

МУҲАММАД АЛ-ХОРАЗМИЙ НОМИДАГИ
ТОШКЕНТ АХБОРОТ ТЕХНОЛОГИЯЛАРИ УНИВЕРСИТЕТИ
ҲУЗУРИДАГИ ПЕДАГОГ КАДРЛАРНИ ҚАЙТА ТАЙЁРЛАШ ВА
УЛАРНИНГ МАЛАКАСИНИ ОШИРИШ ТАРМОҚ МАРКАЗИ

2019



ЎҚУВ-УСЛУБИЙ
МАЖМУА

ВЕБ-ДАСТУРЛАШ

“Информатика ва ахборот технологиялари” йўналиши

**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**

**МУЎАММАД АЛ-ХОРАЗМИЙ НОМИДАГИ
ТОШКЕНТ АХБОРОТ ТЕХНОЛОГИЯЛАРИ УНИВЕРСИТЕТИ
ЎУЗУРИДАГИ ПЕДАГОГ КАДРЛАРНИ ҚАЙТА ТАЙЁРЛАШ ВА
УЛАРНИНГ МАЛАКАСИНИ ОШИРИШ ТАРМОҚ МАРКАЗИ**

«Информатика ва ахборот технологиялари» йўналиши

**“ВЕБ ДАСТУРЛАШ”
МОДУЛИ БЎЙИЧА**

Ў Қ У В – У С Л У Б И Й М А Ж М У А

Тошкент - 2019

Модулнинг ўқув-услубий мажмуаси Олий ва ўрта махсус, касб-хунар таълими ўқув-методик бирлашмалари фаолиятини Мувофиқлаштирувчи кенгашининг 2019 йил 18 октябрдаги 5 – сонли баённомаси билан маъқулланган ўқув дастури ва ўқув режасига мувофиқ ишлаб чиқилган.

Тузувчилар: Назирова И. - ТАТУ “Мультимедиа технологиялари” кафедраси мудири, т.ф.н., доц.

Такризчилар: С.Медетов – Нант политехника университети (Франция), Электрон ва рақамли технологиялар кафедраси профессори.
М.Якубов - ТАТУ “Ахборот технологиялари” кафедраси профессори, ф-м.ф.д.

Модулнинг ўқув-услубий мажмуаси Муҳаммад ал-Хоразмий номидаги Тошкент ахборот технологиялари университети Кенгашининг 2019 йил 29 августдаги 1 (694) – сонли баённомаси билан тавсия қилинган

МУНДАРИЖА

I. Ишчи дастур	4
II. Модулни ўқитишда фойдаланиладиган интерфаол методлари	9
III. Назарий материаллар	16
IV. Амалий машғулот материаллари.....	39
V. Кейслар банки.....	83
VI. Глоссарий	87
VII Адабиётлар рўйхати.....	92

І БЎЛІМ

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-сонли, 2017 йил 7 февралдаги “Ўзбекистон Республикасини янада ривожлантириш бўйича Ҳаракатлар стратегияси тўғрисида”ги ПФ-4947-сонли, 2019 йил 27 августдаги “Олий таълим муассасалари раҳбар ва педагог кадрларининг узлуксиз малакасини ошириш тизимини жорий этиш тўғрисида”ги ПФ-5789-сонли Фармонлари, шунингдек 2017 йил 20 апрелдаги “Олий таълим тизимини янада ривожлантириш чора-тадбирлари тўғрисида”ги ПҚ–2909-сонли қарорида белгиланган устувор вазифалар мазмунидан келиб чиққан ҳолда тузилган бўлиб, у олий таълим муассасалари педагог кадрларининг касб маҳорати ҳамда инновацион компетентлигини ривожлантириш, соҳага оид илғор хорижий тажрибалар, янги билим ва малакаларни ўзлаштириш, шунингдек амалиётга жорий этиш кўникмаларини такомиллаштиришни мақсад қилади.

Дастур доирасида берилаётган мавзулар таълим соҳаси бўйича педагог кадрларни қайта тайёрлаш ва малакасини ошириш мазмуни, сифати ва уларнинг тайёргарлигига қўйиладиган умумий малака талаблари ва ўқув режалари асосида шакллантирилган бўлиб, унинг мазмуни Ўзбекистоннинг миллий тикланишдан миллий юксалиш босқичида олий таълим вазифалари, таълим-тарбия жараёнларини ташкил этишнинг норматив-ҳуқуқий ҳужжатлари, 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, ўқув жараёнини ташкил этишнинг замонавий услублари бўйича сўнгги ютуқлар, педагогнинг креатив компетентлигини ривожлантириш, глобал Интернет тармоғи, мультимедиа тизимларидан фойдаланиш ва масофавий ўқитишнинг замонавий шакллари қўллаш бўйича тегишли билим, кўникма, малака ва компетенцияларни ривожлантиришга йўналтирилган.

Қайта тайёрлаш ва малака ошириш йўналишининг ўзига хос хусусиятлари ҳамда долзарб масалаларидан келиб чиққан ҳолда дастурда тингловчиларнинг махсус фанлар доирасидаги билим, кўникма, малака ҳамда компетенцияларига қўйиладиган талаблар такомиллаштирилиши мумкин.

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, лойихалаш 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 билимларга** эга бўлиши лозим.

Тингловчи:

- HTML-5, CSS, Java Script, JQuery, PHP ва замонавий CMS асосларини ва усулларини ҳамда улар асосида Web иловаларни ишлаб чиқиш, лойихалаш **кўникма ва малакаларини** эгаллаши зарур.

Тингловчи:

- масаланинг қўйилишига қараб янги Web иловаларни ишлаб чиқиш ҳамда ўқув ва ишлаб чиқариш жараёнига тадбиқ **этиш компетенцияларни** эгаллаши лозим.

Модулни ташкил этиш ва ўтказиш бўйича тавсиялар

“Веб дастурлаш” модули 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 “Мобил иловалар яратиш” 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

№	Модуль мавзулари	Аудитория уқув юкламаси			
		Жами	жумладан		
			Назарий	Амаий машғулот	Кўчма машғулоти
1.	Веб-дастурлашнинг асосий тушунчалари. Жараён (процедура) «сўров — жавоб». PHP, MySQL, JavaScript ва CSS имкониятлари. HTML 5. Apache веб сервери	2	2		
2.	PHP тузилмаси.	4		4	
3	MySQL асослари. MySQL буйруқлари. MySQL маълумотлар базасига PHP дан мурожаат қилиш. MySQL маълумотлар базасида ишлаш.	2	2		
4	JavaScript ва ундан фойдаланиш. JavaScript кодларни отлатка қилиш.	4		4	
5	CSS (Cascading Style Sheets) ни CSS3хисобига кенгайтириш.	2	2		
	Жами:	14	6	8	

НАЗАРИЙ МАШҒУЛОТЛАР МАЗМУНИ

1 – мавзу. Веб-дастурлашнинг асосий тушунчалари. PHP, MySQL, JavaScript ва CSS имкониятлари. HTML 5. Apache веб сервери (2 соат).

Веб дастурлашга кириш. PHP, MySQL, JavaScript ва CSS имкониятлари. HTML 5 ва Apache веб сервери.

2 – мавзу. MySQL асослари. MySQL буйруқлари. MySQL маълумотлар базасига PHP дан мурожаат қилиш. MySQL маълумотлар базасида ишлаш (4 соат).

MySQL га кириш. MySQL асослари.

3 - мавзу: CSS (Cascading Style Sheets) ни CSS3хисобига кенгайтириш (2 соат)

CSS нинг CSS3 хисобига кенгайтириши . Атрибутлар селектори. Background -clip хусусиятлари. Background –origin хусусиятлари. HTML5 да Холст ва Геолокация.

АМАЛИЙ МАШҒУЛОТЛАР МАЗМУНИ

1-амалий машғулоти. PHP тузилмаси (4 соат).

2-амалий машғулоти. JavaScript ва ундан фойдаланиш. JavaScript кодларни отлатка қилиш (4 соат).

ЎҚИТИШ ШАКЛЛАРИ

Мазкур модул бўйича қуйидаги ўқитиш шаклларидан фойдаланилади:

- маърузалар, амалий машғулотлар (маълумотлар ва технологияларни англаб олиш, ақлий қизиқишни ривожлантириш, назарий билимларни мустаҳкамлаш);
- давра суҳбатлари (кўрилаётган 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).

II БЎЛИМ

МОДУЛНИ ЎҚИТИШДА
ФОЙДАЛАНИЛАДИГАН
ИНТЕРФАОЛ
МЕТОДЛАРИ

II. МОДУЛНИ ЎҚИТИШДА ФОЙДАЛАНИЛАДИГАН ИНТЕРФАОЛ ТАЪЛИМ МЕТОДЛАРИ

«Блум кубиги» методи

Методнинг мақсади: Мазкур метод тингловчиларда янги ахборотлар тизимини қабул қилиш ва билимларни ўзлаштирилишини енгиллаштириш мақсадида қўлланилади, шунингдек, бу метод тингловчилар учун “Очиқ” саволлар тузиш ва уларга жавоб топиш машқи вазифасини белгилайди.

Методни амалга ошириш тартиби:

1. Ушбу методни қўллаш учун, оддий куб керак бўлади. Кубнинг ҳар бир томонида қўйидаги сўзлар ёзилади:
 - **Санаб беринг, таъриф беринг (оддий савол)**
 - **Нима учун (сабаб-оқибатни аниқлаштиривчи савол)**
 - **Тушинтириб беринг (муаммони ҳар томонлама қараш саволи)**
 - **Таклиф беринг (амалиёт билан боғлиқ савол)**
 - **Мисол келтиринг (ижодкорликни ривожлантиривчи савол)**
 - **Фикр беринг (таҳлил қилиш ва баҳолаш саволи)**
2. Ўқитувчи мавзунини белгилаб беради.
3. Ўқитувчи кубикни столга ташайди. Қайси сўз чиқса, унга тегишли саволни беради.

“KWLH” методи

Методнинг мақсади: Мазкур метод тингловчиларда янги ахборотлар тизимини қабул қилиш ва билимларни тизимлаштириш мақсадида қўлланилади, шунингдек, бу метод тингловчилар учун мавзу бўйича қўйидаги жадвалда берилган саволларга жавоб топиш машқи вазифасини белгилайди.

Изоҳ. KWLH:

Know – нималарни биламан?

Want – нимани билишни хоҳлайман?

How - қандай билиб олсам бўлади?

Learn - нимани ўрганиб олдим?.

“KWLH” методи	
1. Нималарни биламан: -	2. Нималарни билишни хоҳлайман, нималарни билишим керак: -
3. Қандай қилиб билиб ва топиб оламан: -	4. Нималарни билиб олдим: -

“W1H” методи

Методнинг мақсади: Мазкур метод тингловчиларда янги ахборотлар тизимини қабул қилиш ва билимларни тизимлаштириш мақсадида қўлланилади, шунингдек, бу метод тингловчилар учун мавзу бўйича кўйидаги жадвалда берилган олти саволларга жавоб топиш машқи вазифасини белгилайди.

What?	Нима? (таърифи, мазмуни, нима учун ишлатилади)	
Where?	Қаерда (жойлашган, қаердан олиш мумкин)?	
What kind?	Қандай? (параметрлари, турлари мавжуд)	
When?	Қачон? (ишлатилади)	
Why?	Нима учун? (ишлатилади)	
How?	Қандай қилиб? (яратилади, сақланади, тўлдирилади, таҳрирлаш мумкин)	

“SWOT-таҳлил” методи.

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

S – (strength)	• кучли томонлари
W – (weakness)	• заиф, кучсиз томонлари
O – (opportunity)	• имкониятлари
T – (threat)	• хавфлар

“БЕЕР” 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 айна 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 мантиқий фикрлашни муваффақиятли ривожлантиришга ҳамда ўқувчиларнинг мустақил ғоялари, фикрларини ёзма ва оғзаки шаклда тизимли баён этиш, ҳимоя қилишга имконият яратади. “Беер” 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 атрофлича 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.

Муаммоли савол					
1-усул		2-усул		3-усул	
афзаллиги	камчилиги	афзаллиги	камчилиги	афзаллиги	камчилиги
Хулоса:					

“Кейс-стади” методи

«Кейс-стади» - инглизча сўз бўлиб, («case» – аниқ вазият, ҳодиса, «stadi» – ўрганмоқ, таҳлил қилмоқ) аниқ вазиятларни ўрганиш, таҳлил қилиш асосида ўқитишни амалга оширишга қаратилган метод ҳисобланади. Мазкур метод дастлаб 1921 йил Гарвард университетида амалий вазиятлардан иқтисодий бошқарув фанларини ўрганишда фойдаланиш тартибида қўлланилган. Кейсда очик ахборотлардан ёки аниқ воқеа-ҳодисадан вазият сифатида таҳлил учун фойдаланиш мумкин.

“Кейс методи” ни амалга ошириш босқичлари

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

“Ассесмент” методи

Методнинг мақсади: мазкур метод таълим олувчиларнинг билим даражасини баҳолаш, назорат қилиш, ўзлаштириш кўрсаткичи ва амалий кўникмаларини текширишга йўналтирилган. Мазкур техника орқали таълим олувчиларнинг билиш фаолияти турли йўналишлар (тест, амалий кўникмалар, муаммоли вазиятлар машқи, қиёсий таҳлил, симптомларни аниқлаш) бўйича ташҳис қилинади ва баҳоланади.

Методни амалга ошириш тартиби:

“Ассесмент”лардан маъруза машғулотларида талабаларнинг ёки қатнашчиларнинг мавжуд билим даражасини ўрганишда, янги маълумотларни баён қилишда, семинар, амалий машғулотларда эса мавзу ёки маълумотларни ўзлаштириш даражасини баҳолаш, шунингдек, ўз-ўзини баҳолаш мақсадида индивидуал шаклда фойдаланиш тавсия этилади. Шунингдек, ўқитувчининг ижодий ёндашуви ҳамда ўқув мақсадларидан келиб чиқиб, ассесментга қўшимча топшириқларни киритиш мумкин.

Ҳар бир катакдаги тўғри жавоб 5 балл ёки 1-5 балгача баҳоланиши мумкин.



Тест

Муаммоли вазият

**Тушунча таҳлили
(симптом)**

Амалий вазифа

“Инсерт” методи

Методни амалга ошириш тартиби:

- ўқитувчи машғулотга қадар мавзунинг асосий тушунчалари мазмуни ёритилган матнни тарқатма ёки тақдимот кўринишида тайёрлайди;
- янги мавзу моҳиятини ёритувчи матн таълим олувчиларга тарқатилади ёки тақдимот кўринишида намоёниш этилади;
- таълим олувчилар индивидуал тарзда матн билан танишиб чиқиб, ўз шахсий қарашларини махсус белгилар орқали ифодалайдилар. Матн билан ишлашда талабалар ёки қатнашчиларга қуйидаги махсус белгилардан фойдаланиш тавсия этилади:

Белгилар	Матн
“V” – таниш маълумот.	
“?” – мазкур маълумотни тушунмадим, изоҳ керак.	
“+” бу маълумот мен учун янгилик.	
“– ” бу фикр ёки мазкур маълумотга қаршиман?	

Белгиланган вақт яқунлангач, таълим олувчилар учун нотаниш ва тушунарсиз бўлган маълумотлар ўқитувчи томонидан таҳлил қилиниб, изоҳланади, уларнинг моҳияти тўлиқ ёритилади. Саволларга жавоб берилади ва машғулот яқунланади.

Ш БЎЛИМ

НАЗАРИЙ
МАТЕРИАЛЛАР

III. НАЗАРИЙ МАТЕРИАЛЛАР

1 – мавзу. Веб-дастурлашнинг асосий тушунчалари. PHP, MySQL, JavaScript ва CSS имкониятлари. HTML 5. Apache веб сервери (2 соат).

Режа:

- 1.1. Веб дастурлашга кириш.
- 1.2. PHP, MySQL, JavaScript ва CSS имкониятлари.
- 1.3. HTML 5 ва 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 узоклашган 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 буни гипер мурожаатларни узатиш протоколлари ёки (Hyper Text Transfer Protocol (HTTP)) деб аташди)

У махсус разметкали тил ишлаб чиқди бу тил гипер текстли разметкали тил 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 йуққа 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 сахифалар 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 ифодамай олмас эди.

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

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.

Бунда амаллар кетма кетлиги қуйидагича бўлади:

1. Браузер адрес сатрига қуйидагилар киритилади `http://server.com`.
2. Сизнинг браузер IP-адресни излайди, `server.com` номли доменга мос бўлган.
3. `server.com` нинг бош саҳифасига браузер сўров юборади
4. Сўров интернетдан орқали `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).

1.2. 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

PHP тарихи. Кўпгина бошқа дастурлаш тилларидан фарқли равишда, PHP қандайдир ташкилот ёки кучли дастурчи томонидан яратилган эмас. Уни оддий фойдаланувчи Расмус Лерддорф 1994 йили ўзининг бош саҳифасини интерактив услубда кўрсатиш учун яратган. Унга Personal Home Page (PHP – шахсий бош саҳифа) деб ном берган.

1995 йили Расмус PHPни ўзининг HTML формалари билан ишлайдиган

бошқа дастур билан умумлаштириб PHP/FI Version 2 ("Form Interpretator") ҳосил қилди. 1997 йилга бориб PHP дан фойдаланувчи сайтлар 50 мингдан ошди. Шундан сўнг веб технология усталари PHP ғояси асосида мукамал тил яратишга Зива Сураски ва Энди Гутманс асосчилигида киришилди. PHPни самарали деб ҳисобланмагани учун деярли нолдан бошлаб, мавжуд C ва Perl тилларидан ибрат олиб PHP3 талқинини яратилди. 1999 йилга келиб PHP асосида қурилган сайтлар миллиондан ошиб кетди. 2000 йилда эса Zend Technologies ширкати янги кўпгина функцияларни қўшган ҳолда PHP4 шарҳловчисини яратди.

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

Сервер томонидан бажариладиган иловалар Бундай иловалардан фойдаланиш учун PHP-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 PHP тили ёрдамида бирор-бир компьютерда браузер ва web-серверлардан мустақил равишда ўзи бажариладиган скриптларни ҳам яратиш мумкин.

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

Қачонки саҳифага PHP 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 мижозга оддий HTML кўринишида жўнатилади

Шунинг учун Here's the latest news матни браузерда кўринади

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

Натижада бу иккита қисмдан иборат қуйидаги маълумотларни чиқаради:
Today is Wednesday. Here's the latest news.

PHP етарлича осон ва келишувчан тил ҳисобланади баъзи дастурчилар PHP конструкцияларга ҳалақит 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 ўргимак 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 ҳақида гапиришга ҳожат қолмайди.

Худди шундай ҳолатлардаги 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 ўрнатилган 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 ҳисобланади, ичида бир қатор жадваллардан иборат, бу жадвалларда маълумотлар сақланади.

Фараз қилайлик, сиз users (фойдаланувчилар) жадвали билан ишляпсиз, унда графлар ҳосил 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 ҳосил 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 келинган.

Бунда қидириш масалалари ҳам жуда осон хал қилинган. Фараз қилайлик электрон адрес мавжуд бўлса шу адрес эгасининг исми 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 ўзиш 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.

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 хужжатнинг 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 элемети хужжатнинг объект модели DOM (Document Object Model) қўшилгандан кейин янада машхур бўлиб кетди. Объект модели хужжатга янги абҷас 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 гуруҳига 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.

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 ҳар 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 элементида 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

1.3. HTML 5 ва Веб сервер Apache

Веб – стандартларга киритилган қўшимча ўзгартиришлар ҳар қандай шов-шувли бўлмасин бу дастурчилар учун бу етарли бўлмай кўриनावеради.

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 лойиҳалар 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 `` 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 ҳажмдаги маълумотлар керак бўлади, бу иккита MathML ва SVG элемент махсус тушунча учун алоҳида китоблар ёзилган бошқа тўхталиб ўтirmаймиз

PHP, MySQL, JavaScript, CSS ва HTML5 ларга қўшимча қилиб динамик веб технологияларга 6 – технологияни веб-серверларни келтиришимиз мумкин. Биз қуйида Apache веб-сервер ҳақида 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 ҳар бир элемент учун, сервердан бажариши керак бўлган хизматни сўрайди.

Аmmo 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 булаётган 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. Оқиқ 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. Duvanov A.A. WEB. Construction of HTML. - St. Petersburg: BHV, 2003.
3. Barrett D. JavaScript. Web-professionals. - Kiev: BHV - Kyiv, 2001.
4. Brandenbau D. JavaScript: a collection of recipes. - St. Petersburg.: Peter, 2000.

2 – мавзу. MySQL асослари. MySQL буйруқлари. MySQL маълумотлар базасига PHP дан мурожаат қилиш. MySQL маълумотлар базасида ишлаш (2 соат).

Режа:

2.1. MySQL га кириш.

2.2. MySQL асослари.

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

2.1. 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 масштаблашаган тизимдир, яъни сайт хажмига қараб ошиб боради, охири 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.

Author (автор)	Title (название)	Type (категория)	Year (год)
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.

MySQL га командалар сатридан мурожаат

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

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.

robrix# 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.

Команда сатри интерфейсидан фойдаланиш

Ихтиёрий опрецион 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 нуқтали 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.

MySQL командалари

Қуйидаги жадвалда энг кўп ишлатиладиган MySQL командалари келтирилган

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

- 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
\c	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

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 холда сиз «Создание таблицы» пунктига ўтишингиз мумкин бўлади. Агар бундай бўлмаса сиз навбатдаги командани киритишни бошлайсиз, янги маълумотлар базасини қуриш учун куйидаги командани киритасиз:

publications:

CREATE DATABASE publications;

Команда мувоффақиятли бажарилса, куйидагича 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 куйидаги 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 куйидагича командалардан фойдаланилади: GRANT бу команда куйидагича ишлайди: GRANT ПРАВА ON база_данных.объект TO 'имя_пользователя@имя_хоста' IDENTIFIED BY 'пароль';

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

Энди биз publications даги янги маълумотлар базасига фойдаланувчига рухсат берувчи командани келтириб ўтамыз 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 хосил қиламиз ва ўша жадвал мавжудлигини текшираамиз ва ундан кейин ўчираамиз ва уни экранда кўраамиз:

```
CREATE TABLE disposable(trash INT);  
DESCRIBE disposable;  
DROP TABLE disposable;  
SHOW tables;
```

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 сатридан 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 ҳосил 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 ҳосил 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 ҳаммаси 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 тушунасиз?
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. Duvanov A.A. WEB. Construction of HTML. - St. Petersburg: BHV, 2003.
3. Barrett D. JavaScript. Web-professionals. - Kiev: BHV - Kyiv, 2001.
4. Brandenbau D. JavaScript: a collection of recipes. - St. Petersburg.: Peter, 2000.

3- Мавзу: CSS (Cascading Style Sheets) ни CSS3хисобига кенгайтириш (2 соат)

Режа:

- 3.1. CSS нинг CSS3 хисобига кенгайтириши .
- 3.2. Атрибутлар селектори.
- 3.4. Background -clip хусусиятлари. Background –origin хусусиятлари.
- 3.5. HTML5 да Холст ва Геолокация.

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

3.1. CSS нинг CSS3 хисобига кенгайтириши

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 функционалигини 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.

3.2. Атрибутлар Селектори

Селектор бу форматлаш безаш параметрларини ўз ичига олувчи стилдир. Селекторларга теглар, синфлар, идентификаторлар киради. Ўнлаб селекторлар мавжуд бўлиб, CSS3 ишлаб чиққан 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 :

Қуйидаги элемент эса мос келмайди:

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:

 Қуйидаги тег мос келмайди:

Box –sizing хусусияти

W3C модел блокада 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 хошишни амалга ошириш учун CSS3 сизга шундай блокни таклиф қилиши мумкинки, сиз хоҳлаган ўлчамдаги блок билан ишлаш

имконини беради бу `box-sizing` деб аталади.

Масалан объектнинг умумий буйи ва энини чегара ва чеклашларни хисобга 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;
```

3.1. `background-clip` хусусиятлари. `background-origin` хусусияти

`background-clip` хусусияти фон ишлатиш ёки йўқлигини аниқлайди, фон чегараланган сохада ичида бўлсинми ёки ташқарисидами, шуни аниқлаш учун ишлатилади. Масалан қуйидаги эълон фон элементнинг барча қисмларида чегараланган сохагача 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;
```

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 соха 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 хар бир элементнинг ички мазмунини аниқлаш учун ишлатилади.

`background-origin` хусуияти билан фон тасвир қаерда туришини ва қаердан бошлашини аниқлаштириш (кўрсатиш) мумкин бўлади.

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

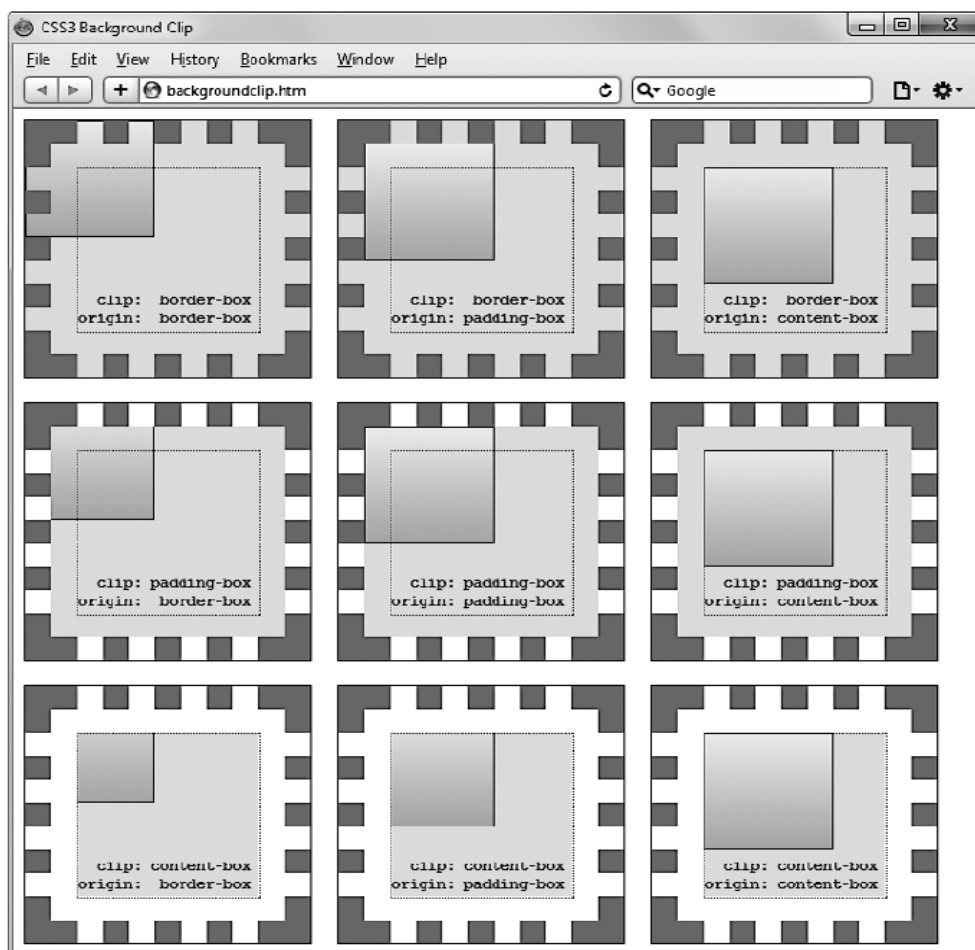


Рис. 20.1. Разные способы сочетания свойств фона CSS3

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;
```

Расмда 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` ишлатилляпти.

background-size хусусияти

Худди шундай `` тегидаги ўлчовларни аниқлаш усули билан бир хил фон тасвири улчамларини аниқлаш. Кейинги барча браузерлар фонни худди шундай ифодаляпти.

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 булар га нисбатан эни ва бўй 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.

3.3. HTML5 да Холст ва Геолокация

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.

Аmmo 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 интелект татбиқлари

Siri, Apple, Cortana, Microsoft , Watson , IBM компанияларидаги каби технолгияларни, 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

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

Canvas стандартлаштиригандан 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

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

Мисол. 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>
```

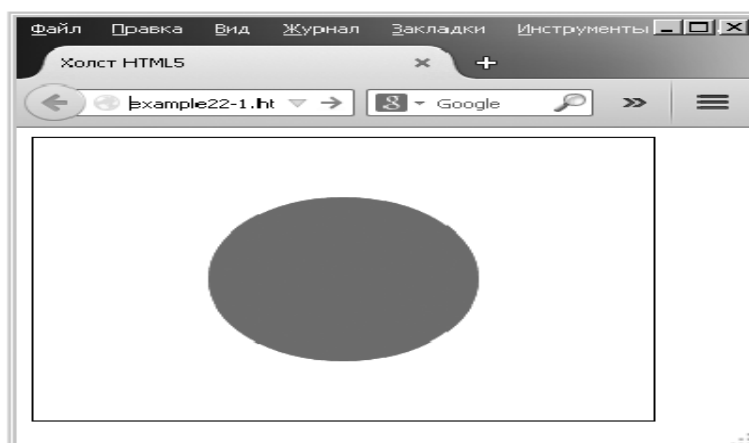


Рис. 22.1. Рисование японского флага на холсте HTML5

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

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 ҳар доим вазиятни назорат қилади, ва хоҳласа бу хизматдан фойдаланади ёки бутунлай бу хизматни блоклай 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 ҳақозалар.

Мисолда Google харитаси келтирилган фойдаланувчи жойлашган ўрнини кўрсатяпти, бунда фойдаланувчи браузерини геолокацияни рухсат берилган ва фойдаланувчи ўзи турган жойни аниқлаган. Акс ҳолда хатолик хақида хабар берилади.

Мисол. 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. Рўйхатларни қандай ташкил этиш мумкин?
6. CSS селекторлар вазифалари нималардан иборат?
7. CSS синфлар вазифасини мада?

Фойдаланилган адабиётлар:

1. VG Olifer, N. A.Olifer. Computer networks. Principles, technologies, protocols. St. Petersburg: Publishing House. "Peter," in 2000.
2. uvanov A.A. WEB. Construction of HTML. - St. Petersburg: BHV, 2003.
3. Barrett D. JavaScript. Web-professionals. - Kiev: BHV - Kyiv, 2001.
4. Brandenbau D. JavaScript: a collection of recipes. - St. Petersburg.: Peter, 2000.



IV. АМАЛИЙ МАШҒУЛОТ МАТЕРИАЛЛАРИ

1 – амалий машғулот. PHP тузилмаси (4 соат)

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 текширамыз. Ушбу мисолни кўрамыз:

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 компьютерида 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 созила 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 конфигурация файлга `short_open_tag` буйруқни ўрнатиш керак, ёки *PHP* дастурлаш тилида `enable-short-tags` параметр билан компиляция қилиш керак. Агарда `php.ini-dist` буйруқка 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 *RHP* 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 *RHP* дастурлаш тили синтаксисига нисбатан нималарни билиш керак. Бу HTML-код ичига ўрнатилган ва *RHP* 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.

RHP 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. *RHP* 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 ҳажмдаги 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 комментарий ёзиб қўйилади. Барча дастурлаш тилларида дастур ичига комментарий қўшиш 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* олдига доллар белгиси (“\$”) қўйиб эълон қилинади, масалан, \$my_var.

Ўзгарувчилар номлари регистрларни фарқлайди, яъни \$my_var ҳамда бош харфли \$My_var ўзгарувчилари турли хил ўзгарувчилардир.

PHP дастурлаш тилида ўзгарувчилар номи қолган дастурлаш тиллари коидалари 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 ўзлаштирадик, 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 ўзлаштирадик
$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 скриптни 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 олдига 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 скриптни 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 сегирлиги катта. Ҳар бир ўзгармасларда уни ўзгартириш мумкин, яъни `регистрга_сезгирлиги_кичик` аргументни қиймати сифатида `True` биймати 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 *PHP* 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 қуйида тўлиқроқ муҳокама қиламиз:

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

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

Ўзлаштириш амаллари.			
Белгиланиши	Номланиши	Изох	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 сатрлар қўшилади, сўнгра ҳосил бўлган сатр ўзгарувчига ўзлашади).	$\$b = "Ҳаммага";$ $\$b .= "салом";$ ($\$b = \$b . "салом"$ ифодага эквивалент;) Натижаси: $\$b = "Ҳаммага салом"$

Матикий амаллар.			
Белгиланиши	Номланиши	Изох	Мисол
and	ВА	$\$a$ ва $\$b$ рост (True)	$\$a$ and $\$b$
&&	ВА		$\$a$ && $\$b$
or	ЁКИ	$\$a$ ёки $\$b$ ўзгарувчилардан ҳеч бўлмаганда	$\$a$ or $\$b$

		биттаси рост бўлса (иккаласи ҳам рост бўлишиш мумкин).	\$b \$a \$b
	ЁКИ		\$a xor \$b
xor	Инверсия ЁКИ	Ўзгарувчилардан биттаси рост бўлса. Агарда иккаласи ҳам рост бўлса инверсияланади.	! \$a
!	Инверсия (NOT)	Агарда \$a=True, у ҳолда !\$a=False ва акс ҳолда тескариси бўлади.	

Таққослаш амаллари.

Белгиланиши	Номланиши	Изох	Мисол
==	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

Инкремент ва декремент амаллари.

Белгиланиши	Номланиши	Изох	Мисол
++\$a	Пре-инкремент	\$a қиймати бирга оширилади ва \$a қиймати қайтарилади	<? \$a=4;
\$a++	Пост-инкремент	\$a қиймати қайтарилади ва сўнгра \$a қиймати бирга оширилади	echo "4 бўлиши шарт:" . \$a++;
--\$a	Пре-декремент	\$a қиймати бирга камайтиради ва \$a қиймати қайтарилади	echo "6 бўлиши шарт:" . --\$a;
\$a--	Пост-декремент	\$a қиймати қайтарилади ва сўнгра \$a қиймати бирга камайтиради	?>

Маълумотлар типлари.

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

Тўрттаси скаляр *типлар*:

- *boolean* (мантиқий);
- *integer* (бутун);
- *float* (нуқтаси силжсийдиган);
- *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 контекстан, 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 типи(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 конструкцияларда* (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 конструкцияларда* 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 юкоридагидек $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 *қўзгалувчан вергулли сонга* ўзгартиради. Худди шундай *бутун сон* чегарасидан ташқари чиқиб кетадиган бирор амал бажарсангиз ҳам бу сонни *қўзгалувчан вергулли сонга* ўзгартирилади.

PHP дастурлаш тилида бутун сонларни бўлиш амали мавжуд эмас. 1/2 ифода қиймати *қўзгалувчан 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 олдиға кавс 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 (қўзгалувчан вергулли сон) типи.

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

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

Қўзгалувчан вергулли сонни ўлчами ҳам платформага боғлиқ, лекин қоидага қўра 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 типидagi 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
кўйилади.
echo ' Белгини \' чиқариш учун ундан олдин' '\ белгини кўйиш керак';
// Экранга чиқаради: Сиз шуни ўчирмоқчимисиз C:\*.*?
echo ' Сиз шуни ўчирмоқчимисиз C:\*.*?';
// Экранга чиқаради: Буни кўйманг: \n янги қаторга
echo ' Буни кўйманг: \n янги қаторга ';
// Экранга чиқаради: ўзгарувчи $expand ҳам $either кўйилмайди
echo 'ўзгарувчи $expand ҳам $either кўйилмайди';
?>

```

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 атаймиз. *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 кўшиб 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 қўштирноққа 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 ўзлаштирадиз, 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.

```
<?php
$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 қидирилади. 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 (калити 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 кўйилган 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 типидан* фойдаланиш кулай. Мактуб матни ҳар доим ўзгармас, шунинг учун уни *ўзгармас* деб бериш мақсадга мувофиқдир. Жуда узун ва кўпол сатрларни ёзмаслик учун сатрлар *конкатенация*(кўшиш) 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");
$sevents = array(

```



```

"f" => "очик эшиклар куни",
"o" => "кўргазманинг очилиши",
"p" => "битирувчилар бали");

// таклифнома матнини тузамиз.
$str = "Хурматли, $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). Скриптни қайта ишлаш жараёнида ифода мантикий 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 мантикий типга ўзлаштириш қондаси:

1. `FALSE` қиймати қуйидаги қийматларга тўғри келади:
 - мантикий `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.

```

<?
$names = array("Карим", "Салим", "Содик");
if ($names[0]=="Карим") {
    echo "Салом, Азамат!";
    $num = 1;
    $account = 2000;
}

```

```

if ($num) echo "Карим рўйхатда биринчи!";
$бах = 30;
if ($account > 100*$бах+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?!";
$бах = 30;
if ($account > 100*$бах+3)
    echo " Бу сатр экранга чиқмайди, чунки шарт бажарилмайди ";
else echo "Шундай бўлса ҳам экранга чиқди!";
?>

```

elseif 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.

```
<?
$names = array("Карим", "Салим", "Содик");
if ($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] == "Содик"){
    // массивда биринчи элемент ҳам Карим, ҳам Салим эмас Содик бўлса
    echo "Салом, Содик!";
}else {
    // массивда биринчи элемент на Карим, на Салим, на Содик 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
$names = array("Карим", "Салим", "Содик");
if ($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 қуйидагича 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 конструкцияси учун худди *if* оператори каби альтернатив синтаксиси мавжуд. Бу ерда *switch* операторидаги очиладиган фигурали қавс икки нуқтага ўзгартирилади, ёпиладигани эса мос равишда *endswitch*; калит сўзига ўзгартирилади.

ЦИКЛЛАР.

PHP дастурлаш тилида шартга боғлиқ равишда қайтариладиган амаллардан иборат бир нечта конструкциялар мавжуд. Бу *while*, *do..while*, *foreach* ҳамда *for* цикллардир. 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) бўлгунича *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 *RHP* 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 қуйидагича 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 қуйидагича 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 блокдан кейинги инструкцияларни ўтказиб юбориш ва янги доира билан амалга оширишни давом эттириш имконини беради. *continue* ни унинг таркибида бошқарилувчи конструкциялар ишини яқунлаш 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:

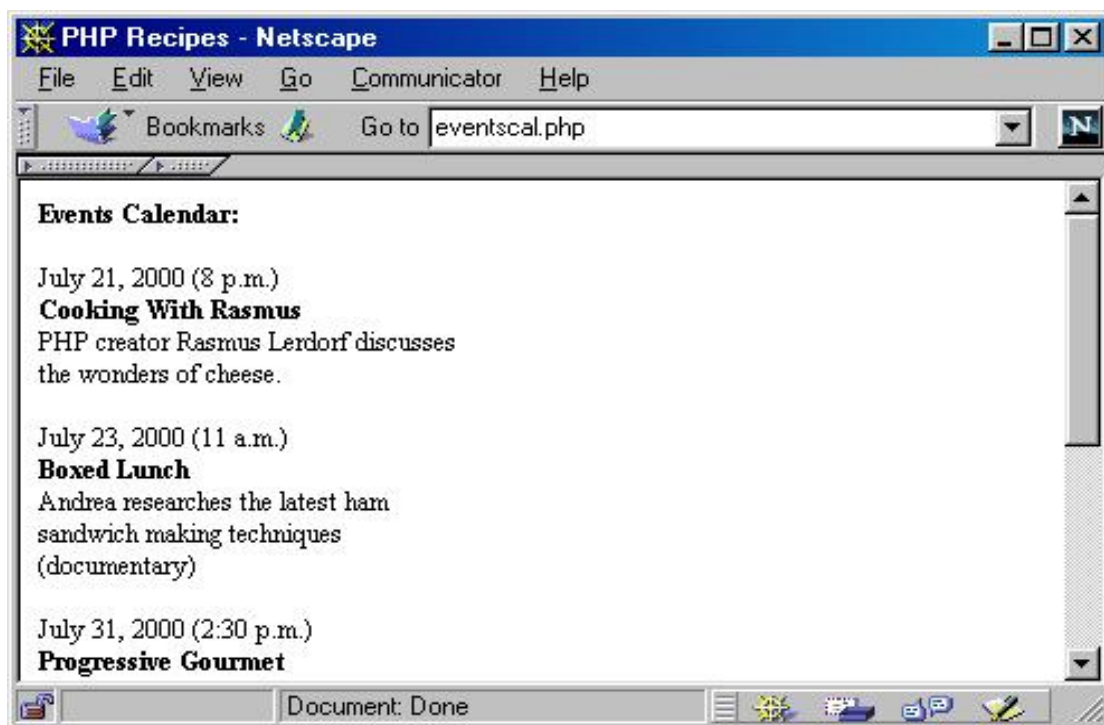
```
<?php
$i=1;
while ($i<4) {
    $n = rand(1,10);
    // исталган сонни умумлаштирамиз 1 дан 10 гача
    if ($n!=5) {
        echo "$i:$n <br>";
    // интерация рақамини чиқарамиз ва умумлаштирилган сон
    $i++;
    }
}
?>
```

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.

Амалий вазифалар:

Проект: календарни яратиш

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. Duvanov A.A. WEB. Construction of HTML. - St. Petersburg: BHV, 2003.
3. Barrett D. JavaScript. Web-professionals. - Kiev: BHV - Kyiv, 2001.
4. Brandenbau D. JavaScript: a collection of recipes. - St. Petersburg.: Peter, 2000.

2 - амалий машғулот. JavaScript ва ундан фойдаланиш. JavaScript кодларни отлатка қилиш (4 соат)

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 харакати эса эълон қилинган ўзгарувчилар ичидаги функциялар билан чегараланган. Basic дастурлаш тили сингари JavaScript сценарийсини яратаётган вақтда аввалдан эълон қилинмаган ўзгарувчиларни ишлатиш мумкин.

Ўзгарувчилар эълони

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

```
var MyHelloMsg;
```

Ўзгарувчи типи ўзлаштириладика 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, ыуидаги оддий ыоидаларни ушлаб ыщйиш керак:

- Ўзгарувчи номи харфлардан ёки "_", "\$" белгилардан бошланиш керак ва фақат харфлардан, 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 ўйидаги ўзгарувчи келтирилган ва унда матнли қатор ёзилган:

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

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

```
MyHelloMsg = 4;
```

Бу оператор бажарилгандан сўнг ўзгарувчи типи ўзгаради, шунингдек интерпретация жараёнида браузер ҳеч қандай огохлантувчи 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 руҳсат 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 холларда "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 холларда 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 ўзгарувчисидида матнли қатор сақланмоқда (бўш бўлса ҳам), иккинчисидида 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 хали 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. Мантикий 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, қўшиш, қўпайтириш ҳамда бўлинмани қолдиғини ҳисоблаш учун ишлатилади (кўрилади):

- Айириш

+ Қўшиш

* Қўпайтириш

/ 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 операцисини бажариш учун 3 та оператор кшрилган:

>> 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 мантикий 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

Мантикий 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 осонлаштиради бошланғич

текстни содалаштиради.

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.

JavaScript сценарийли тили объектга-мўлжалланган тилдир. JavaScript объеклари 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 ошириш 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 охирги рақамлилигини аниқлаш
isNaN	Аргументнинг рақам ёки рақам эмаслигини аниқлаш
parseFloat	Қаторни кўчиб юривчи 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 типи 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 охириги 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 булса 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 ихсобланиб, агарда n=10 бўлса, parseInt(s) функция сатрдаги 10 лик санок системасидаги сонларни текширади. n=8 бўлса, 8 лик соноқ тизимидаги сонлар мавжудлигини (бунда n 0 дан 7 гача бўлган сонлар қийматига эга бўлиши мумкин). n=16 бўлса, 16 лик соноқ 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 классси методлари

abs(x) (x-сон еки ифода) – абсолют қийматни ҳисоблаш;

acos(x) (x бу ерда [-1.0;1.0] радиан интервалдаги сон еки ифода) – арккосинусни ҳисоблаш. Қайтариладиган қиймат 0 дан p радиан оралиғида бўлади.

asin(x) (x бу ерда [-1.0;1.0] радиан интервалдаги сон еки ифода) – арксинусни ҳисоблаш. Қайтариладиган қиймат $-p/2$ дан $p/2$ радиан оралиғида бўлади.

atan (x) (x – сон еки ифода) – арктангенсни радианларда ҳисоблаш. Қайтариладиган қиймат $-p/2$ дан $p/2$ радиан оралиғида бўлади.

atan2(x,y)(x,y — тўғри бурчакли кордината системаси кордината нуқталари) – кутб кординатасида (x,y) нуқталар бурчагини ҳисоблайди. Қиймати 0 дан 2p радиан оралиғида бўлади.

ceil(x) (x — сон ёки сонли ифода) – сонни бутун сонга йўналтирилган ҳолда яхлитлаш. Манфий сонлар 0 сони йўналишига караб яхлитланади.

cos(x) (x – радиандаги бурчак) – косинусни ҳисоблаш, қайтариладиган қиймат -1.0 дан 1.0 радиан оралиғида бўлади.

sin(x) (x – радиандаги бурчак) – косинусни ҳисоблаш, қайтариладиган қиймат -1.0 дан 1.0 радиан оралиғида бўлади.

Exp(x) (x — сон ёки сонли ифода) — e экспонентсини ҳисоблаш.

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

Log(x) (x — мусбат сон ёки ифода) – натурал лагориғмни ҳисоблаш.

max(a,b) (a,b — сон ёки ифода) – икки қийматдан каттасини қайтаради.

min(a,b) (a,b — сон ёки ифода) – икки қийматдан кичигини қайтаради.

`pow(x,y)` — x ни ҳисоблаш (биринчи аргументни даражага кўтариш).
`random` — 0 дан 1 гача интервалдаги тасодифий сонларни ҳисоблаш.
`round` — сонни бутун қисмига қараб яхлитлаш (масалан, `round(15.5)` натижаси 16 ни беради, `round(-15.5)` даёт -15).

`Math.round(x)` (x — сон ёки ифода)

`Math.sin(x)` (x — радианда берилган бурчак)

`Math.sqrt(x)` (x — 0 га тенг ёки катта бўлган сон ёки ифода)

`tan` — тангенсни ҳисоблаш.

`Math.tan(x)` (x — радианда берилган бурчак)

Класс Date

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

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

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

`Date` объектини яратишнинг юқоридаги синтактикасида кўрсатилгани бўйича, сана ва вақт худудий вақт бўйича берилади. 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 худудий 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 кийматини беради;

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

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

setDate() - Date объекти вақт кўрсаткичини ўрнатади;

date.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

Қуйида 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 яратди. 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());
```

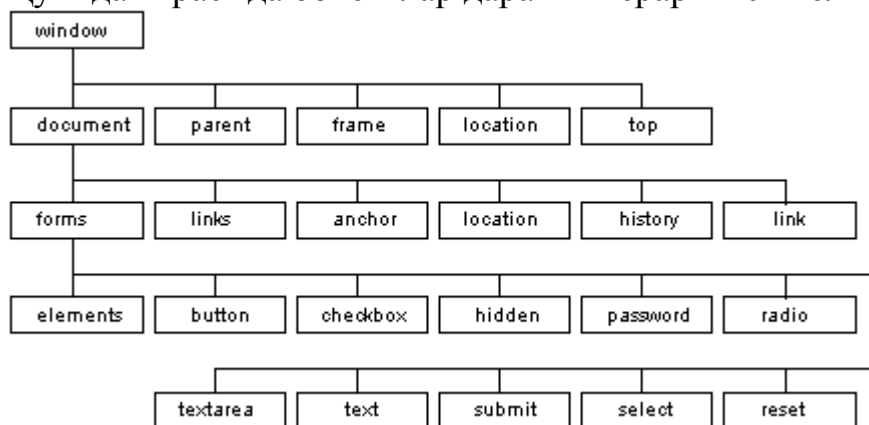
Браузер объектлари

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 усулларидан фойдаланиб бир класс асосида бошқа класс яратиб бўлмайди.

Браузер объектлари иерархияси

Қуйидаги 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 ойнани 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 муружаатини 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. Темура 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.

Амалий вазифалар:

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) $14 \& 56 | 11 =$

3. JavaScript нинг танланган объекти асосида мисол келтиринг.

- 1. Объект Global (ошқормас ҳолда бериш)
- 2. Объект Math
- 3. Объект Window
- 4. Коллекция фреймов (window.frames)
- 5. Объект Document (window.document)
- 6. Объект Location (window.location)
- 7. Объект Link (document.link)
- 8. Объект History
- 9. Объект MimeType
- 10. Объект Navigator

Фойдаланилган адабиётлар:

- 1. VG Olifer, N. A.Olifer. Computer networks. Principles, technologies, protocols. St. Petersburg: Publishing House. "Peter," in 2000.
- 2. Duvanov A.A. WEB. Construction of HTML. - St. Petersburg: BHV, 2003.
- 3. Barrett D. JavaScript. Web-professionals. - Kiev: BHV - Kyiv, 2001.
- 4. Brandenbau D. JavaScript: a collection of recipes. - St. Petersburg.: Peter, 2000.

VII БҮЛҮМ

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.

2-Кейс. Дастурчи Веб иловага You Tube сайтидаги бирор видео иловани жойлаштирмоқчи бўлди. Яъни шу илова ишга тушганда илова тугридан тугри You Tube сайтидан видео олиб кўрсатиши керак эди аммо илова кутилган натижани бермади.

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.

4-Кейс. Дастурчи веб саҳифани фонини ўзгартирмоқчи бўлди аммо буни амалга ошира олмади. Илова кутилган натижани бермади.

5-Кейс. Дастурчи веб саҳифадаги ёзувни бир хил интервалда жойлаштирмоқчи бўлди аммо буни амалга ошира олмади. Илова кутилган натижани бермади.

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 кутилган натижани бермади.

7-Кейс. Дастурчи веб саҳифага квадрат тенглама формуласини жойлаштирмоқчи бўлди аммо буни амалга ошира олмади. Илова кутилган натижани бермади.

8-Кейс. Дастурчи веб саҳифага CSS ёрдамида фото суратни эллипс ичига жойлаган ҳолда жойлаштирмоқчи бўлди, аммо буни амалга ошира олмади. Илова кутилган натижани бермади.

9-Кейс. Дастурчи веб саҳифага CSS ёрдамида фото суратни айлана ичига жойлаган ҳолда жойлаштирмоқчи бўлди, аммо буни амалга ошира олмади. Илова кутилган натижани бермади.

10-Кейс. Дастурчи веб саҳифага CSS ёрдамида маълум узунликдаги сатрларни ҳар хил рангларда ифодалашни ва мос сатрлар устида шу сатр рангини инглизча ёзмоқчи бўлди, аммо буни амалга ошира олмади. Илова кутилган натижани бермади.

11-Кейс. Дастурчи веб саҳифага CSS ёрдамида маълум тугмалар(button) ҳар хил рангларда ифодалашни ва мос тугмалар устида шу тугмалар рангини инглизча ёзмоқчи бўлди, аммо буни амалга ошира олмади. Илова кутилган натижани бермади.

12-Кейс. Дастурчи веб саҳифада бир саҳифадан иккинчи саҳифага ўтказувчи код ёзди, аммо буни амалга ошира олмади. Илова кутилган натижани бермади.

13-Кейс. Дастурчи веб саҳифага бир нечта фото суратлар ва уларнинг остига изох ёзмоқчи бўлди, аммо буни амалга ошира олмади. Илова кутилган натижани бермади.

14 – Кейс. Дастурчи HTML да веб саҳифа ёзиб CSS да унга дизайн бермоқчи бўлди , аммо буни амалга ошира олмади. Илова кутилган натижани бермади.

15 – Кейс. Дастурчи PHP да браузерда ўзгарувчига қиймат бериб натижани саҳифада кўрмоқчи ва шунга мос код ёзмоқчи бўлди, аммо буни амалга ошира олмади. Илова кутилган натижани бермади.

16 – Кейс. Дастурчи PHP да include функциясидан фойдаланиб дастур ёзмоқчи бўлди. Яъни ўзгарувчи қийматларини бошқа файлдан ўқиб олмоқчи ва унга мос код ёзмоқчи бўлди, аммо буни амалга ошира олмади. Илова кутилган натижани бермади.

17–Кейс. Дастурчи PHP 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.

18–Кейс. Дастурчи PHP 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.

19 – Кейс. Дастурчи PHP да формада иккита соннинг экубини ва экукини ҳисобловчи функция ёзмоқчи бўлди, аммо буни амалга ошира олмади. Илова кутилган натижани бермади.

20–Кейс. Дастурчи PHP дастурда бирор ўзгарувчи ва қийматини мавжуд ёки йўқлигини аниқловчи(isset ёки unset) код ёзмоқчи бўлди, аммо буни амалга ошира олмади. Илова кутилган натижани бермади.

Кейси бажариш босқичлари ва топшириқлар:

- Кейсдаги асосий тушунчаларга таърифларни келтиринг.
- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг.
- Веб иловани ишга тушириш учун бажариладиган ишлар кетма-кетлигини белгиланг.

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

ГЛОССАРИЙ

VI. 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 (компьютерда) ўрнатилган бўлади. Сервер мижознинг	

	буйруқлари бўйича маълум бир амалларни миждозга хизматлар тақдим этган ҳолда бажаради.	
Cracer	қонунсиз йўллар билан ҳимояланган ресурсларга масалан, конфиденсиал маълумотларга киришга рухсат олиб маълумотларни қидириш билан шуғулланувчи ҳисоблаш тизимининг (одатда компьютер 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	An XML file that each application must define, to describe the application's package name, version, components (activities,

	боғланиш)	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 framework ҳисобланади	It's a framework that you can buy from other developers and on marketplaces.
Surface	Canvas объектини бошқариш учун объект тип	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 компонентлар (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	иловани 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.

VII БЎЛИМ

АДАБИЁТЛАР
РЎЙХАТИ

VII. АДАБИЁТЛАР РЎЙХАТИ

I. Ўзбекистон Республикаси Президентининг асарлари

1. Каримов И.А. Ўзбекистон мустақилликка эришиш оstonасида. - Т.:“Ўзбекистон”, 2011.
2. Мирзиёев Ш.М. Буюк келажакимизни мард ва олижаноб халқимиз билан бирга курашимиз. – Т.: “Ўзбекистон”. 2017. – 488 б.
3. Мирзиёев Ш.М. Миллий тараққиёт йўлимизни қатъият билан давом эттириб, янги босқичга кўтарамиз – Т.: “Ўзбекистон”. 2017. – 592 б.

II. Норматив-ҳуқуқий ҳужжатлар

4. Ўзбекистон Республикасининг Конституцияси. – Т.: Ўзбекистон, 2019.
5. Ўзбекистон Республикасининг “Таълим тўғрисида”ги Қонуни.
6. Ўзбекистон Республикасининг “Коррупцияга қарши курашиш тўғрисида”ги Қонуни.
7. 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-сонли Фармони.
8. Ўзбекистон Республикаси Президентининг 2017 йил 7 февралдаги “Ўзбекистон Республикасини янада ривожлантириш бўйича Ҳаракатлар стратегияси тўғрисида”ги 4947-сонли Фармони.
9. Ўзбекистон Республикаси Президентининг 2018 йил 3 февралдаги “Хотин-қизларни қўллаб-қувватлаш ва оила институтини мустаҳкамлаш соҳасидаги фаолиятни тубдан такомиллаштириш чора-тадбирлари тўғрисида”ги ПФ-5325-сонли Фармони.
10. Ўзбекистон Республикаси Президентининг 2019 йил 17 июндаги “2019-2023 йилларда Мирзо Улуғбек номидаги Ўзбекистон Миллий университетида талаб юқори бўлган малакали кадрлар тайёрлаш тизимини тубдан такомиллаштириш ва илмий салоҳиятини ривожлантириш чора-тадбирлари тўғрисида”ги ПҚ-4358-сонли Қарори.
11. Ўзбекистон Республикаси Президентининг 2019 йил 11 июлдаги «Олий ва ўрта махсус таълим тизимига бошқарувнинг янги тамойилларини жорий этиш чора-тадбирлари тўғрисида»ги ПҚ-4391- сонли Қарори.
12. Ўзбекистон Республикаси Президентининг 2019 йил 11 июлдаги «Олий ва ўрта махсус таълим соҳасида бошқарувни ислоҳ қилиш чора-тадбирлари тўғрисида»ги ПФ-5763-сон фармони.
13. Ўзбекистон Республикаси Президентининг 2019 йил 27 августдаги “Олий таълим муассасалари раҳбар ва педагог кадрларининг узлуксиз малакасини ошириш тизимини жорий этиш тўғрисида”ги ПФ-5789-сонли фармони.
14. Ўзбекистон Республикаси Президентининг “2019-2021 йилларда Ўзбекистон Республикасини инновацион ривожлантириш стратегиясини

тасдиқлаш тўғрисида”ги 2018 йил 21 сентябрдаги ПФ-5544-сонли Фармони.

15. Ўзбекистон Республикаси Президентининг 2019 йил 27 майдаги “Ўзбекистон Республикасида коррупцияга қарши курашиш тизимини янада такомиллаштириш чора-тадбирлари тўғрисида”ги ПФ-5729-сон Фармони.

16. Ўзбекистон Республикаси Президентининг 2017 йил 2 февралдаги “Коррупцияга қарши курашиш тўғрисида”ги Ўзбекистон Республикаси Қонунининг қоидаларини амалга ошириш чора-тадбирлари тўғрисида”ги ПҚ-2752-сонли қарори.

17. Ўзбекистон Республикаси Президентининг "Олий таълим тизимини янада ривожлантириш чора-тадбирлари тўғрисида”ги 2017 йил 20 апрелдаги ПҚ-2909-сонли қарори.

18. Ўзбекистон Республикаси Президентининг “Олий маълумотли мутахассислар тайёрлаш сифатини оширишда иқтисодиёт соҳалари ва тармоқларининг иштирокини янада кенгайтириш чора-тадбирлари тўғрисида”ги 2017 йил 27 июлдаги ПҚ-3151-сонли қарори.

19. Ўзбекистон Республикаси Президентининг “Нодавлат таълим хизматлари кўрсатиш фаолиятини янада ривожлантириш чора-тадбирлари тўғрисида”ги 2017 йил 15 сентябрдаги ПҚ-3276-сонли қарори.

20. Ўзбекистон Республикаси Президентининг “Олий таълим муассасаларида таълим сифатини ошириш ва уларнинг мамлакатда амалга оширилаётган кенг қамровли ислохотларда фаол иштирокини таъминлаш бўйича қўшимча чора-тадбирлар тўғрисида”ги 2018 йил 5 июндаги ПҚ-3775-сонли қарори.

21. Ўзбекистон Республикаси Вазирлар Маҳкамасининг 2012 йил 26 сентябрдаги “Олий таълим муассасалари педагог кадрларини қайта тайёрлаш ва уларнинг малакасини ошириш тизимини янада такомиллаштириш чора-тадбирлари тўғрисида”ги 278-сонли Қарори.

Ш. Махсус адабиётлар

1. Duvanov A.A. WEB. Construction of HTML. - St. Petersburg: BHV, 2003.
2. Barrett D. JavaScript. Web-professionals. - Kiev: BHV - Kyiv, 2001.
3. Brandenbau D. JavaScript: a collection of recipes. - St. Petersburg.: Peter, 2000.
4. Budilov B. JavaScript, XML and the Document Object Model. - St. Petersburg.: S & T, 2001.
5. R. Wagner JavaScript. Encyclopedia member (+ CD-ROM). - Kiev: DiaSoft, 2001.
6. Vayk A. JavaScript in examples. - Kiev: DiaSoft, 2000.
7. Rev. E. Vander JavaScript for "Dummies." - Dialectic, 2001.
8. P. Weiner JAVA programming languages, and JavaScript. - M: LORI, 2000.
9. Garnaev A Web-programming in Java and JavaScript. - St. Petersburg.: BHV St. Pereburg, 2002.
10. Dmitriev M. Tutorial JavaScript. - St. Petersburg.: BHV St. Pereburg, 2001.
11. Dmitriev M. JavaScript. Quick start. - St. Petersburg.: BHV St. Pereburg, 2002.

Интернет сайтлари

1. <http://www.intuit.ru>
2. <http://etuit.uz/>
3. www.tuit.uz
4. www.ziyonet.uz

