! </p>
</body></html>
```

Бу ҳолатда мижоз қандай код бажарилаётганини билмайди. Ўз серверингизни HTML-файлларни *PHP* процессори қайта ишлайдиган қилиб сошлаб олишингиз ҳам мумкин. Яъни клиентлар оддий HTML-файлни қабул қилдими ёки скрипт натижасини кўрдими буни била олмайди. Агарда скрипт клиент компьютерида қайта ишланса (масалан, JavaScript тилидаги дастур), у ҳолда клиент скрипт кодидан иборат HTML-саҳифани кўради.

Биз юқорида айтиб ўтгандикки, *PHP-скриптлар HTML*-код ичида ёзилади. Қандай қилиб деган савол туғилади. Бунинг бир нечта усуллари мавжуд. Булардан бири биринчи мисолда келтирилганидек, `<?php` теги билан бошланиб `?>` теги билан тугаган синтаксис. Бундай кўринишдаги махсус теглар HTML ва *PHP* режимидагина ишлатилади. Бу синтаксис *PHP* тилини *XML* ҳужжатлари билан биргаликда ишлайдиган дастурларида жуда маъқул кўрилади (масалан, XHTML тилида ёзилган дастурларда). Бироқ базан қуйидаги альтернатив вариантдан фойдаланса ҳам бўлади (echo "Some text" буйруғи «Some text» матнини экранга чиқаради.):

```
<? echo "Бу PHP тилида оддий қайта ишлашнинг инструкцияси"; ?>
<script language="php">
  echo "Бир нечта редакторлар (FrontPage) қуйидагича қабул қилишади";
</script>
<% echo " ASP технологиясидаги тегдан ҳам фойдаланса бўлади"; %>
```

Бу келтирилган усуллардан биринчиси ҳар доим ҳам бажарилавермайди. Ундан фойдаланиш учун қисқа тегларни ишлатиш керак, ёки *PHP3* учун `short_tags()` функцияни ишлатиш керак, ёки *PHP* тилининг конфигурацион файлига `short_open_tag` буйруқни ўрнатиш керак, ёки *PHP* дастурлаш тилида `enable-short-tags` параметр билан компиляция қилиш керак. Агарда `php.ini-dist` буйруққа юқоридагилар автоматик қўшилган бўлса, у ҳолда қисқа теглардан фойдаланиш тавсия этилмайди. Иккинчи усул худди ўрнига қўйишга ўхшайди, масалан, JavaScript кодлари ва унинг учун мос html теглар. Шунинг учун ундан ҳар доим фойдаланиш мумкин, лекин бу ноқулайлиги учун камдан-кам ишлатилади. Учинчи усулдан фақат ASP технологиясидаги теглар `asp_tags` конфигурациясида ишлатилгандагина фойдаланилади.

*PHP* дастурлаш тили файлни қайта ишлаётганда у оддий матнни *PHP* код интерпретация қилиши керак бўлган махсус тегларни учратмагунча қайтариб беради. Интерпретатор ҳақида гапирганда у топилган барча кодни ёпиладиган теггача бажаради, сўнг яна оддий матн қайтарилади. Бу механизм *PHP*-кодни HTML саҳифага айлантиради, яъни барча *PHP* теглардан ташқари барча матнларни ўзгаришсиз сақлайди ва ичкаридагиларни эса интерпретациялайди. Яна шуни айтиш керакки, `php`-файл *CGI*-скриптга

ўхшамайди. php-файл бажарилиши шарт эмас, ёки яна қандайдир белгиланади.

php-файлни серверда қайта ишлаш учун жўнатишда сервер томонидан браузер сатрида бу файлни йўлини кўрсатиш шарт. PHP скриптлар www орқали киришга рухсат этилган жойда жойлашиши шарт. Агарда php-файл локал компьютерда мавжуд бўлса, у ҳолда уни буйруқлар сатри интерпретатори ёрдамида қайта ишлаш мумкин.

### **Асосий синтаксислар.**

Инструкцияни бир нечта қисмга бўлиб кўриб чиқамиз, яъни комментарийлар яратиш, ўзгарувчилар, ўзгармаслар ва маълумот типлари, операторларга.

Биз энди PHP дастурлаш тилининг асосий синтаксис элементларини ўрганишга ўтамиз. Мисол сифатида электрон мактуб тайёрлаш масаласини кўриб ўтайлик. Унинг маъноси қуйидагидан иборат.

Фараз қиламизки, сизда қандайдир эълон ва эълонни жўнатишингиз керак бўлган бир нечта одамлар мавжуд бўлсин. Бунинг учун сиз эълонни ичида ўзгарадиган (қабул қилувчи билан боғлиқ бўлмаган) бир нечта параметрлари мундарижаси билан тайёрлайсиз.

Биринчи навбатда PHP дастурлаш тили синтаксисига нисбатан нималарни билиш керак. Бу HTML-код ичига ўрнатилган ва PHP дастурлаш тилидаги коддир, уни интерпретатор фарқлай билади. Аввалги бўлимларда булар ҳақида айтиб ўтгандик. Ҳаммасини қайтариб ўтмаймиз, фақат биз кўп ҳолларда мисолларда `<?php ?>` вариант ўрнига қисқартирилган `<? ?>` теглардан фойдаланишни айтиб ўтамиз.

### **Инструкцияларни ажратилиши.**

PHP дастурлаш тилидаги дастур(ихтиёрий дастурлаш тилидаги) – бу буйруқлар (инструкциялар) тўпламидир. Дастурни қайта ишлаш учун бир буйруқни бошқа буйруқдан фарқини билиш керак. Бунинг учун махсус символлар – ажратгичлардан фойдаланилади. PHP дастурлаш тилида инструкцияларни худди Си ёки Perl дастурлаш тиллари каби ажратилади, яъни ҳар бир ифода нуқтали вергул (“;”) билан тугайди.

«?»» ёпиладиган тег ҳам инструкцияни тугагини англатади, шунинг учун ундан олдин нуқтали вергул қўйилмайди. Масалан, қуйидаги икки фрагментлар эквивалентдир:

```
<?php
echo "Hello, world!"; // буйруқлар охирида нуқтали вергул қўйиш шарт
?>
<?php
echo "Hello, world!" ?>
<!-- "?"> борлиги учун нуқтали вергул ташлаб кетилди -->
```

## Комментарийлар.

Кўп ҳолларда дастур тузганда кодни тушунарли бўлиши учун унга қандайдир изоҳ-*комментарийлар* қўйиш керак бўлиб қолади. Бу ҳолат катта хажмдаги дастурлар яратганда ҳамда агарда битта дастур устида бир нечта дастурчи ишлаётганда жуда муҳим. Комментарийлар дастурнинг коди тушунарли бўлиши учун ёзилади. Бундан ташқари масалани қисмларга ажратиб ҳал қилинганда ишнинг камчилиги бор жойида кейинчалик эсан чикмаслиги учун комментария ёзиб қўйилади. Барча дастурлаш тилларида дастур ичига комментария қўйиш имконияти мавжуд. *PHP* дастурлаш тили бир қанча кўринишдаги комментарийларни қўллаб қувватлайди: `Си`, `С++` дастурлаш тиллари стилидаги ҳамда `Unix` қобиғидаги комментарийлар. `//` ва `#` белгилар бир сатрли комментарийларни англатса, `/*` ва `*/` белгилар эса мос равишда кўп сатрли комментарийларнинг бошланиш ва тугашини англатади.

```
<?php
echo "Мени исмим Алишер";
// Бу бир сатрли комментарий С++ дастурлаш тили стилидаги
echo "Мени фамилиям Болиев";
/* Бу кўп сатрли комментарий. Бу ерга бир қанча сатр ёзиш мумкин.
Дастур бажарилиш жараёнида бу ердаги барча ёзувлар (комментарийланган),
ўқилмайди. */
echo "Мен PHP дастурлаш тилини INTUIT.ru дан ўрганияпман";
# Бу комментарий Unix қобиғидаги комментарий.
?>
```

## Ўзгарувчилар, ўзгармаслар ва операторлар.

Ҳар бир дастурлаш тилида муҳим элементлардан бири бу *ўзгарувчилар*, *ўзгармаслар* ва улар қўлланиладиган *операторлар*дир. *PHP* дастурлаш тили бу элементларни қандай белгилаши ва қайта ишлашини кўриб чиқамиз.

## Ўзгарувчилар

*PHP* дастурлаш тилида *ўзгарувчилар* олдида доллар белгиси (“\$”) қўйиб эълон қилинади, масалан, `$my_var`.

*Ўзгарувчилар* номлари регистрларни фарқлайди, яъни `$my_var` ҳамда бош ҳарфли `$My_var` ўзгарувчилари турли хил ўзгарувчилардир.

*PHP* дастурлаш тилида ўзгарувчилар номи қолган дастурлаш тиллари қоидалари каби эълон қилинади: ўзгарувчи номи лотин алфавити билан бошланиши ва ундан кейин ҳарфлар ёки тагига чизилган белги ёки рақамлар бўлиши мумкин.

*PHP3* дастурлаш тилида ўзгарувчилар ҳар доим бирор-бир қийматга ўзлаштирилади. Яъни ўзгарувчини бирор-бир ифодага ўзлаштирсак, ифоданинг қиймати ўзгарувчига ўзлашади. Буни қуйидаги мисолда кўриш мумкин, яъни бир ўзгарувчини қиймати бошқасига ўзлаштирилганда улардан бирини қийматини бошқасига таъсир кўрсатмайди.

### Мисол. Қиймат бўйича ўзлаштириш

```
<?php
$first = ' Text '; // $first ўзгарувчига ' Text ' қийматни ўзлаштирдик
$second = $first; // $second ўзгарувчига $first ўзгарувчи қиймати
ўзлаштирилди
$first = ' New text '; // $first ўзгарувчи қиймати ' New text ' қийматга
ўзгартирилди
echo "first номли ўзгарувчи қиймати ".
    "$first га тенг. <br>";
    // $first ўзгарувчи қийматини экранга чиқарамиз
echo "second номли ўзгарувчи қиймати ".
    "$second га тенг. ";
    // $second ўзгарувчи қийматини экранга чиқарамиз
?>
```

Бу скриптнинг натижаси қуйидагича бўлади:

first номли ўзгарувчи қиймати Text га тенг.

second номли ўзгарувчи қиймати New text га тенг.

*PHP* дастурлаш тилида булардан ташқари ўзгарувчига қиймат ўзлаштиришнинг яна бир усули мавжуд: *ссылка бўйича ўзлаштириш*. Ссылка бўйича ўзгарувчига қиймат ўзлаштириш учун уни номи бўлиши шарт, яъни у қандайдир ўзгарувчини тақдим этиши керак. Бир ўзгарувчи қийматини бошқа ўзгарувчига *Ссылка бўйича ўзлаштириш* учун биринчи ўзгарувчи олдида амперсанд & белгиси қўйиш шарт.

Бунга юқоридаги мисолни кўриб чиқамиз, фақат first ўзгарувчи second ўзгарувчига *ссылка бўйича ўзлаштирилади*:

### Мисол. Ссылкалар бўйича ўзлаштириш

```
<?php
$first = ' Text '; // $first ўзгарувчига ' Text ' қиймат ўзлаштирилди
$second = &$first;
/* $second.орқали $first ўзгарувчига ссылка қиламиз. Энди бу
ўзгарувчилар қийматлари ҳар доим тенгдир */
// $first ўзгарувчи қийматини ' New text ' қийматга ўзгартирамиз
$first = ' New text ';
echo "first номли ўзгарувчи қиймати ".
    "$first га тенг <br>";
// $second ўзгарувчи қийматини экранга чиқарамиз
echo "second номли ўзгарувчи қиймати ".
    "$second га тенг";
?>
```

Бу скриптнинг натижаси эса қуйидагича бўлади:

first номли ўзгарувчи қиймати New text га тенг.

second номли ўзгарувчи қиймати New text га тенг.

Яъни \$first ўзгарувчи қиймати ўрнига \$second ўзгарувчи қиймати ўзлаштирилди.

## Ўзгармаслар

Скрипт бажарилиш жараёнида ўзгармайдиган қийматли катталикларни сақлаш учун **ўзгармаслар**дан фойдаланилади. Бундай катталиклар математик ўзгармаслар, пароллар, файлларнинг йўллари ва бошқалар бўлиши мумкин. Ўзгармасларнинг ўзгарувчилардан асосий фарқи шуки, уларни фақат бир мартагина ўзлаштирилади ва уни қийматини эълон қилингандан кейин бекор қилиб бўлмайди. Бундан ташқари ўзгармаслар олдида доллар белгиси қийилмайди ҳамда уни оддий қиймат ўзлаштириш каби қараш мумкин эмас. Ўзгармаслар қандай аниқланади? Бунинг учун махсус `define()` функцияси мавжуд, унинг синтаксиси қуйидагичадир:

```
define("Ўзгармас номи",  
      "Ўзгармас қиймати",  
      [регистрга_сезгирлиги_кичик])
```

Ўзгармаслар номи регистрга сезгирлиги катта. Ҳар бир ўзгармасларда уни ўзгартириш мумкин, яъни `регистрга_сезгирлиги_кичик` аргументни қиймати сифатида `True` қиймати кўрсатилади. Ўзгармаслар номи ҳар доим катта регистр билан ёзишга келишиб олинган.

Ўзгармасни қийматини билиш учун уни номини кўрсатиш керак. Ўзгарувчидан фарқи ўзгармас номи олдида `$` белги қўйилмайди. Бундан ташқари ўзгармасни қийматини билиш учун константа номи билан параметр сифатида `constant()` функциясидан фойдаланиш мумкин.

### Мисол. PHP дастурлаш тилида ўзгармаслар

```
<?php  
// ўзгармасни аниқлаймиз PASSWORD  
define("PASSWORD","qwerty");  
// регистрланмаган PI ўзгармасни қийматини аниқлаймиз 3.14  
define("PI","3.14", True);  
// PASSWORD ўзгармас қийматини оламиз, яъни qwerty  
echo (PASSWORD);  
// бу ҳам qwerty ни чиқаради  
echo constant("PASSWORD");  
echo (password);  
/* password ни чиқаради ва биз регистрланган ўзгармас PASSWORD ни  
кутгандик.*/  
echo pi;  
// 3.14 ни чиқаради, чунки ўзгармас PI регистрланмаган ва аниқланган.  
?>
```

Дастурчи томонидан ўзгарувчилардан ташқари юқорида айтиб ўтганимиздек *PHP* дастурлаш тилида мавжуд ўзгармаслар ҳам интерпретатор томонидан аниқланади. Масалан, `__FILE__` ўзгармас дастур бажарилиш жараёнида файл номини (ва файл йўлини), `__FUNCTION__` функция номидан ташкил топади, `__CLASS__` – синф номи, `PHP_VERSION` – *PHP* дастурлаш тили



интерпретатори версиясини ўзида сақлайди. Бундай ўзгармасларнинг барча рўйхатини *RHP* дастурлаш тили учун мўлжалланган қўлланмалардан топиш мумкин.

### Амаллар

Ўзгарувчилар, ўзгармаслар ва ифодалар устида турли ҳисоблашларни бажарадиган бу **амаллар**дир. Биз ҳали бу ифодалар ҳақида тўхтаб ўтганимиз йўқ. Ифодалар қийматини ушбу амаллар ёрдамида аниқланади. Ўзгарувчилар ва ўзгармаслар – бу ифодаларнинг асосий ва жуда содда шаклидир. Шундай ифодаларни кўпайтириши мумкин бўлган амаллар тўплами мавжуд. Уларни куйида тўлиқроқ муҳокама қиламиз:

9.1-жадвал. Арифметик амаллар.		
Белгиланиши	Номланиши	Мисол
+	Қўшиш	$\$a + \$b$
-	Айириш	$\$a - \$b$
*	Кўпайтириш	$\$a * \$b$
/	Бўлиш	$\$a / \$b$
%	Бўлишдаги қолдиқ	$\$a \% \$b$

9.2-жадвал. Сатрли амаллар.		
Белгиланиши	Номланиши	Мисол
.	Конкатенация (сатрларни қўшиши)	$\$c = \$a . \$b$ (бу $\$c$ сатр $\$a$ ва $\$b$ сатрлардан иборат)

9.3-жадвал. Ўзлаштириш амаллари.			
Белгиланиши	Номланиши	Изоҳ	Мисол
=	Ўзлаштириш	Оператордан ўнг томонда турган ўзгарувчилар устида бажарилган амаллардан ҳосил бўлган натижа қиймати ўзлаштирилади.	$\$a = (\$b = 4) + 5;$ ( $\$a$ 9 га тенг, $\$b$ 4 га тенг)
+=		Қисқартириш. Ўзгарувчига сон қўшилади ва кейин натижа ўзлаштирилади.	$\$a += 5;$ ( $\$a = \$a + 5$ ифодага эквивалент;)
.=		Ўзлаштириш ва конкатенация амаллари комбинациясини қисқартирилган шакли(даставвал	$\$b = "Ҳаммага";$ $\$b .= "салом";$ ( $\$b = \$b .$

		сатрлар қўшилади, сўнгра ҳосил бўлган сатр ўзгарувчига ўзлашади).	"салом" ифодага эквивалент;) <b>Натижаси:</b> \$b="Ҳаммага салом"
--	--	-------------------------------------------------------------------	-------------------------------------------------------------------------

<b>9.4-жадвал. Матикий амаллар.</b>			
<b>Белгиланиши</b>	<b>Номланиши</b>	<b>Изох</b>	<b>Мисол</b>
and	ВА	\$a ва \$b рост (True)	\$a and \$b
&&	ВА		\$a && \$b
or	ЁКИ	\$a ёки \$b ўзгарувчилардан ҳеч бўлмаганда биттаси рост бўлса (иккаласи ҳам рост бўлишиш мумкин).	\$a or \$b
	ЁКИ		\$a    \$b
xor	Инверсия ЁКИ	Ўзгарувчилардан биттаси рост бўлса. Агарда иккаласи ҳам рост бўлса инверсияланади.	\$a xor \$b
!	Инверсия (NOT)	Агарда \$a=True, у ҳолда !\$a=False ва акс ҳолда тескариси бўлади.	! \$a

<b>9.5-жадвал. Такқослаш амаллари.</b>			
<b>Белгиланиши</b>	<b>Номланиши</b>	<b>Изох</b>	<b>Мисол</b>
==	Тенглик	Ўзгарувчилар қийматлари тенг	\$a == \$b
===	Эквивалентлик	Ўзгарувчилар қийматлари ва типлари тенг	\$a === \$b
!=	Тенгсизлик	Ўзгарувчилар қийматлари тенг эмас	\$a != \$b
<>	Тенгсизлик		\$a <> \$b
!==	Ноэквивалентлик	Ўзгарувчилар эквивалент эмас	\$a !== \$b
<	Кичик		\$a < \$b
>	Катта		\$a > \$b
<=	Кичик ёки тенг		\$a <= \$b
>=	Катта ёки тенг		\$a >= \$b

<b>9.6-жадвал. Инкремент ва декремент амаллари.</b>			
<b>Белгиланиши</b>	<b>Номланиши</b>	<b>Изох</b>	<b>Мисол</b>
<code>++\$a</code>	Пре-инкремент	<code>\$a</code> қиймати бирга оширилади ва <code>\$a</code> қиймати қайтарилади	<pre>&lt;? \$a=4; echo "4 бўлиши шарт:" .\$a++; echo "6 бўлиши шарт:" .++\$a; ?&gt;</pre>
<code>\$a++</code>	Пост-инкремент	<code>\$a</code> қиймати қайтарилади ва сўнгра <code>\$a</code> қиймати бирга оширилади	
<code>--\$a</code>	Пре-декремент	<code>\$a</code> қиймати бирга камайтиради ва <code>\$a</code> қиймати қайтарилади	
<code>\$a--</code>	Пост-декремент	<code>\$a</code> қиймати қайтарилади ва сўнгра <code>\$a</code> қиймати бирга камайтиради	

### Маълумотлар типлари.

*PHP* дастурлаш тили саккизта содда маълумот типларини қўллаб қувватлайди:

Тўрттаси скаляр типлар:

- *boolean* (мантиқий);
- *integer* (бутун);
- *float* (нуқтаси силжийдиган);
- *string* (сатрли).

Иккитаси аралиш типлар:

- *array* (массив);
- *object* (объект).

Иккитаси махсус типлар:

- *resource* (ресурс);
- *NULL*.

*PHP* дастурлаш тилида ўзгарувчилар типлари ошкора эълон қилинмайди. Кўпинча ўзгарувчи қўлланилган контекздан, яъни ўзгарувчига ўзлаштирилган қиймат итпидан мустақил равишдаги дастур бажарилиш жараёнидан интерпретатор ўзи бу ишни бажаради. Қуйида юқорида санаб ўтилган маълумотлар типларини бирма-бир кўриб чиқамиз.

### Boolean тип (Буль ёки мантиқий тип).

Бу содда тип қийматни рост эканлигини ифодалайди, яъни ўзгарувчи фақат иккита қиймат қабул қилади – рост TRUE ёки ёлғон FALSE.

Мантиқий типларни аниқлаш учун TRUE ёки FALSE калит сўзларидан фойдаланамиз. Бу иккала типлар регистрланмаган.

```
<?php $test = True; ?>
```

Мантиқий типлар турли *бошқариладиган конструкциялар*да (цикллар, шартлар ва шунга ўхшаш, булар ҳақида кейинроқ айтиб ўтамиз) қўлланилади. Бир қанча амаллар (масалан, тенглик амали) ҳам мантиқий тип қабул қилиши мумкин, яъни фақат икки қиймат рост ёки ёлғон қийматни қабул қилади. Улар *бошқариладиган конструкциялар*да шартларни текшириш учун қўлланилади. Масалан, шартли конструкторда амаллар ёки ўзгарувчилар қиймати ҳақиқийлигини текширади ва натижадан қатъий назар шу ёки бошқа амалларни бажарилишини текширади. Бу ерда шарт рост ёки ёлғон бўлиши мумкин, чунки *мантиқий тип амаллари* ва *ўзгарувчилар* кўрсатилган.

### Мисол. Мантиқий типларнинг қўлланилиши

```
<?php
// '==' амал тенгликка текширади
// мантиқий қийматни
// қайтаради.
if ($know == False) { // агар $know қиймат false бўлса
echo "PHP дастурлаш тилини ўрган!";
}
if (!$know) { // худди юқоридагидек $know қиймати false бўлади
echo " PHP дастурлаш тилини ўрган!";
}
/* == амал $action ўзгарувчи қиймати билан "PHP дастурлаш тилини
ўрганиш!" сатрни
устма-уст тушишини текширади. Агар устма-уст тушса true қийматни
қайтаради, бошқа ҳолда false ни қайтаради. Агар true ни қайтарса фигурали
кавс ичидаги амаллар бажарилади. */
if ($action == " PHP дастурлаш тилини ўрганиш ") { echo "Ўрганишни
бошладим";}
?>
```

### Integer (бутун) типи.

Бу тип бутун сонлар тўпламидан  $Z = \{ \dots, -2, -1, 0, 1, 2, \dots \}$  бирини қайтаради. Бутун сонлар хоҳишга қараб олдиға «-» ёки «+» белгиларни қўйиб санок системасини ўнлик, ўн олтилик ёки саккизлик тизимларида кўрсатилган бўлиши мумкин.

Агар сиз саккилик санок системасидан фойдаланаётган бўлсангиз, олдиндан 0 (ноль) рақамини кўрсатишингиз керак. Ўн олтилик санок системасида эса рақамлар олдиға 0x белгини қўйиш шарт.

```
<?php
# ўнлик рақам
$a = 1234;
```

```
# манфий сон
$a = -123;
# саккизлик сон (ўнлик системасидаги 83 сонга эквивалент)
$a = 0123;
# ўн олтилик сон (ўнлик системасидаги 26 сонга эквивалент)
$a = 0x1A;
?>
```

*Бутун сонни* ўлчами платформага боғлиқ, лекин қоидага кўра максимал қиймати икки миллиард (бу ишорали 32 битли қиймат) атрофида бўлади. Ишорасиз *бутун сонни PHP* дастурлаш тили қўллаб қувватламайди.

Агар сиз *бутун сон* чегарасидан ташқари бирор қиймат берсангиз интерпретатор бу сонни *қўзғалувчан вергулли сонга* ўзгартиради. Худди шундай *бутун сон* чегарасидан ташқари чиқиб кетадиган бирор амал бажарсангиз ҳам бу сонни *қўзғалувчан вергулли сонга* ўзгартирилади.

*PHP* дастурлаш тилида бутун сонларни бўлиш амали мавжуд эмас.  $1/2$  ифода қиймати *қўзғалувчан вергулли сон* 0.5 га тенг. Сиз натижангизни бутун типга стандарт қоида асосида ёки `round()` функциясидан фойдаланган тақдирда ўзгартиришингиз мумкин. Ўзгарувчини аниқ бир типга ўзгартириш учун унинг олдида қавс ичида керакли типни ёзиш керак бўлади. Масалан,  $a=0.5$  ўзгарувчини бутун типга ўзгартириш учун `(integer)(0.5)` ёки `(integer) $a` кўринишда ёки қисқартирилган `(int)(0.5)` кўринишда ёзиш керак бўлади. Бундай ошқора янги типга ўтиш имконияти барча *маълумотлар типлари* учун ўринли бўлади (албатта, ҳар доим ҳам қийматни бир типдан бошқасига олиб ўтиш шарт эмас). Биз келтирилган барча типларни чуқур ўрганишимиз шарт эмас, чунки *PHP* дастурлаш тили контекстан мустақил равишда ўзи бу ишларни бажаради.

### **Float (қўзғалувчан вергулли сон) типи.**

*Қўзғалувчан вергулли сонлар* (улар икки карра аниқлик ёки ҳақиқий сонлардир) қуйидаги синтаксислар ёрдамида аниқланиши мумкин:

```
<?php
$a = 1.234;
$b = 1.2e3;
$c = 7E-10;
?>
```

*Қўзғалувчан вергулли сонни* ўлчами ҳам платформага боғлиқ, лекин қоидага кўра максимал қиймати  $\sim 1.8e308$  аниқлик билан 14 хонали рақам атрофида бўлади.

### **String (сатр) типи**

**Сатр** – бу белгилар тўпламидир. *PHP* дастурлаш тилида белги бу бир байт ва 256 та турли белгилар мавжуд. *PHP* дастурлаш тили Unicode типдаги белгиларни қабул қилмайди. *PHP* дастурлаш тилида амалда сатрларга чегирма мавжуд эмас, шунинг учун сатрларни ишлатганда унинг аниқ узунлиги ҳақида ўйлаш шарт эмас.

*PHP* дастурлаш тилида сатрлар учта турли хил усулларда аниқланади:

- *битталиқ қўштирноқлар* ёрдамида (‘’);
- *қўштирноқлар* ёрдамида (“”);
- *heredoc-синтаксиси* ёрдамида.

### Биттали тирноқлар

Сатрларнинг аниқлашнинг оддий усули – у «'» *биттали қўштирноқлар* ичида ёзилади. Агарда сатр ичида ҳам биттали тирноқ ишлатишга тўғри келиб қолса, биттали тирноқдан олдин «\» белгини қўйиш, яъни уни экранлаш шарт. Агарда «\» белги биттали тирноқдан олдин ёки сатрнинг охирида бўлса, у ҳолда белгини иккилантириш керак, яъни «\\'».

Агарда биттали тирноқ ичидаги сатр ичида ихтиёрий белгидан олдин («\» ва «'» лардан фарқли равишда) тескари слэш «\» белгиси учраса, у ҳолда уни оддий белги деб қараб барча белгиларни ўз ҳолича экранга чиқаради. Шунинг учун тескари слэш «\» белгисини сатр охирида ёпиладиган қўштирноқдан аввал турганини экранлаш шарт.

*PHP* дастурлаш тилида тескари слэш «\» белгиси билан ифодаланадиган бир қатор белгилар мажмуи мавжуд. Уларни **кетма-кетликни бошқарувчилар** деб аталади ҳамда улар махсус вазифаларни бажаради. Улар ҳақида кейинроқ тўхталиб ўтамыз. Ўзгарувчилар ва кетма-кетликни бошқарувчилар битталиқ қўштирноқлар сатри ичида учрашса, улар ўртасидаги фарқ *кетма-кетликни бошқарувчиларни қайта ишланмайди*.

```
<?php
echo 'Сатрлар мажмуи';
```

// Экранга чиқаради: ' белгини чиқариш учун ундан олдин \ белги қўйилади.

```
echo ' Белгини \' чиқариш учун ундан олдин' '\\ белгини қўйиш керак';
```

```
// Экранга чиқаради: Сиз шунини ўчирмоқчимисиз C:\*.*?
```

```
echo ' Сиз шунини ўчирмоқчимисиз C:\\*.*?';
```

```
// Экранга чиқаради: Буни қўйманг: \n янги қаторга
```

```
echo ' Буни қўйманг: \n янги қаторга ';
```

```
// Экранга чиқаради: ўзгарувчи $expand ҳам $either қўйилмайди
```

```
echo 'ўзгарувчи $expand ҳам $either қўйилмайди';
```

```
?>
```

### Array (массив) типи.

*PHP* дастурлаш тилида **массив** типи тартибланган карталарга ўхшайди ва қийматини калитга ўзлаштирадиган типдир. Бу тип бир неча йўналишларда оптималлаштирилади, шунинг учун сиз уни хусусий *массив*,

рўйхат (вектор), хеш-жадвали (картани амалга ошириш учун ишлатилади), стэк, навбат ва бошқалар сифатида фойдаланишингиз мумкин. Модомики, *PHP* дастурлаш тилида бир массивни қийматини бошқасига ўзлаштириш учун дарахтлардан фойдаланасиз.

*Массивларни array()* конструкцияси ёрдамида аниқланади ёки элементларига қиймат бериш билан аниқланади.

### **array()** конструкцияси ёрдамида аниқлаш.

```
array ([key] => value, [key1] => value1, ... )
```

*PHP* дастурлаш тилининг *array()* конструкцияси вергул билан ажратилган жуфт параметрлар *калит* => *қиймат* билан ажратилган. => белги мос равишда қиймат ва унинг калити ўртасида алоқа ўрнатади. Калит бутун сон бўлиши мумкин, унинг қиймати эса *PHP* дастурлаш тилидаги ихтиёрий типни қабул қилиши мумкин. Калит рақамини биз кўпинча индекс деб атаيمиз. *PHP* дастурлаш тилида индекслаш нолдан бошланади. Массив элементининг қийматини олиш учун массив номи ва квадрат қавс ичида унинг калити кўрсатилиши керак. Агар массив калити стандарт бутун сон бўлса, у ҳолда унинг қийматини бутун сон деб қараса бўлади, акс ҳолда у сатр деб қаралади. Шунинг учун `$a["1"]` ёзув `$a[1]` ёзувга тенг кучли, `$a["-1"]` ёзув эса `$a[-1]` ёзувга тенг кучли.

```
<?php
$books = array("php" =>"PHP users guide",12 => true);
echo $books["php"];
//экранга чиқаради: "PHP users guide"
echo $books[12]; //экранга чиқаради: 1
?>
```

### **Мисол. PHP** дастурлаш тилида массивлар.

Агарда элемент учун калит берилмаган бўлса, у ҳолда калит сифатида калитнинг максимал қийматига бир қўшиб ҳисобланади. Агарда қиймати мавжуд калит кўрсатилган бўлса, у ҳолда шу калит қийматини экранга чиқаради. *PHP* 4.3.0 дастурлаш тили версиясидан бошлаб калитнинг максимал қиймати манфий сон деб қаралса, у ҳолда массивнинг кейинги калити ноль (0) бўлади.

```
<?php
// $arr ҳамда $arr1 массивлар эквивалентдир.
$arr = array(5 => 43, 32, 56, "b" => 12);
$arr1 = array(5 => 43, 6 => 32,
             7 => 56, "b" => 12);
?>
```

### **Мисол. PHP** дастурлаш тилида массивлар.

Агарда `TRUE` ёки `FALSE` калит сифатида қўлланилса, у ҳолда унинг қиймати мос равишда *integer* типининг бир ва нолига ўзлаштирилади. Агар `NULL` дан фойдаланилса, у ҳолда калит ўрнига бўш

сатр ҳосил бўлади. Бу бўш сатрни калит сифатида фойдаланса бўлади, аммо уни кўштирноққа олиш керак бўлади. Бу усул бўш квадрат қавсни ишлатиш каби эмас. Массивлар ёки объектлар калити сифатида фойдаланиш мумкин ҳам эмас.

### Квадрат қавс синтаксиси ёрдамида аниқлаш.

Массивга қиймат бериш орқали массив яратиш мумкин. Биз юқорида айтиб ўтганимиздек, массив элементи қийматига эга бўлиш учун квадрат қавс ичига унинг калити кўрсатилиши керак, масалан, `$book["php"]`. Агарда янги калит ва янги қиймат кўрсатсангиз қуйидагича бўлади: `$book["new_key"]="new_value"` ҳамда массивга янги элемент қўшилади. Агарда калитни кўрсатмай фақат қийматни ўзлаштирсак, яъни `$book[]="new_value"`, у ҳолда массивга янги элемент қўшилади ва уни калити мавжуд максимал қийматга бир қўшилади. Агарда биз қиймат берган массив яратилмаган бўлса, у ҳолда биз қиймат бергандан кейин у *яратилади*.

```
<?
```

```
$books["key"]= value; // key калити билан value қиймат $books массивига қўшилади
```

```
$books[] = value1; /* 13-калит билан value1 қиймати массивга қўшилади, чунки бизда калитнинг максимал қиймати 12 эди. */
```

```
?>
```

Массивнинг аниқ бир элементини ўзгартириш учун унинг шу калити билан янги қийматга ўзлаштириш керак. Массив элементи калитини ўзгартириш мумкин эмас, фақат ўчириш (калит ва элементи жуфтлигини) ва янги қўшиш мумкин холос. Массив *элементини ўчириш* учун `unset()` функциясидан фойдаланиш керак.

```
<?php
```

```
$books = array ("php" =>"PHP users guide",12 => true);
```

```
$books[] = "Book about Perl"; /* 13-калит(индекс) билан янги элемент қўшилди, бу қуйидагига эквивалент $books[13]="Book about Php"; */
```

```
$books["lisp"] = 123456; /* Бу массивга янги "lisp" калитли 123456 қиймали янги элемент қўшиш*/
```

```
unset($books[12]); // Бу 12-калитли элементни массивдан ўчириш
```

```
unset ($books); // массивни бутунлай ўчириш
```

```
?>
```

Бўш квадрат қавсдан фойдаланганда калитнинг максимал қиймати массивда мавжуд охирига қайта индексланган калитлар орасидан кидирилади. Массивни `array_values()` функцияси ёрдамида **қайта индекслаш** мумкин.

```
<?php
```

```
$arr =
```

```
array ("a","b","c"); /* "a", "b" ва "c" қийматли массивни яратамиз. Бу ерда калит кўрсатилмаган бироқ мос равишда улар 0,1,2 бўлади. */
```

```
print_r($arr); // массивни экранга чиқарамиз (калит ва қийматини)
```

```
unset($arr[0]);
```



```

unset($arr[1]);
unset($arr[2]);
// массивдан ҳамма элементини ўчирамиз
print_r($arr); // массивни экранга чиқарамиз (калити ва қийматини)
$arr[] = "aa"; // массивга янги элемент қўшамиз. Уни индекси(калити) 3
бўлади, 0 эмас.
print_r($arr);
$arr = array_values($arr); // массивни қайта индекслаймиз.
$arr[] = "bb"; // бу элементни калити 1 бўлади.
print_r($arr);
?>

```

**Мисол.** Массивни қайта индекслаймиз.

Бу скриптнинг натижаси қуйидагича бўлади:

```
Array ( [0] => a [1] => b [2] => c )
```

```
Array ( )
```

```
Array ( [3] => aa )
```

```
Array ( [0] => aa [1] => bb )
```

### **Object (объектлар) типи.**

**Объектлар** – объектга йўналтирилган дастурлашдан кириб келган *маълумот типидир*. Объектга йўналтирилган дастурлаш тамойилига кўра, синф – аниқ хоссаларга эга ва улар билан ишлайдиган методли объектлар тўплами. Объект эса мос равишда синф нусхасидир. Масалан, дастурчилар – бу дастурни тузувчи, компьютер адабиётларини ўрганадиган одамлар синфи ва бундан ташқари ҳамма одамлар қатори исм ва фамилияси мавжуд. Энди агарда бирор аниқ дастурчи – Азамат Бобоевни олсак, у ҳолда уни шу хоссага эга бўлган дастурчи синфини объекти сифатида қараш мумкин ва у ҳам дастур тузади, ҳамда исми мавжуд ва бошқалар.

*PHP* дастурлаш тилида *объект методига* мурожаат -> *амалидан* фойдаланилади. Объектни инициализация қилишда *объектни ўзгарувчан нусхасини* яратадиган *new* ифодасидан фойдаланилади.

**Мисол.** PHP дастурлаш тилида объектлар.

```

<?php
class Person { // PHP дастурлаш тилини ўрганадиган одам методи
    function know_php()
    {
        echo "Энди мен PHP дастурлаш тилини биламан!";
    }
}
$bob = new Person; // одам синфини объектини яратамиз.
$bob -> know_php(); // уни PHP тилига ўргатамиз.
?>

```

### Resource (ресурслар) типи.

*Ресурс* – бу ташки ресурсга (масалан, маълумотлар базаси билан боғланиш) ссылка орқали боғланган махсус ўзгарувчидир. Ресурслар махсус функциялар (масалан, `mysql_connect()`, `pdf_new()` ва шунга ўхшашлар) ёрдамида яратилади ва фойдаланилади.

### Null типи.

Махсус *NULL* қиймати *ўзгарувчини* қийматга эга эмаслиги ҳақида огоҳлантиради.

*Ўзгарувчи NULL* қиймат қабул қилади, агарда:

- унга *ўзгармас NULL* (`$var = NULL`) ўзлаштирилган бўлса;
- унга ҳеч қандай қиймат берилмаган бўлса;
- у *unset()* функция ёрдамида тозаланган бўлса.

*NULL тип*ли фақат битта қиймати мавжуд – регистрга сезгирлиги кичик *NULL* калит сўзидир.

### Масаланинг ечилиши.

Энди бўлимнинг бошида қўйилган масалага қайтсак. У турли сабаблар бўйича ҳар хил одамларга тузилган мактубни жўнатишдан иборат эди. Бу масалани ҳал этиш учун ўрганилган воситалардан – *ўзгарувчилар*, *амаллар*, *ўзгармаслар*, *сатрлар* ва *массивлардан* фойдаланишга ҳаракат қиламиз. Кўрсатилган мактуб қабул қилувчига боғлиқ равишда мурожаат ва ҳолати ўзгаради, шунинг учун табиий равишда бу катталиқни *ўзгарувчи* деб белгилаймиз. Бундан ташқари ҳодисалар ва одамлар кўп, шунинг учун *массив ўзгарувчи тип*дан фойдаланиш қулай. Мактуб матни ҳар доим ўзгармас, шунинг учун уни *ўзгармас* деб бериш мақсадга мувофиқдир. Жуда узун ва кўпол сатрларни ёзмаслик учун сатрлар *конкатенация*(қўшиш) амалидан фойдаланамиз. Шундай қилиб, қуйидагига эга бўламиз:

```
<?
// бизнинг ёзувимиз ўзгармас бўлсин.
define("SIGN","Ҳурмат билан, Азамат");
// одамлар ва ҳодисалар массивини берамиз
$names = array("Иван Иванович",
               "Петр Петрович",
               "Семен Семенович");
$events = array(
    "f" => "очик эшиклар куни",
    "o" => "кўргазманинг очилиши",
    "p" => "битирувчилар бали");

// таклифнома матнини тузамиз.
$str = "Ҳурматли, $names[0]";
$str .= "<br> Сизни таклиф этамиз ";
$events["f"];
$str .= "<br>" . SIGN;
echo $str; // матнни экранга чиқарамиз.
```

?>

## Шарт операторлари if оператори.

Бу *PHP* дастурлаш тилидаги барча дастурлаш тиллари каби жуда муҳим оператордир. У шартга боғлиқ равишда код фрагментини бажаришга мўлжалланган. *if* операторининг структурасини қуйидагича ифодалаш мумкин:

*if* (ифода) *бажариладиган\_блок*

Бу ерда ифода *PHP* дастурлаш тилидаги ихтиёрий тўғри ифодадир (яъни бирор қийматга эга). Скрипти қайта ишлаш жараёнида ифода мантиқий типга ўзлаштирилади. Агар натижада қайта ишланган ифода қиймати рост (True) бўлса, у ҳолда *бажариладиган\_блок* бажарилади. Акс ҳолда *бажариладиган\_блок* йўқотилади. Агарда *бажариладиган\_блок* бир нечта буйруқлардан иборат бўлса, у ҳолда улар фигурали қавсларга { } олиниши шарт.

Ифодани мантиқий типга ўзлаштириш қоидаси:

1. FALSE қиймати қуйидаги қийматларга тўғри келади:
  - мантиқий False
  - бутун сон – ноль (0)
  - ҳақиқий сон – ноль (0.0)
  - бўш сатр ва "0" сатр;
  - элементларсиз массив
  - ўзгарувчиларсиз объект (объектлар ҳақида кейинги бўлимларда тўлиқ маълумот берилган)
  - махсус тип NULL бўлганда
2. Қолган барча қийматларда TRUE қийматга ўзлаштирилади.

**Мисол.** *if* шарт оператори.

<?

```
$names = array("Карим", "Салим", "Содиқ");
if ($names[0]=="Карим") {
    echo "Салом, Азамат!";
    $num = 1;
    $account = 2000;
}
if ($num) echo "Карим рўйхатда биринчи!";
$bax = 30;
if ($account > 100*$bax+3)
    echo "Бу сатр экранга чиқмайди, чунки шарт бажарилмайди";
?>
```

## **else оператори.**

Биз юқорида фақат *if* операторининг асосий қисминигина кўрдик. Бу операторнинг бир нечта кенгайган шакли мавжуд. *else* оператори *if*

операторида текширилаётган ифода нотўғри бўлган ҳолатдагина кенгайтиради ҳамда бу ҳолатда янги шартда бирор амал бажаради.

*else* оператори ёрдамида кенгайтирилган *if* операторининг структурасини қуйидагича ифодалаш мумкин:

*if* (ифода) *бажариладиган\_блок*

*else* *бажариладиган\_блок1*

Бу *if...else* конструкцияси қуйидагича интерпретация қилиниши мумкин: агар шарт бажарилса (яъни ифода=true), у ҳолда *бажариладиган\_блок*даги амаллар бажарилади, акс ҳолда *бажариладиган\_блок1*даги амаллар бажарилади. *else* операторидан фойдаланиш мажбурий эмас.

Юқоридаги мисолни бажарилмайдиган шарт ҳолатида қандай кўриниш олишини кўриб чиқайлик.

**Мисол.** *else* оператори.

```
<?
```

```
$names = array("Карим","Салим","Содик");
```

```
if ($names[0]=="Карим") {
```

```
    echo "Салом, Азамат!";
```

```
    $num = 1;
```

```
    $account = 2000;
```

```
} else {
```

```
    echo "Салом, $names[0]. Биз Азаматни кутгандик :(";
```

```
}
```

```
if ($num) echo " Карим рўйхатда биринчи!";
```

```
else echo " Карим рўйхатда биринчи эмас?!";
```

```
$bax = 30;
```

```
if ($account > 100*$bax+3)
```

```
    echo " Бу сатр экранга чиқмайди, чунки шарт бажарилмайди ";
```

```
    else echo "Шундай бўлса ҳам экранга чиқди!";
```

```
?>
```

### **else if оператори.**

*if* шарт операторининг яна бир кенгайган шакли – бу *elseif* операторининг қўлланилишидир. *elseif* – бу *else* ҳамда *if* операторларининг комбинациясидир. У худди *else* оператори каби *if* операторида шарт бажарилмаган ҳолда кенгайтиради. Бироқ *else* операторидан фарқи бир-бирига зид амалларни фақат агарда *elseif* шарт рост бўлгандагина бажаради. *else* ҳамда *elseif* операторлари ёрдамида кенгайтирилган *if* операторининг структурасини қуйидагича ифодалаш мумкин:

*if* (ифода) *бажариладиган\_блок*

*elseif* (ифода1) *бажариладиган\_блок1*

.....

*else* *бажариладиган\_блокN*

*elseif* операторлари битта *if*-блокида бир неча марта учраши мумкин. *elseif* тасдиғи фақат олдинда турган *if*-шартлари ҳамда *elseif*-

шартлари `False` қийматни, берилган *elseif*-шарти эса `True` қийматни қайтаргандагина бажарилади.

### Мисол. *elseif* оператори.

```
<?
$names = array("Карим", "Салим", "Содиқ");
if ($names[0]=="Карим") {
    // массивда биринчи элемент Карим бўлса
    echo "Салом, Карим!";
}elseif ($names[0] == "Салим"){
    // массивда биринчи элемент Карим эмас Салим бўлса
    echo "Салом, Салим!";
}elseif ($names[0] == "Содиқ"){
    // массивда биринчи элемент ҳам Карим, ҳам Салим эмас Содиқ бўлса
    echo "Салом, Содиқ!";
}else {
// массивда биринчи элемент на Карим, на Салим, на Содиқ бўлса
    echo "Салом, $names[0]. Сен кимсан?";
}
?>
```

### Альтернатив синтаксислар.

*PHP* дастурлаш тили ўзининг бир нечта *if*, *while*, *for*, *foreach* ҳамда *switch* бошқариладиган структуралари учун альтернатив синтаксисни тақдим этади. Ҳар бир ҳолатда очиладиган қавс икки нуқтага (:), ёпиладигани эса мос равишда *endif*;, *endwhile*; ва ҳоказоларга ўзгартирилади.

Масалан, *if* шарт оператори синтаксисини қуйидагича тфодалаш мумкин:

*if* (ифода) : бажариладиган\_блок *endif*;

Маъноси ўзгармасдан қолади: агар *if* шарт оператори думалоқ қавси ичидаги шарт рост бўлса, икки нуқтадан «:» то *endif*; буйруғигача барча код бажарилади. Бундай синтаксисдан фойдаланиш *html*-код ичида қурилган *php*-код учун қулайдир.

### Мисол. Альтернатив синтаксисдан фойдаланиш.

```
<?php
$names = array("Карим", "Салим", "Содиқ");
if ($names[0]=="Карим"):
?>
Салом, Карим!
<?php endif ?>
```

Агарда *else* ҳамда *elseif* конструкцияларидан фойдаланилса, у ҳолда ҳам альтернатив синтаксисдан фойдаланса бўлади:

```
<?php
if ($a == 5):
    print "a ўзгарувчи 5 га тенг";
```

```

    print "...";
elseif ($a == 6):
    print "a ўзгарувчи 6 га тенг ";
    print "!!!";
else:
    print "a ўзгарувчи на 5 га ва на 6 га тенг ";
endif;
?>

```

### switch оператори.

Яна бир шартни текшириб турли амалларга боғлиқ равишда иш кўрсатадиган конструкция бу – *switch* операторидир. Бу операторни узбек тилига таржима қилинганда “йўналишни ўзгартиргич” маъносини беради ҳамда бу операторнинг вазифаси ҳам шунга ўхшашдир. Ўзгарувчини қандай қийматни қабул қилишига боғлиқ равишда у йўналишни ўзгартириб турли блоклардаги амалларни бажаради. *switch* оператори *if...elseif...else* ёки *if* оператори мажмуига жуда ўхшаш бўлади. *switch* операторининг структурасини куйидагича ифодалаш мумкин:

```

switch (ифода ёки ўзгарувчи){
case қиймат1:
    амаллар_блоки1
break;
case қиймат2:
    амаллар_блоки2
break;
...
default:
    амаллар_блоки_автоматик_тарзда
}

```

*if* операторидан фарқли томони бу ерда ифодалар мантиқий тип қабул қилмай, балки фақат *case* калит сўзидан кейинги қийматларни (*қиймат1*, *қиймат2* ва ҳ.к.) таққослайди холос. Агар ифода қиймати қандайдир вариант билан устма-уст тушса, икки нуқтадан кейинги *break* операторигача бўлган *амаллар\_блоки*даги амалларни бажаради. Агарда ифода қиймати берилган вариантлардан ҳеч бирига устма-уст тушмаса, *default* калит сўзидан кейинги автоматик тарзда бажариладиган блок (*амаллар\_блоки\_автоматик\_тарзда*) бажарилади. *switch* операторидаги ифода фақат бир марта ҳисобланади, *elseif* операторида эса ҳар бир текширишда ҳисобланади, шунинг учун агарда ифода етарли даражада мураккаб бўлса, у ҳолда *switch* оператори тезроқ ишлайди.

мисолни *switch* операторидан фойдаланган ҳолда куйидагича ёзиш мумкин:

<?

```

$names = array("Карим", "Салим", "Содиқ");
switch ($names[0]){
case "Карим":
    echo "Салом, Карим!";
break;
case "Салим":
    echo "Салом, Салим!";
break;
case "Содиқ":
    echo "Салом, Содиқ!";
break;
default:
    echo "Салом, $names[0] ";
}
?>

```

Агарда берилган мисолда *break* операторини ташлаб кетсак, масалан, *case* "Салим": ҳоли учун, у ҳолда агарда ўзгарувчи сатр қиймати "Салим" бўлса экранга "Салом, Салим!" маълумотини чиқаради ва ишини давом эттириб "Салом, Содиқ!" маълумотни чиқаради ва *switch* операторининг охирига келиб дастур *break* операторини бажаради.

*switch* операторининг констркуцияси учун худди *if* оператори каби альтернатив синтаксиси мавжуд. Бу ерда *switch* операторидаги очиладиган фигурали кавс икки нуктага ўзгартирилади, ёпиладигани эса мос равишда *endswitch*; калит сўзига ўзгартирилади.

## ЦИКЛЛАР.

*RНР* дастурлаш тилида шартга боғлиқ равишда қайтариладиган амаллардан иборат бир нечта конструкциялар мавжуд. Бу *while*, *do..while*, *foreach* ҳамда *for* цикллاردир. Уларни батафсил кўриб чиқамиз.

### **while**

Структураси:

```
while (ифода) { бажариладиган_блок }
```

ёки

```
while (ифода): бажариладиган_блок endwhile;
```

*while* – бу оддий цикл. У ифода қиймати True (бу ерда худди *if* оператори каби ифода мантиқий типга ўзлаштирилади) бўлгунча *бажариладиган\_блок*даги буйруқларни бажаришга буюради. Ифода қиймати ҳар цикл бошланганда текшириб борилади, агарда унинг қиймати *бажариладиган\_блок* бажарилиш жараёнида ўзгарган тақдирда ҳам итерация тугамагунча (яъни *бажариладиган\_блок*даги барча буйруқлар бажарилмагунча) цикл тўтатилмайди.

### Мисол. while оператори

```
<? // Бу дастур барча жуфт сонларни экранга чиқаради.  
  $i = 1;  
  while ($i < 10) {  
    if ($i % 2 == 0) print $i;  
    // агар у жуфт бўлса экранга чиқаради.  
    $i++;  
    // $i ўзгарувчи биттага оширилади  
  }  
?>
```

### do... while

*do..while* цикли *while* циклга ўхшайди, аммо фарқли томони шундаки, ифоданинг ростлигига цикл бошида эмас, балки охирида текширилади. Қулай томони шундаки, *бажариладиган\_блок do..while* цикли ичида ҳеч бўлмаганда бир марта бажарилади.

Структураси:

```
do { do..while цикли } while (ифода);
```

### Мисол. do..while оператори.

```
<?  
// бу дастур шарт бажарилмаса ҳам 12 рақамини экранга чиқаради.  
$i = 12;  
do{  
  if ($i % 2 == 0) print $i;  
  // агар сон жуфт бўлса уни экранга чиқарамиз  
  $i++;  
  // сонни биттага оширамиз  
}while ($i<10)  
?>
```

### for

Бу *PHP* дастурлаш тилидаги энг мураккаб циклдир. Улар *C* дастурлаш тилидаги циклларни эслатади.

Структураси:

```
for (ифода1; ифода2; ифода3) { бажариладиган_блок }  
ёки
```

```
for (ифода1; ифода2; ифода3): бажариладиган_блок endfor;
```

Бу ерда кўриниб турибдики шар учта ифодадан ташкил топади. Биринчи *ифода1* ифода цикл бошида шартсиз бажарилади. Ҳар бир итерациянинг бошланишида *ифода2* бажарилади. Агар у *True* қийматни қабул қилса, у ҳолда цикл ўз ишини давом эттиради ва *бажариладиган\_блок*даги барча буйруқларни бажаради. Агар *ифода2* *False* қийматни қабул қилса, у ҳолда цикл тўхтатилади. Ҳар бир итерация (яъни *бажариладиган\_блок*даги барча буйруқларни бажарилишидан кейин) охирида *ифода3* бажарилади.



Ҳар бир 1-,2- ва 3-ифодалар бўш бўлиши мумкин. Агар *ифода2* бўш бўлса, бу циклни чексиз (бу ҳолда *PHP* дастурлаш тили бу ифодани ҳар доим рост деб ҳисоблайди) бажарилишини билдиради. Бу унчалик бефойда эмас, чунки циклни *break* оператори ёрдамида тўхтатса бўлади.

Масалан, барча жуфт сонларни *for* цикли ёрдамида куйидагича экранга чиқариш мумкин:

```
<?php
for ($i=0; $i<10; $i++){
    if ($i % 2 == 0) print $i;
    // жуфт сонларни экранга чиқарамиз
}
?>
```

Агарда иккинчи ифодани ( $i < 10$  шартни) ташлаб кетсак, бу масаладаги циклни ҳам *break* оператори ёрдамида тўхтатса бўлади.

```
<?php
for ($i=0; ; $i++){
    if ($i>=10) break;
    // агар $i катта ёки тенг 10 бўлса, у ҳолда цикл ишини тўхтатамиз.
    if ($i % 2 == 0) print $i;
    // агар сон жуфт бўлса, уни экранга чиқарамиз.
}
?>
```

Барча учала ифодани ҳам тушириб қолдириш мумкин. Бу ҳолда **счётчик**  $i$  ўзгарувчини бошланғич қиймати берилмайди ва ҳар бир цикл охирида у ўзгармайди. Бу барча буйруқларни алоҳида буйруқлар кўринишида ёки циклдан аввал *бажариладиган\_блок* ичида ёзса ҳам бўлади:

```
<?php
$i=0; // счётчикни бошланғич қийматини берамиз
for ( ; ; ){
    if ($i>=10) break;
    // агар $i катта ёки тенг 10 бўлса, у ҳолда цикл ишини тўхтатамиз.
    if ($i % 2 == 0) print $i;
    // агар сон жуфт бўлса, уни экранга чиқарамиз.
    $i++; // счётчик қийматини биттага оширамиз.
}
?>
```

*for* цикли конструкциясидаги учинчи ифодада вергулдан кейин яна бир нечта оддий буйруқларни ҳам ёзса бўлади. Масалан, агар биз оддийгина барча сонларни экранга чиқармоқчи бўлсак, дастурни куйидагича ёзса бўлади:

```
<?php
for ($i=0; $i<10; print $i, $i++)
/* Агарда бажариладиган_блок буйруқлардан ташкил топмаган ёки битта буйруқдан ташкил топган бўлса,
```

фигурали қавсга олинган қисмни  
ташлаб кетса бўлади.\*/  
?>

### **foreach**

Яна битта фойдали конструкция. У фақат *PHP4* дастурлаш тилида  
учрайди ва массивлар билан ишлашга мўлжалланган.

Синтаксиси қуйидагича:

```
foreach ($array as $value) { бажариладиган_блок }  
ёки  
foreach ($array as $key => $value)  
{ бажариладиган_блок }
```

Биринчи ҳолда берилган `$array` ўзгарувчи массивнинг  
элементлари формаллаштирилади. Ҳар бир цикл қадамида  
массивнинг жорий элементи қиймати `$value` ўзгарувчига  
ўзлаштирилади ва массивнинг ички ҳисоблагичи биттага  
ортади (чунки кейинги қадамда массивнинг кейинги  
элементи керак бўлади). *бажариладиган\_блок* ичидаги  
массивнинг жорий элементи қиймати `$value` ўзгарувчи  
ёрдамида қийматга эга бўлади. *бажариладиган\_блок* `$array`  
массивнинг элементлари нечта бўлса шунча марта бажарилади.

Юқорида келтирилган иккинчи тўлдирилган шаклда ҳар бир цикл  
қадамида массивнинг жорий элементи калити *бажариладиган\_блок*да  
қўлланса ҳам бўладиган `$key` ўзгарувчига ёзиб  
борилади.

*foreach* цикли ишини бошлаганда массивнинг ички кўрсатгичи  
автоматик равишда биринчи элементни кўрсатади.

**Мисол.** *foreach* оператори.

```
<?php  
$names = array("Карим","Салим","Содик");  
foreach ($names as $val) {  
    echo "Салом, $val <br>";  
    // барча саломлашишларни экранга чиқарамиз  
}  
foreach ($names as $k => $val) {  
    // саломлашишдан ташқари рўйхатдаги рақамини, яъни калитини экранга  
    чиқарамиз  
    echo "Салом, $val ! Сен рўйхатда $k – рақамдасан.<br>";  
}  
?>
```

### **Бошқарув ўтказувчи операторлар**

Баъзида цикл ёки унинг алоҳида итерация ишини тезда тўхтатишга  
тўғри келади. Бунинг учун *break* ҳамда *continue* операторлари керак бўлади.

### **Break**

Break оператори мавжуд циклни амалга оширишни тугаллайди, *for*, *foreach*, *while*, *do while* ёки *switch break* структурани бошқарувчи, тугаллаш кераклигини билдирувчи, унинг таркибига кирувчи рақамли аргумент билан қўлланилади.

**Мисол.** Break оператори

```
<?php
$i=1;
while ($i) {
    $n = rand(1,10);
    // исталган сонни умумийлаштирамиз 1 дан 10 гача
    echo "$i:$n ";
    // итерация рақамини чиқарамиз ва умумийлаштирилган сон
    if ($n==5) break;
/* Агар умумийлаштирилган сон 5 бўлса, цикл ишини тўхтатамиз. Бу ҳолда
бу қатордан кейин цикл ичида нима мавжуд бўлса, амалга оширилмайди */
    echo "Цикл ишламоқда <br>";
    $i++;
}
echo "<br> итерация цикли сони $i ";
?>
```

Бу скрипт ишининг натижаси қуйидагича:

1:7 Цикл ишляпти

2:2 Цикл ишляпти

3:5

Цикл итерацияси сони

Агар *break* операторидан сўнг сон кўрсатилса, бу цикл операторларидан таркиб топган айнан шундай миқдор бузилади. Модомики, юқорида келтирилган мисолда циклдан фойдаланилмаган экан, бу унчалик тўғри эмас. Скриптимизни бироз ўзгартирамиз:

```
<?php
$i=1;
while ($i) {
    $n = rand(1,10);
    // Исталган сонни умумлаштирамиз 1 дан 10 гача
    switch ($n){
        case 5:
            echo "<font color=blue>
                switch дан чиқиш (n=$n)</font>";
            break 1;
        // switch ишини тўхтатамиз (break мавжуд биринчи циклни)
        case 10:
            echo "<font color=red>switch дан чиқиш ва while (n=$n)</font>";
            break 2;
        // switch ишини тўхтатамиз ва while (иккита break мавжуд цикл)
```

```

default:
    echo "switch ишляпти (n=$n), ";
}
echo " while ишляпти -$i <br>" амал;
$i++;
}
echo "<br>цикл итерацияси сони $i ";
?>

```

### **continue**

Баъзан цикл ишини бутунлай тўхтатиш лозим бўлмайди, фақатгина унинг янги итерациясини бошлаш керак. *Continue* оператори исталган циклни амалга ошириш блокидан кейинги инструкцияларни ўтказиб юбориш ва янги доира билан амалга оширишни давом эттириш имконини беради. *continue* ни унинг таркибида бошқарилувчи конструкциялар ишини якунлаш кераклигини кўрсатувчи рақамли аргумент тарзида ишлатиш мумкин.

**Олдинги параграфда берилган мисолдаги *break* операторини *continue* га алмаштирамиз. Бундан ташқари тўрт цикли миқдорини камайтирамиз.**

```

<?php
$i=1;
while ($i<4) {
    $n = rand(1,10);
    // исталган сонни умумлаштирамиз 1 дан 10 гача
    echo "$i:$n ";
    // интерация рақамини чиқарамиз ва умумлаштирилган сон
    if ($n==5) {
        echo "Янги интерация ";
        continue;
    }
}

```

```

/*Агар умумлаштирилган сон 5 бўлса, янги цикл интерациясини бошлаймиз,
$i катталашмайди */
}
echo "Цикл ишляпти <br>";
$i++;
}
echo "<br>цикл интерацияси сони $i ";
?>

```

Бу скрипт ишининг натижаси қуйидагича

```

1:10 Цикл ишляпти
2:5 Янги итерация
2:1 Цикл ишляпти
3:1 Цикл ишляпти
4 цикли интерацияси сони

```

*continue* оператори амалга ошгандан сўнг цикл иши тугалланмаганини ҳисобга оламиз. Мисолда цикл ҳисоблагичи 5 сони олингандан тақдирда у *continue* операторидан кейин бўлса ўзгармайди. Аслида *continue* ёрдамида 5 сони умумлаштирилганда бу ҳолатдан четлашамиз. Шунинг учун *continue* операторини ифода ҳақиқийлигини текширишга алмаштириб ёзиб қўйиш кифоя:

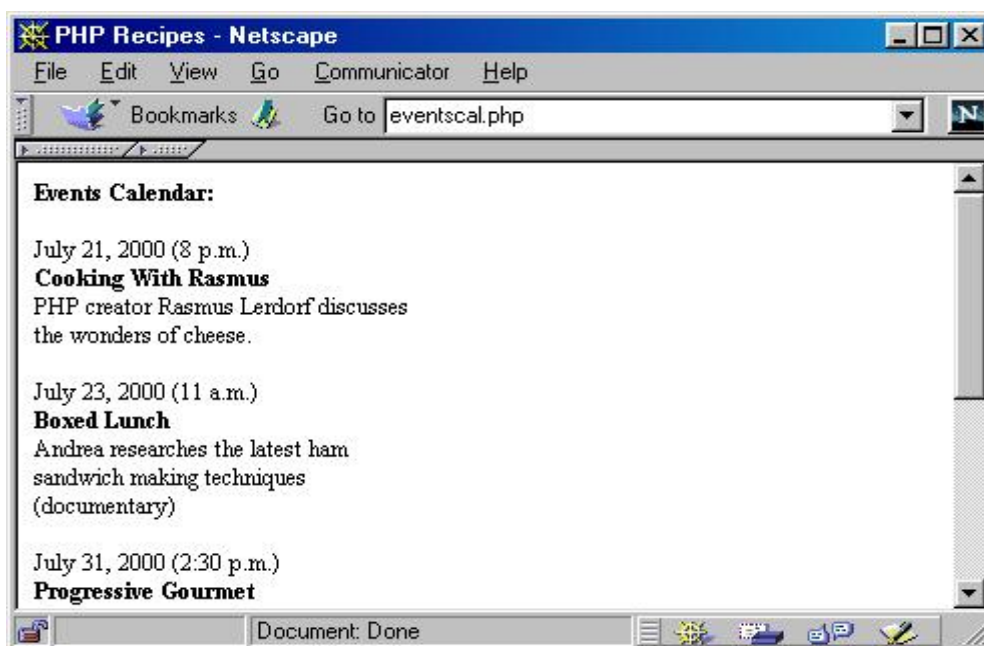
```
<?php
$i=1;
while ($i<4) {
    $n = rand(1,10);
    // исталган сонни умумлаштирамиз 1 дан 10 гача
    if ($n!=5) {
        echo "$i:$n <br>";
    }
    // интерация рақамини чиқарамиз ва умумлаштирилган сон
    $i++;
}
?>
```

PHP да *continue* дан фойдаланишнинг бир хусусияти мавжуд – *switch* конструкциясида у худди *break* каби ишлайди. Агар *switch* цикл ичида жойлашган бўлса ва янги цикл интерациясини бошлаш керак бўлса, *continue* 2 дан фойдаланиш лозим бўлади.

## Назорат саволлари

### Проект: календар

1. Қуйидаги расмда келтирилган натижани берувчи скрипт тузинг



### **Фойдаланилган адабиётлар.**

1. VG Olifer, N. A.Olifer. Computer networks. Principles, technologies, protocols. St. Petersburg: Publishing House. "Peter," in 2000.
2. R. Darnell, "JavaScript: Handbook." - St. Petersburg: Publishing House. "Peter," in 1998.
3. Duvanov A.A. WEB. Construction of HTML. - St. Petersburg: BHV, 2003.
4. R. Eisenmenger. HTML 3.2/4.0. Handbook. - M: Univ. "Bean", 1998.
5. Stephen Speynaur, Valerie Kuersia. Directory WEB-master. - Ed. BHV
6. Barrett D. JavaScript. Web-professionals. - Kiev: BHV - Kyiv, 2001.
7. Brandenbau D. JavaScript: a collection of recipes. - St. Petersburg.: Peter, 2000.

V. БЎЛИМ

КЕЙСЛАР БАНКИ

## V. КЕЙСЛАР БАНКИ

**1-Кейс.** Видео плеер вазифасини бажарувчи веб илова ёзилди. Илова ишга туширилганда видео тасвир экранда намоён бўлмади. Яъни илова кутилган натижани бермади.

**Кейсни бажариш босқичилари ва топшириқлар:**

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).

- Веб иловани ишга тушириш учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш)

**2-Кейс.** Дастурчи Веб иловага You Tube сайтидаги бирор видео иловани жойлаштирамоқчи бўлди. Яъни шу илова ишга тушганда илова тугридан тугри You Tube сайтидан видео олиб кўрсатиши керак эди аммо илова кутилган натижани бермади.

**Кейсни бажариш босқичилари ва топшириқлар:**

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).

- Веб иловани ишга тушириш учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш)

**3-Кейс.** Дастурчи доира узунлиги 180 пи бўлган чегараси 8 пиксел бўлган айлана чизишда ўзи билмаган баъзи хатоликлар қилди. Натижада айлана тасвири тўлиқ намоён бўлмади. Илова кутилган натижани бермади.

**Кейсни бажариш босқичилари ва топшириқлар:**

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).

- Веб иловани ишга тушириш учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш)

**4-Кейс.** Дастурчи веб саҳифани фонини ўзгартирмоқчи бўлди аммо буни амалга ошира олмади. Илова кутилган натижани бермади.

**Кейсни бажариш босқичилари ва топшириқлар:**

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).

- Веб иловани ишга тушириш учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш)

**5-Кейс.** Дастурчи веб саҳифадаги ёзувни бир хил интервалда жойлаштирамоқчи бўлди аммо буни амалга ошира олмади. Илова кутилган натижани бермади.



### **Кейсни бажариш босқичилари ва топшириқлар:**

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).

- Веб иловани ишга тушириш учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш)

**6-Кейс.** Дастурчи веб саҳифага гуруҳдаги барча талабаларнинг фото суратларини жойлади. Жорий ҳолатда фақат 4 та талаба фотосурати кўринишини хоҳлайди, буни амалга ошира олмади. Илова қутилган натижани бермади.

### **Кейсни бажариш босқичилари ва топшириқлар:**

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).

- Веб иловани ишга тушириш учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш)

**7-Кейс.** Дастурчи веб саҳифага квадрат тенглама формуласини жойлаштира олмади бўлди аммо буни амалга ошира олмади. Илова қутилган натижани бермади.

### **Кейсни бажариш босқичилари ва топшириқлар:**

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).

- Веб иловани ишга тушириш учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш)

**8-Кейс.** Дастурчи веб саҳифага CSS ёрдамида фото суратни эллипс ичига жойлаган ҳолда жойлаштира олмади бўлди, аммо буни амалга ошира олмади. Илова қутилган натижани бермади.

### **Кейсни бажариш босқичилари ва топшириқлар:**

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).

- Веб иловани ишга тушириш учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш)

**9-Кейс.** Дастурчи веб саҳифага CSS ёрдамида фото суратни айлана ичига жойлаган ҳолда жойлаштира олмади бўлди, аммо буни амалга ошира олмади. Илова қутилган натижани бермади.

### **Кейсни бажариш босқичилари ва топшириқлар:**

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).

- Веб иловани ишга тушириш учун бажариладиган ишлар кетма-

кетлигини белгиланг (жуфтликлардаги иш)

**10-Кейс.** Дастурчи веб саҳифага CSS ёрдамида маълум узунликдаги сатрларни ҳар хил рангларда ифодалашни ва мос сатрлар устида шу сатр рангини инглизча ёзмақчи бўлди, аммо буни амалга ошира олмади. Илова қутилган натижани бермади.

**Кейсни бажариш босқичлари ва топшириқлар:**

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- Веб иловани ишга тушириш учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш).

**11-Кейс.** Дастурчи веб саҳифага CSS ёрдамида маълум тугмалар(button) ҳар хил рангларда ифодалашни ва мос тугмалар устида шу тугмалар рангини инглизча ёзмақчи бўлди, аммо буни амалга ошира олмади. Илова қутилган натижани бермади.

**Кейсни бажариш босқичлари ва топшириқлар:**

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- Веб иловани ишга тушириш учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш)

**12-Кейс.** Дастурчи веб саҳифада бир саҳифадан иккинчи саҳифага ўтказувчи код ёзди, аммо буни амалга ошира олмади. Илова қутилган натижани бермади.

**Кейсни бажариш босқичлари ва топшириқлар:**

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- Веб иловани ишга тушириш учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш) .

**13-Кейс.** Дастурчи веб саҳифага бир нечта фото суратлар ва уларнинг остига изох ёзмақчи бўлди, аммо буни амалга ошира олмади. Илова қутилган натижани бермади.

**Кейсни бажариш босқичлари ва топшириқлар:**

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- Веб иловани ишга тушириш учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш) .

**14 – Кейс.** Дастурчи HTML да веб саҳифа ёзиб CSS да унга дизайн бермақчи бўлди , аммо буни амалга ошира олмади. Илова қутилган натижани бермади.

**Кейсни бажариш босқичлари ва топшириқлар:**

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг

(индивидуал ва кичик гуруҳда).

•Веб иловани ишга тушириш учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш) .

**15 – Кейс.** Дастурчи РНР да браузерда ўзгарувчига қиймат бериб натижани саҳифада кўрмоқчи ва шунга мос код ёзмоқчи бўлди, аммо буни амалга ошира олмади. Илова кутилган натижани бермади.

**Кейсни бажариш босқичилари ва топшириқлар:**

•Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).

•Веб иловани ишга тушириш учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш)

**16 – Кейс.** Дастурчи РНР да include функциясидан фойдаланиб дастур ёзмоқчи бўлди. Яъни ўзгарувчи қийматларини бошқа файлдан ўқиб олмақчи ва унга мос код ёзмоқчи бўлди, аммо буни амалга ошира олмади. Илова кутилган натижани бермади.

**Кейсни бажариш босқичилари ва топшириқлар:**

•Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).

•Веб иловани ишга тушириш учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш).

**17–Кейс.** Дастурчи РНР да формада бир нечта маълумотлар ва парол киритди натижада у шу маълумотларни бошқа саҳифада кўрмоқчи эди, аммо буни амалга ошира олмади. Илова кутилган натижани бермади.

**Кейсни бажариш босқичилари ва топшириқлар:**

•Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).

•Веб иловани ишга тушириш учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш).

**18–Кейс.** Дастурчи РНР да формада квадрат тенглама коэффициентларини киритувчи ва тугма ёрдамида илдизларни ҳисобловчи код ёзди янги коэффициент киритганда эскилари ўз ўзидан ўчириб ташловчи дастур тузмоқчи бўлди, аммо буни амалга ошира олмади. Илова кутилган натижани бермади.

**Кейсни бажариш босқичилари ва топшириқлар:**

• Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).

• Веб иловани ишга тушириш учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш) .

**19 – Кейс.** Дастурчи РНР да формада иккита соннинг экубини ва экукини

хисобловчи функция ёзмокчи бўлди, аммо буни амалга ошира олмади. Илова кутилган натижани бермади.

**Кейсни бажариш босқичлари ва топшириқлар:**

•Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).

•Веб иловани ишга тушириш учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш)

**20–Кейс.** Дастурчи PHP дастурда бирор ўзгарувчи ва қийматини мавжуд ёки йўқлигини аниқловчи(isset ёки unset) код ёзмокчи бўлди, аммо буни амалга ошира олмади. Илова кутилган натижани бермади.

**Кейсни бажариш босқичлари ва топшириқлар:**

• Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).

•Веб иловани ишга тушириш учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш)

**Амалий топшириқлар**

**I. JavaScript тилида берилган мисолларни ечинг?**

1. Комплекс сонлар устида амаллар бажаринг (Мисол учун:  $2+3i, 3+4i \Rightarrow 5+7i$ )?
2. Иккита вақт оралиғидаги фарқни топувчи дастур тузинг?
3. Берилган натурал сон квадратини топувчи дастурни қуйидаги қонуният бўйича тузинг:  
 $1^2 = 1$   
 $2^2 = 1 + 3$   
 $3^2 = 1 + 3 + 5$   
 $4^2 = 1 + 3 + 5 + 7$   
.....  
 $N^2 = 1 + 3 + 5 + 7 + \dots + (2N-1)$
4. Қуйидаги ифодани ҳисобланг  
$$\frac{\cos 1}{\sin 1} * \frac{\cos 1 + \cos 2}{\sin 1 + \sin 2} * \dots * \frac{\cos 1 + \cos 2 + \dots + \cos n}{\sin 1 + \sin 2 + \dots + \sin n} ;$$

**II. Синфлар ва объектлардан фойдаланиб дастурлаш. ОЙД принципларидан (ворислик, полиморфизм) фойдаланинг**

1. Шакл, (айлана, тўртбурчак, учбурчак)

2. Мавжудот, (одам, ҳайвон, ҳашорат)
3. Транспорт (енгил авто, оғир авто, сувтранспорти, ҳавотранспорти)
4. Спортчи (футболчи, боксчи, теннисчи)
5. ОТМ(олий таълим муассаси: университет, институт, консерватория, коллеж, )
6. Шифокор (хирург, терапевт, онколог, психолог, генеколог)
7. Мушуксимонлар ( Йўлбарс, илвирс, мушук)
8. Ноутбук ( Dell, HP, Lenovo, ASER)

III. **Қуйидаги берилган вариантлар бўйича. JavaScript нинг танланган объекти асосида мисол келтиринг.**

1. Объект Form (document.forms[i])
2. Объекты Button, Checkbox, Hidden, Password, Radio, Reset, Submit, Text, Textarea
3. Объект Checkbox
4. Объект Radio
5. Объект Reset
6. Объекты Text и Password
7. Объект Textarea
8. Объекты Select и Option
9. Объект Object
10. Объект Number
11. Объект Boolean
12. Объект String
13. Объект Array
14. Объект Function
15. Объект JSONArray
16. Объект JavaClass
17. Объект JavaObject
18. Объект JavaPackage
19. Объект Screen

# VI. БЎЛИМ

МУСТАҚИЛ ТАЪЛИМ  
МАВЗУЛАРИ

## VI. МУСТАҚИЛ ТАЪЛИМ МАВЗУЛАРИ

### Мустақил ишни ташкил этишнинг шакли ва мазмуни

Тингловчи мустақил ишни муайян модулни хусусиятларини ҳисобга олган ҳолда қуйидаги шакллардан фойдаланиб тайёрлаши тавсия этилади:

- меъёрий хужжатлардан, ўқув ва илмий адабиётлардан фойдаланиш асосида модул мавзуларини ўрганиш;

- тарқатма материаллар бўйича маърузалар қисмини ўзлаштириш;

- автоматлаштирилган ўргатувчи ва назорат қилувчи дастурлар билан ишлаш;

- махсус адабиётлар бўйича модул бўлимлари ёки мавзулари устида ишлаш;

- тингловчининг касбий фаолияти билан боғлиқ бўлган модул бўлимлари ва мавзуларни чуқур ўрганиш.

### Мустақил таълим мавзулари

1. HTML-4 нинг HTML-5 дан фарқи.
2. CSS нинг CSS3 дан фарқи.
3. PHP ва JQuery имкониятлари.
4. PHP дастурлашда фреймворкларнинг ўрни.
5. Шаблонлар.
6. CSS контейнерлар.
7. Замонавий CMS лардан фойдаланиш.
8. Bootstrap имконияларини таҳлил қилиш.
9. CSS2 да конформлилик.
10. URL+URN=URI таҳлил қилинг.
11. Синф селекторлари.
12. MS SQL, AJAX, MySQL имкониятларини солиштириш ва таҳлил қилиш
13. PHP5 имкониятларини таҳлил қилиш.
14. JavaScript имкониятларини таҳлил қилиш.
15. AngularJS имкониятларини таҳлил қилиш.
16. Визуал эффе́ктлар

# VII. БҮЛІМ

## ГЛОССАРИЙ



## VII. ГЛОССАРИЙ

Термин	Ўзбек тилидаги шарҳи	Инглиз тилидаги шарҳи
.dex file	Компиляция қилинган Android дастури коди файли	Compiled Android application code file.
Action	иловада Intent орқали жўнатилувчи хабар	A description of something that an Intent sender wants done. An action is a string value assigned to an Intent.
Activity	илованинг биронта ойнаси (интерфейс) бошқарувчи Java файл	A single screen in an application, with supporting Java code, derived from the Activity class.
Application	бу ишлаб чиқилган илова	From a component perspective, an Android application consists of one or more activities, services, listeners, and intent receivers.
Broadcast Receiver	иловага кириб келаётган ёки ундан жўнатилаётган интент объектини бошқарувчи илова синфи	An application class that listens for Intents that are broadcast, rather than being sent to a single target application/activity.
Canvas	иловада чизилган ёки юклаб олинган тасвир	A drawing surface that handles compositing of the actual bits against a Bitmap or Surface object.
Client server	Мижоз-сервер. тармоқдаги турли дастурларнинг ишлаш схемаси. Шундай схема билан ишловчи дастурки, у икита ўзаро таъсирлашувчи	

	қисмлардан иборат: мижоз ва сервердан. Мижоз фойдаланувчининг машинасида, сервер эса унга мос серверда (компютерда) ўрнатилган бўлади. Сервер мижознинг буйруқлари бўйича маълум бир амалларни мижозга хизматлар тақдим этган ҳолда бажаради.	
Cracer	қонунсиз йўллар билан ҳимояланган ресурсларга масалан, конфиденсиал маълумотларга киришга рухсат олиб маълумотларни қидириш билан шуғулланувчи ҳисоблаш тизимининг (одатда компютер тармоғининг) фойдаланувчиси. У хакердан фарқ қилади!	
Dialog	фойдаланувчи интерфейс учун мулоқот ойнаси	A floating window that acts as a lightweight form.
Drawable	илованинг визуал ресурслари сақланувчи папка	A compiled visual resource that can be used as a background, title, or other part of the screen.
Intent	иловада ойналар/activity бўйлаб жўнатиловчи хабар объекти	An message object that you can use to launch or communicate with other applications/activities asynchronously.
Intent Filter	манифест файлида эълон қилинадиган интент объектини филтрловчи объект	A filter object that an application declares in its manifest file, to tell the system what types of Intents

		each of its components is willing to accept and with what criteria.
Layout Resource	илова ойналарининг кўринишларини сақловчи XML файл	An XML file that describes the layout of an Activity screen.
Manifest File	илова учун керакли барча маълумотларни сақловчи XML файл (мисол учун, илова номи, интент филтрлар, интернетга боғланиш)	An XML file that each application must define, to describe the application's package name, version, components (activities, intent filters, services), imported libraries, and describes the various activities, and so on.
OpenGL ES	OpenGL ES кутубхонасидан мураккаб 3D расмларни тузишда фойдаланиш мумкин	Android provides OpenGL ES libraries that you can use for fast, complex 3D images.
Resources	илова учун керакли бўлган ресурслар (расм, аудио, видео ва бошқа файллар)	Nonprogrammatic application components that are external to the compiled application code, but which can be loaded from application code using a well-known reference format.
Service	иловаорти хизматлар яратиш учун синф	An object of class Service that runs in the background (without any UI presence) to perform various persistent actions, such as playing music or monitoring network activity.
Source code	Бу бошқа ишлаб чиқарувчилардан сотиб олиш мумкин бўлган	It's a framework that you can buy from other developers and on

	framework ҳисобланади	marketplaces.
Surface	Canvas объектини бошқариш учун объект типи	An object of type Surface representing a block of memory that gets composited to the screen.
SurfaceView	Canvas объектини бошқариш учун динамик объект	A View object that wraps a Surface for drawing, and exposes methods to specify its size and format dynamically.
View	илова интерфейсида фойдаланиладиган компоненталар (button, checkbox, listview, gallery ва бошқалар)	An object that draws to a rectangular area on the screen and handles click, keystroke, and other interaction events.
Viewgroup	илова интерфейсида фойдаланиладиган компоненталар гуруҳи ёки жамланмаси	A container object that groups a set of child Views.
Web server	ўзига мурожаат қилишларни кутувчи ёки сўровлар ва сўровларга жавоб берган сервердаги дастур, яъни кўпинча талаб этилган файлни жўнатувчи дастурдир.	
Widget	илова интерфейсида фойдаланиладиган компоненталардан фойдаланиб ишлаб чиқилган график компонент	One of a set of fully implemented View subclasses that render form elements and other UI components, such as a text box or popup menu.
Window	илловани интерфейсларини яратиш учун абстракт синф	In an Android application, an object derived from the abstract class Window that specifies the elements of a

		generic window, such as the look and feel (title bar text, location and content of menus, and so on).
Universal app	Телефон қурилмаларида ҳам планшет қурилмаларида ҳам ишлайдиган илова	A single app that can run on both phone and tablet devices.
GUI	Фойдаланувчи график интерфейси	Graphic User Interface.
IDE	Қурилган ривожланиш муҳити	Integrated Development Environment.
OS (Operating System)	Операцион тизим. Қурилмадаги энг муҳим дастур	Operating System. The most important program on a device.

# VIII. БЎЛИМ

АДАБИЁТЛАР  
РЎЙХАТИ

## VIII. АДАБИЁТЛАР РЎЙХАТИ

### Махсус адабиётлар.

1. Jamsil K. Lalani, S., S. Weekly Programming Web professionals Trans. from English. A.I.Panasyuk - Mn. LLC "Potpourri", 1997. - 632 p.: Ill.
2. Malchuk EV. HTML and CSS. self-teacher.
3. VG Olifer, N. A.Olifer. Computer networks. Principles, technologies, protocols. St. Petersburg: Publishing House. "Peter," in 2000.
4. R. Darnell, "JavaScript: Handbook." - St. Petersburg: Publishing House. "Peter," in 1998.
5. Duvanov A.A. WEB. Construction of HTML. - St. Petersburg: BHV, 2003.
6. R. Eisenmenger. HTML 3.2/4.0. Handbook. - M: Univ. "Bean", 1998.
7. Stephen Speynaur, Valerie Kuersia. Directory WEB-master. - Ed. BHV
8. Barrett D. JavaScript. Web-professionals. - Kiev: BHV - Kyiv, 2001.
9. Brandenbau D. JavaScript: a collection of recipes. - St. Petersburg.: Peter, 2000.
10. Budilov B. JavaScript, XML and the Document Object Model. - St. Petersburg.: S & T, 2001.
11. R. Wagner JavaScript. Encyclopedia member (+ CD-ROM). - Kiev: DiaSoft, 2001.
12. Vayk A. JavaScript in examples. - Kiev: DiaSoft, 2000.
13. Rev. E. Vander JavaScript for "Dummies." - Dialectic, 2001.
14. P. Weiner JAVA programming languages, and JavaScript. - M: LORI, 2000.
15. Garnaev A Web-programming in Java and JavaScript. - St. Petersburg.: BHV St. Pereburg, 2002.
16. Darnell R. JavaScript. Handbook. - St. Petersburg.: Peter, 2000.
17. Dmitriev M. Tutorial JavaScript. - St. Petersburg.: BHV St. Pereburg, 2001.
18. Dmitriev M. JavaScript. Quick start. - St. Petersburg.: BHV St. Pereburg, 2002.

### Интернет ресурслар

1. <http://www.intuit.ru>
2. <http://etuit.uz/>
3. [www.tuit.uz](http://www.tuit.uz)
4. [www.ziyonet.uz](http://www.ziyonet.uz)