

14-ЛАБОРАТОРИЯЛЫҚ ЖҰМЫС. JAVASCRIPT ТІЛІНДЕГІ ОБЪЕКТІЛЕР

1. Объект түсінігі. Тәсілдер мен қасиеттер

Объект — бұл мәліметтер мен функциялар жиынтығынан тұратын бірыңғай конструкция немесе JavaScript терминологиясында қасиеттер мен тәсілдер жиынтығы.

Функция = тәсіл (метод).

Айнымалы = қасиет (свойства).

Инкапсуляция термині – «қара жәшік» ретінде қарастырылатын объектінің ішкі құрылымын жасыру деген сөз. Объектінің қасиеттері белгілі болып саналады, яғни олар – сырттан қол жеткізуге болатын айнымалылар. Бірақ бұл функциялардың қалай құрылғандығы, олардың қандай алгоритммен жұмыс істейтіндігі туралы программа-лаушыға айтылмайды. Программалаушы немесе объектіні тұтынушы адам объектінің қосымша ішкі функциялары мен айнымалылары бар ма, олар қол жеткізуге болатын қасиеттер мен тәсілдермен қалай байланысқан, ол жағын білмейді.

1.1. Объект және объектінің бір данасы (экземпляры)

Мысалы, нақты теледидар – бұл JavaScript терминологиясында объект емес, ол объектінің бір данасы (экземпляры). Объект болып зауытта нақты өнім шығаруға арналған құжаттамалар комплексі саналады. Конвейерден шығып жатқан барлық теледидарлар бейнелерінің *қасиеттері* бірдей және оларды басқаратын *тәсілдер* де бірдей экземпляр болып табылады.

Программалауда да осылай. Объект — бұл шаблон, құжаттар жиынтығы. Объектінің бір данасы – ол оның жұмыс көшірмесі ғана.

1.2. Объект интерфейсі және объектінің ішкі құрылымы.

Rectangle объектісі

Мысал ретінде Rectangle объектісін қарастырайық. Ол тіктөртбұрыштармен жұмыс істеуге арналған объект. Объектінің бір данасын жасау үшін былай жазу керек:

```
var x = new Rectangle(a, b);
```

// Мұндағы x Rectangle объектінің бір данасы.

// **a** мен **b** тіктөртбұрыш ені мен биіктігі.

// **new** сөзі бір дана жасау үшін керек.

Объектінің бір данасы жасалған соң, келесі тәсілдер мен қасиеттерді пайдалануға болады:

Қасиеттер:

width

height

Тәсілдер:

square()

perimeter()

radius()

Пайдалану мысалы:

```
var p = x.perimeter();
```

```
var r = x.radius();
```

```
var m = x.width;
```

```
if(x.height > m)
```

```
  m = x.height;
```

Сипатталуы:

Тіктөртбұрыш ені

Тіктөртбұрыш биіктігі

Сипатталуы:

Тіктөртбұрыш ауданы

Тіктөртбұрыш периметрі

Сырттай сызылған шеңбердің радиусы

Периметрді есептеу

Сырттай сызылған шеңбердің радиусын есептеу

Үлкен қабырғасын анықтау

Мұнда объект *интерфейсі* келтірілген, яғни объектімен қатынасуға қажет ақпарат берілген.

Мұнда square, perimeter, radius функцияларының программалық кодтары келтірілмеген. Басқаша айтқанда, объектінің ішкі құрылымы көрсетілмеген. Жұмыс кезінде тек интерфейсті пайдаланып, оның ішкі құрылымын пайдаланбауға болады.

Объект интерфейсі – пайдалануға болатын объектінің айнымалылары мен функциялары.

Объектінің ішкі құрылымы – программалау тілінде объектінің ішкі айнымалылары мен функцияларын сипаттау.

JavaScript тілінде *Rectangle* объектісінің *x* экземплярының қасиеттері мен тәсілдері жазылады:

x.height; x.perimeter();

Бұлардың жалпы жазылу форматы мынадай болады:

экземпляр_аты.объект_қасиеті_не_тәсілі

мұндағы нүкте – сатылы бөлу таңбасы: ол тегін мұрагерінен (атасын баласынан) бөліп тұрады.

1.3. Құрамдас ішкі объектілер және тұтынушы объектісі

JavaScript тілінде ішкі құрамдас объектілер көп. Оларды программалау қажет емес, олар тіл ішінде орнатылған. Бұл – браузердің программалық кодына осы объектілер коды кіреді деген сөз. Программалаушы осы объектілердің интерфейсін білуі тиіс, олардың бір экземплярын жасай білуі керек, сонда ол өз қалауынша ішкі объектілерді пайдалана алады. JavaScript жаңа объектілерді программалауға және олардың ішкі объектілерін өзгертуге мүмкіндік береді. Енді бірнеше ішкі объект жұмыстарын қарастырайық.

2. Date объектісі

JavaScript тілінің ішкі объектілерін қарастыруды өте пайдалы болып саналатын **Date** объектісінен бастайық. Бұл объект күн-ай мерзімімен және уақытпен жұмыс істеу үшін керек.

Объект экземплярын жасау үшін (**Date** объектісінің ғана емес, одан басқасының да) JavaScript тілінде **new** түйінді сөзі қолданылады:

var now = new Date();

Енді **now** айнымалысы **Date** объектісінің экземпляры болып табылады да, ол үстіміздегі күн-ай мерзімі мен уақытты береді. Жалпы экземпляр жасау былай орындалады:

var айнымалы = new Date (параметрлер);

Мысалы, келесі параметрлерді көрсетуге болады:

Келесі параметрлерді көрсетуге болады:

var now = new Date();

var birthday = new Date(1954, 1, 8);

var bell = new Date(2003, 0, 14, 12, 20, 0);

Date объектісінің бір данасы құрылғаннан кейін, оның ішкі мәліметтерін көруге болады, оны өзгерту мүмкіндігі де бар. Ол үшін көптеген тәсілдер бар, олардың тізімі тілге арналған кітап қосымшаларында келтіріледі.

Объект тәсілінің аты (тәсіл – JavaScript терминологиясындағы функция) экземпляр атынан нүктемен бөлініп жазылады.

Былай жазуға болады:

var year = bell.getYear();

year айнымалысы мәні 2003 болады.

Date объектісінің бірнеше қарапайым скриптілерін қарастырайық.

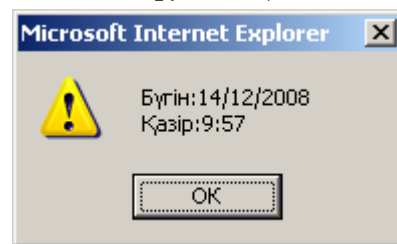
Ағымдағы күн-ай мерзімі және уақыт (14.1-сурет):

var now = new Date();

alert("Бүгін:" + now.getDate() + "/" +

now.getMonth() + 1) + "/" + now.getYear() +

"\nҚазір:" + now.getHours() + ":" + now.getMinutes());

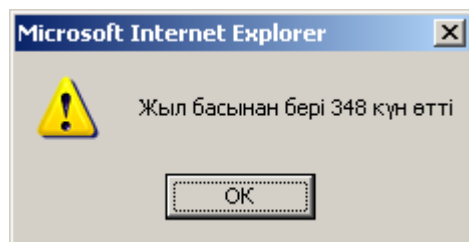


14.1-сурет

Осы кодтарды жазып, оны орындау нәтижесінде 14.1-суреттегі уақытты көрсететін хабарлама шығады.

Жыл басынан осы сәтке дейінгі күндер саны (14.2-сурет):

```
var now = new Date(); // Ағымдағы мерзім мен уақыт.  
var begin = new Date(now.getFullYear(), 0, 1);  
// Ағымдағы жыл басы.  
// жыл басы ішіндегі миллисекунд саны:  
var num=now.getTime()-  
begin.getTime();  
var msPerDay = 24 * 60 * 60 * 1000;  
// тәулік ішіндегі миллисекунд саны.  
num /= msPerDay;  
// жыл басынан бергі күндер саны.  
// Нәтижесін көрсетеміз:  
alert("Жыл басынан бері " +  
Math.floor(num) + " күн өтті");
```



14.2-сурет

Осы кодтарды жазған кезде скриптіні орындау 14.2-суреттегі хабарламаның шығуына себепші болады.

2.1. Date объектісінің тәсілдері

1. Тәсіл түрі – `getFullYear()`. Жыл нөмірін береді (14.3-сурет).

Мысалы:

```
var d = new Date(2008,1,2);  
var y = d.getFullYear();  
alert(y);
```



14.3-сурет

2. Тәсіл түрі – `setYear()`. Жыл нөмірін тағайындайды.

Мысалы:

```
var d = new Date();  
d.setYear(2008);  
alert(d.getFullYear());
```



14.4-сурет

3. Тәсіл түрі – `getMonth()`. Ай нөмірін береді (14.4сурет).

```
var d = new Date(2008,1,2);  
var m = d.getMonth();  
alert(m);
```

4. Тәсіл түрі – `setMonth()`. Айды енгізеді.

Мысалы:

```
var d = new Date();  
d.setMonth(4);  
alert(d.getMonth());
```



14.5-сурет

5. Тәсіл түрі – `getDate()`. Ай күнін береді (14.5сурет).

```
var d = new Date(2008,1,2);  
var m = d.getDate();  
alert(m);
```

6. Тәсіл түрі – `setDate()`. Айдың күнін енгізеді (тағайындайды).

Мысалы:

```
var d = new Date();  
d.setDate(2);
```

alert(d.getDate());

7. Тәсіл түрі – `getDay()`. Апта күнін береді (жексенбі – 0, дүйсенбі – 1, ..., сенбі – 6):

var d = new Date(2008,1,2);

var m = d.getDay();

alert(m);

8. Тәсіл түрі – `getHours()`. Уақыттың сағатын береді:

var d = new Date();

var t = d.getHours();

alert(t);

9. Тәсіл түрі – `setHours()`. Сағатты енгізеді (тағайындайды).

Мысалы:

var d = new Date();

d.setHours(22);

alert(d.getHours());

10. Тәсіл түрі – `getMinutes()`. Уақыт минутын береді.

var d = new Date();

var m = d.getMinutes();

alert(m);

11. Тәсіл түрі – `setMinutes()`. Уақыт минутын енгізеді.

var d = new Date();

d.setMinutes(32);

var t = d.getMinutes();

alert(t);

12. Тәсіл түрі – `getTime()`.

1970 жылдың қаңтар айының бірі күнгі 0 сағаттан осы сәтке дейінгі миллисекунд мөлшерін береді.

Мысалы:

var d = new Date();

var t = d.getTime();

alert(t);

Тапсырмалар:

1. Үстіміздегі күн-ай мерзімі мен уақытты анықтау керек. Сағат пен минутты жеке-жеке бөліп жазыңыздар.
2. 25 мамыр 2008 жыл мерзімін орнатыңыздар. Жылды, айды және күнді жеке-жеке бөліп жазыңыздар.
3. Жоғарыдағы күннің аптаның қай күні болатынын анықтаңыздар.
4. Мына уақытты: 15 сағат 25 минут және 32 секунд орнатыңыздар.
5. 21 ғасыр басынан бері неше күн өткенін анықтап, 20 ғасыр басынан бері де неше күн өткенін де табыңыздар.
6. Өзіңіздің неше күн, неше минут және неше секунд өмір сүргеніңізді анықтаңыз.
7. Ұлы Отан соғысы аяқталғаннан бері неше жыл, неше күн, неше сағат өтті?

3. Array объектісі

Бұл объект мәліметтер жиымын (массивін) жасау үшін керек. *Жиым – элементтердің реттелген жиынтығы.* Жеке элементтің орны оның аты мен индексі (нөмір) көрсету арқылы орындалады. JavaScript тілінде элементтерді нөмірлеу нөлден басталады.

Мысал: апта күндерінің аттары жиымы.

```
var dayNames = new Array("жексенбі", "дүйсенбі", "сейсенбі", "сәрсенбі",  
"бейсенбі", "жұма", "сенбі");
```

Жиымның жеке элементтерін пайдалану үшін былай жазылады:

массив_аты [индекс]

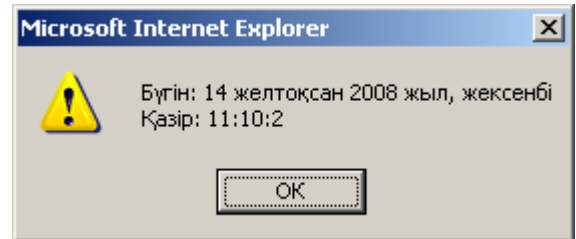
Төмендегі скрипт:

```
var dayNames = new Array("жексенбі", "дүйсенбі", "сейсенбі", "сәрсенбі",  
"бейсенбі", "жұма", "сенбі"); alert(dayNames[0]);
```

осы скрипт жұмысы нәтижесінде **alert** терезесіне «жексенбі» мәтіні шығады.

Мысалы: ағымдағы күн-ай мерзімі мен уақытты көрсету қажет болсын делік (14.6-сурет).

```
// Ай аттарын көрсету:  
var monthNames = new Array  
("қаңтар", "ақпан",  
"наурыз", "сәуір", "мамыр", "маусым",  
"шілде", "тамыз", "қыркүйек",  
"қазан", "қараша", "желтоқсан");  
// Апта күндері:  
var dayNames = new Array  
("жексенбі", "дүйсенбі", "сейсенбі",  
"сәрсенбі", "бейсенбі", "жұма", "сенбі");  
var today = new Date();// Ағымдағы мерзім мен уақыт.  
// Нәтижені шығаруға дайындау:  
var str = "Бүгін: "  
// Айдың күні:  
str += today.getDate() + " ";  
// айдың аты: str+=monthNames[today.getMonth()]+ " ";  
// жыл:  
str += today.getYear() + " жылы, ";  
// Апта күні:  
str += dayNames[today.getDay()] + "\n";  
// Уақыт:  
str += "Бүгін: " + today.getHours() + ":" +  
today.getMinutes() + ":" +  
today.getSeconds();  
// Нәтижені шығару:  
alert(str);
```



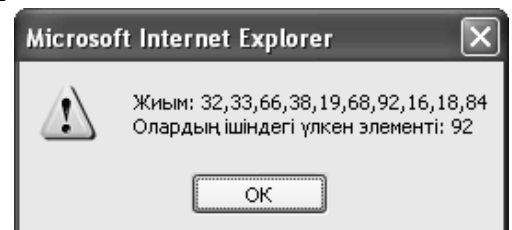
14.6-сурет

Жиым ұзындығы (оның элементтерінің саны) программа жұмысы кезінде өзгере алады:

```
var f = new Array(); // Қазір жиым бос, оның элементі жоқ.  
f[0] = 1; // Жиымда бір элемент бар.  
f[1] = 1; // Жиымда екі элемент бар.  
f[2] = f[0] + f[1]; // Жиымда үш элемент бар.  
f[5] = 8; // Жиымда алты элемент бар  
// f[0]...f[5]
```

Мысалы: Жиымның ең үлкен элементін анықтау керек болсын делік (14.7 сурет).

```
// num кездейсоқ сандардан жиым құрайық,  
// олардың әрқайсысы [a,b] аралығында жатады  
var num = 10; // кездейсоқ сандардың саны  
var a = 1; // аралықтың сол жақ шекарасы  
var b = 100; // аралықтың оң жақ шекарасы  
var set = new Array(); // жиым құрылды  
// жиымды кездейсоқ сандармен толтыру
```



14.7-сурет

```

for (i=0; i<num; i++)
set[i] = Math.round(a+(b-a)*Math.random());
// Ең үлкен элементін анықтау
var max = a;
for (i=0; i<num; i++)
if(set[i] > max) max = set[i];
// Жиым элементтері мен үлкен элементін көрсетейік:
alert("Жиым: "+set+"\nОлардың ішіндегі үлкен
элементі: "+max);

```

Қарастырылған мысалда екі цикл бір циклге біріктірілген:

```

var max = a;
for(i = 0; i < num; i ++){
set[i]=Math.floor(a+(b-a+1)*Math.random());
if(set[i] > max) max = set[i];
}

```

Осыған дейін біз объектілер тәсілдерімен танысқанымызбен, олардың қасиеттерін қарастырмадық.

Объект қасиеттері дегеніміз JavaScript терминологиясы бойынша объект тұтынушысына арналған интерфейстік айнымалылар болып табылады. Негізінде, мұнда объект туралы емес, тек объектінің нақты бір данасы, яғни экзemplяры жайлы сөз болады. Қасиеттерді пайдалану үшін объект экзemplяры атымен нүкте арқылы бөлінген қасиет аты жазылады, мысалы:

```
var set=new Array("Алпамысов", "Азамат", "Адамұлы"); alert
(set.length);
```

length қасиеті 14.8-суретте көрсетілгендей массив элементтерінің санын – оның ұзындығын береді.

JavaScript тілінде объектілермен жұмыс істеу кезінде бірсыпыра «еркіндіктерге» жол беріледі. Мысалы, массив экзemplярын **new** түйінді сөзінсіз және **Array** объектісін де көрсетпей жазуға болады:

```
var set = [1, 4, 9, 16, 25, 36];
```

Әрине, браузер, мұндай жазбаны кездестіріп, бәрібір **Array** объектісінің бір данасын төмендегі жазбада берілгендей түрде ашады:

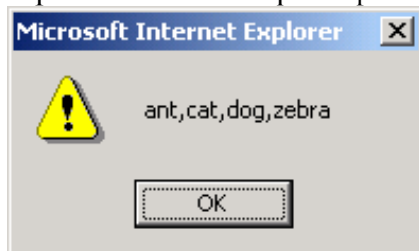
```
var set = new Array(1, 4, 9, 16, 25, 36);
```

3.1. Array объектісінің sort тәсілі

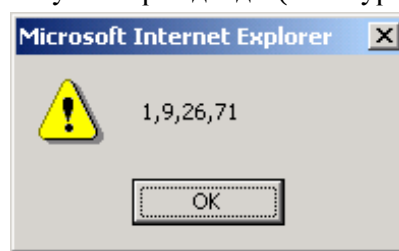
Array объектісінің **sort** тәсілін қарастырайық. Мәліметтерді сұрыптау үшін **sort(function)** немесе **sort()** тәсілі енгізілген, мұнда **function** параметрі екі элементті салыстыру ережесін береді, ол болмаған жағдайда сұрыптау лексографикалық тәртіпте жүргізіледі. Мысалы:

```
var set = new Array("zebra", "ant", "dog", "cat");
set.sort(); alert(set);
```

Бұл жиым элементтерін алфавит бойынша сұрыптау ісін орындайды (14.9-сурет).



14.9-сурет



14.10-сурет



14.8-сурет

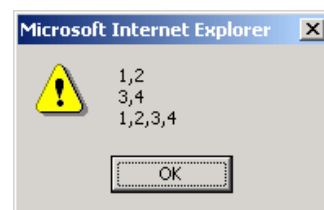
Негізінде, **sort(function)** тәсілінде функцияның екі аргументі болуы тиіс, ол мынадай мәндер қайтарады:

— теріс сан, реттелуі бойынша бірінші аргументтің мәні екіншісінен төмен болып, одан сол жақта орналасқанын білдіреді;

— 0, аргументтерінің мәндері тең болғанын көрсетеді;

— оң сан, реттелуі бойынша бірінші аргумент екіншісінен артық болып, оның оң жақта орналасқандығының белгісі болып табылады (14.10-сурет). Мысалы:

```
var set = new Array(26,71,9,1);  
function Compare(a,b)  
{ return a - b; }  
set.sort(Compare);  
alert(set);
```



14.11-сурет

3.2. Array объектісінің тәсілдері

1. Тәсіл түрі – **concat(array)**. Бұрынғы массивке **array** массиві қосылған жаңа массив жасап береді. Бастапқы массив өзгермейді.

Мысалы:

```
var set1 = new Array(1,2);  
var set2 = new Array(3,4);  
var set = set1.concat(set2);  
alert(set1+"\n"+set2+"\n"+set);
```

2.Тәсіл түрі – **join (ажыратқыш)**. Ажыратқыш арқылы бөлініп орналасқан массив элементтері тізбегін құрайды. Бастапқы массив өзгермейді.

Мысалы:

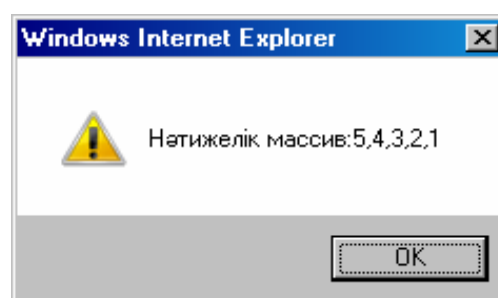
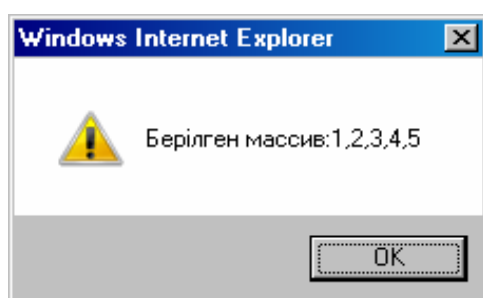
```
var set= new Array(1,2,3,4,5);  
var set2 = set.join("+");  
alert(set1+"\n"+set2);
```

14.12-сурет



3.Тәсіл түрі – **reverse()**. Массив элементтерін оның 1-элементі соңғысы болатындай етіп кері бағытта орын ауыстырып береді.

```
var set= new Array(1,2,3,4,5);  
alert("Берілген массив:"+set);  
var y = set.reverse();  
alert(" Нәтижелік массив:"+y);
```



14.13-сурет. Берілген және реттелген массивтер

4.Тәсіл түрі – **slice(ind1,ind2)** не **slice(ind1)**. Бастапқы массивтен позициясы ind1-ден бастап ind2-1 позициясына дейінгі элементтерден тұратын жаңа массив жасап береді. Егер 2-индекс жоқ болса, онда массив соңына дейінгі элементтер алынады.

```
var set= new Array(0,1,2,3);  
var set1= set.slice(1,3);  
var set2= set.slice(1);  
alert("set="+set+"\nset1="+  
set1+"\nset2="+set2);
```



14.14-сурет

5. **length** тәсілі – массив ұзындығы (оның элементтері саны). Мысалы:
`var set = new Array(0,1,2,3,4,5,6,7,8,9,10);`
`alert(set.length);`



Нәтижесі 11 болады (14.8-суретті қара).

Тапсырмалар:

Мына скриптілерден соң **S** айнымалысының мәні қандай болады?

- а) `var a = new Array (1, 2 , 3, 4 , 5, 6, 7, 8, 9, 10);`
`var s = 0;`
`for(var i = 0; i < a.length; i++)`
`s+= a[i];`
- ә) `var a = new Array(1,2,3,4,5,6,7);`
`var s = 0;`
`for(var i = a.length; --i >= 0;)`
`s += a[i];`
`s/= a.length;`
- б) `var a = new Array(6,2,3,4,7,1,5);`
`var s = a[0] ;`
`for (var i = 0; i < a.length; i++)`
`if (a[i] > s) s = a[i];`
- в) `var a = new Array (6,2,3,4,7,1,5);`
`var d = a[0];`
`var s = 0;`
`for (var i = 0; i < a.length; i++)`
`if(a[i] > d) { d = a[i]; s = i; }`
- г) `var a = new Array("2","48","000","99") ;`
`a.sort(Compare);`
`var s = a[0];`
`function Compare(a,b)`
`{ return b.length-a.length; }`
- д) `var a = new Array("34","81","234","6739");`
`a.sort(Compare) ;`
`var s = a[0];`
`function Compare(a,b)`
`{`
`if (a < b) return 1;`
`else if(a > b) return -1;`
`else return 0;`
`}`
- ж) `var str = new String("3481234627892");`
`var s = 0;`
`for (var i = 0; i < str.length; i++)`
`if (str.charAt(i) == "2") s++;`
- з) `var str = new String ("3481234627892");`
`var s=0;`
`for(var i = str.indexOf("2"); i >= 0;`
`i = str.indexOf("2",i + 1)) s++;`
- и) `var str = new String ("университет");`
`var sub = str.substring(3,5);`
`var s = 0;`
`for (var i = str.indexOf(sub); i >= 0;`
`i = str.indexOf(sub,i + 1)) s++;`

4. String объектісі

Осыған дейін біз **String** объектісінің бір данасын – экземплярын басқа жолдармен былай құрған болатынбыз:

```
var title = "Ақыл арымас, алтын шірімес";
```

тізбегі төмендегі жазбамен бірдей болып табылады:

```
var title = new String("Ақыл арымас, алтын шірімес");
```

Жалпы **String** объектісінің бір данасы былай құрылады:

```
new String(str);
```

мұндағы **str** параметрі тіркестік өрнекті тағайындайды. Айнымалыны төмендегідей түрде сипаттап ашқанда, объект экземпляры, келісім бойынша, бірден құрылады:

```
var str = "Студенттер! Емтиханға дайындалыңдар.";
```

Бұл тіркесті былай да құруға болады:

```
var str = new String("Студенттер! Емтиханға дайындалыңдар.");
```

4.1. String объектісінің тәсілдері

1. **charAt(ind)** тәсілі берілген тіркестің **ind** позициясында (орынында) тұрған символды береді (нөмірлеу нөлден басталады). Мысалы (14.15-сурет):

```
var str = new String("абвгд");  
var ind = 1;  
alert(str.charAt(ind));
```



14.15-сурет

2. **indexOf(Substr)** немесе **indexOf(Substr, StartInd)** тәсілдері. Мұнда сөз тіркесіндегі **Substr** ішкі тізбегі сөздің **StartInd** позициясынан бастап, сөз соңына дейін ізделеді. Тәсіл нәтижесі болып табылған сөз тіркесінің алғашқы позициясының нөмірі есептеледі. Егер **StartInd** берілмесе, іздеу нөлінші позициядан басталады. Ондай сөз тіркесі табылмаса, тәсіл нәтижесі **-1**-ге тең болып саналады. Мысалы:

```
var str = new String("Қожа Ахмет Яссауи");  
var ind = str.indexOf("мет");  
alert(ind);
```

Бұл скриптің нәтижесі 7 болады.

3. Тәсіл түрі – **lastIndexOf(Substr)** немесе **lastIndexOf(Substr, StartInd)**. Мұнда сөз тіркесіндегі **Substr** ішкі тізбегі сөздің **StartInd** позициясынан бастап, сөздің басына дейін ізделеді. Тәсіл нәтижесі болып табылған сөз тіркесінің алғашқы позициясының нөмірі есептеледі. Егер **StartInd** берілмесе, іздеу сөз тіркесінің соңғы позициясынан басталады. Ондай сөз тіркесі табылмаса, тәсіл нәтижесі **-1**-ге тең болып саналады. Мысалы:

```
var str = new String("Қожа Ахмет Яссауи");  
var ind = str.lastIndexOf("ожа", 10);  
alert(ind);
```



14.16-сурет

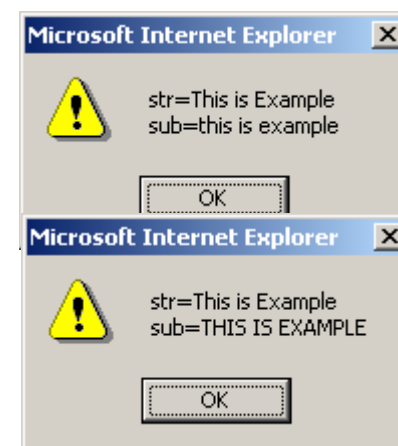
4. Тәсіл түрі – **substring(ind1)** немесе **substring(ind1, ind2)**. **ind1** позициясынан басталып, **ind2-1** позициясымен аяқталатын ішкі сөз тіркесін береді. Егер:

- **ind1** нөлден кіші болса, ол нөл болып есептеледі;
- **ind2** сөз тіркесі ұзындығынан артық болса, ол сол тіркес ұзындығына тең болып саналады;
- **ind1 ind2**-ге тең болса, онда бос сөз тіркесі қайтарылады.
- **ind1** көрсетілмесе, **ind1**-ден басталатын сөз соңы беріледі. Мысалы:

```
var str = new String("Қожа Ахмет Яссауи");  
var sub = str.substring(6, 9);  
alert("str="+str+"\nsub="+sub);
```

5. Тәсіл түрі – **toLowerCase()**. Бұл сөз тіркесін төменгі регистрге ауыстырып береді, яғни бас әріптерді кіші әріптерге түрлендіреді. Мысалы (14.17-сурет):

```
var str = "This is Example";  
var sub = str.toLowerCase();  
alert("str="+str+"\nsub="+sub);
```



14.18-сурет

6. Тәсіл түрі – `toUpperCase()`. Бұл сөз тіркесін жоғарғы регистрге ауыстырып береді, яғни кіші әріптерді бас әріптерге түрлендіреді. Мысалы (14.18-сурет):

```
var str = "This is Example";  
var sub = str.toUpperCase();  
alert("str="+str+"\nsub="+sub);
```

7. Қасиет түрі – `length`. Ол сөз тіркесінің ұзындығын анықтайды. Мысалы (14.19-сурет):

```
var str = "Біз студентпіз";  
alert(str.length);
```



14.19-сурет

Бақылау сұрақтары:

1. Тіркестік айнымалылар қалай сипатталады?
2. Сөз тіркесінің бір данасы қалай құрылады?
3. Тіркестік айнымалының немесе сөз тіркесінің ұзындығы қалай анықталады?
4. Тіркестің ішкі символдарын қалай бөліп алуға болады?
5. Тіркестік айнымалылар мен тұрақтыларға қандай тәсілдер қолданылады? Оларды қалай пайдаланады және олар қалай жазылады?

ТАПСЫРМАЛАР:

1. Топ студенттері фамилиялары мен аттарының бірінші әрпін шығарыңыздар.
2. Әрбір сөзден кейін бір бос орын қалдырылған сөйлемдер берілген. Құрамында берілген әріптен басталатын сөздердің санын анықтаңыздар.
5. Әрбір сөзден кейін бір бос орын қалдырылған сөйлем берілген. Сөйлемдегі ең ұзын сөзді табыңыздар.
6. Әрбір сөзден кейін бір бос орын қалдырылған сөйлем берілген. Сөйлем ішіндегі жақшалардың дұрыс қойылғанын тексеріңіздер.
8. Берілген сөз тіркесіндегі әрбір нүктені көп нүктемен (яғни үш нүктемен) алмастырыңыздар.
9. Алдыңғы есеп шартындағы қатар келген нүктелердің әрбір тобын бір нүктемен алмастырыңыздар.
10. Дүкендегі кассир көмекшісіне арналған программа құрыңыздар. Программа тауардың бағасын, мөлшерін, сатып алынған тауарлар бағасының қосындысын есептеп, сатып алушының берген ақшасының мөлшерін сұрап, оған қайтарылатын соманы да анықтайды.
11. Әрбір сөзден кейін бір бос орын қалдырылған сөйлем берілген. Барлық сөздерді керісінше жазып шығыңыздар.
12. Пернетақтадан енгізілген сөздердегі әріптердің санын есептейтін программаны тәсілдерді қолданбай, тапсырманы `do ... while` циклінде орындаңыздар.
17. Енгізілген сөз тіркесіндегі сөздердің санын анықтайтын программа жазыңыздар. Бір сөз екіншісінен бір бос орын арқылы айырылады деп санау керек.
18. Мәтіні пернетақтадан енгізілген телеграмманың бағасын есептейтін программа жазыңыздар.
19. Берілген сөзде бірінші және соңғы әріптердің қайсысы көп кездесетінін анықтайтын программа жазыңыздар.
20. “а” әрпімен аяқталатын атау септігіндегі зат есім берілген. Осы сөзді септеп, басып шығарыңыздар.