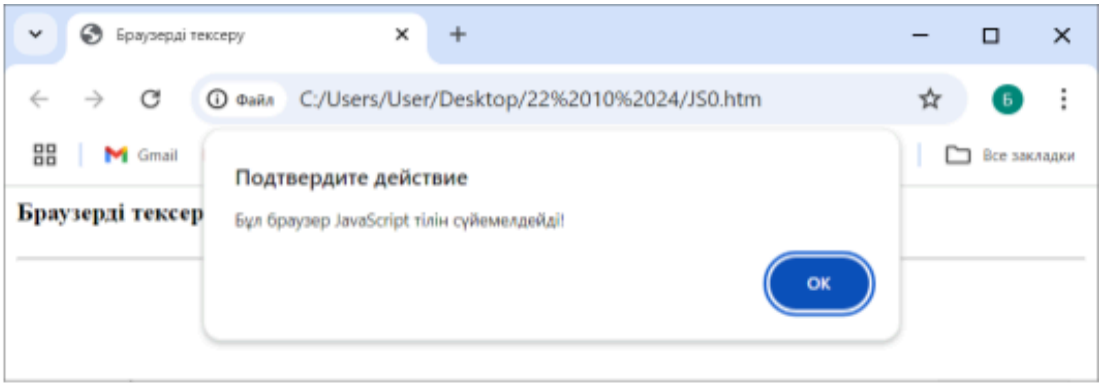
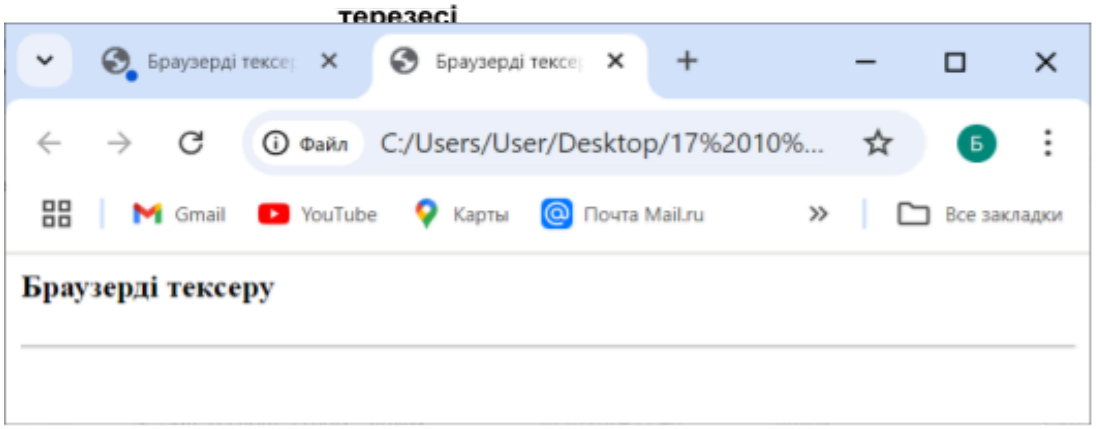
HTML-тіліне скриптер енгізу (alert, prompt, confirm)



alert ақпараттық терезесі



Скриптерді көрсететін браузердің ОК батырмасын басқан соңғы көрінісі

Мұнда JavaScript тілінің экранға мәлімет шығаратынalert **alert** функциясы қолданылған, ол ішінде **ОК** батырмасы бар ақпараттық шағын терезе ашады.

Тұтынушы мәліметті оқыған соң, ОК батырмасын басады, сонда ақпараттық терезе жабылады.

Енді браузер HTML-кодтағы келесі командаларды орындай бастайды да, жоғарыдағы суреттегі мәліметті экранға шығарады.

```
<HTML>
 <HEAD>
        <TITLE>Браузерді тексеру</TITLE>
 </HEAD>
 <BODY bgcolor=white text=black link=blue
          alink=red vlink=purple>
  <H4> Браузерді тексеру </H4> <HR>
  <SCRIPT language=JavaScript>
  <!--
   alert("Бұл браузер JavaScript тілін сүйемелдейді!");
  -->
  </SCRIPT>
```

```
  <NOSCRIPT>
        <H2>Ескерту</H2>
        <P> Бұл браузер JavaScript тілін сүйемелдемейді.
  </NOSCRIPT>
 </BODY>
</HTML>
```

Егер браузер JavaScript тілін сүйемелдей алмайтын болса, онда <NOSCRIPT> **тәгі ішіндегі мәлімет экранға шығарылады.**

Егер alert функциясының аргументі ретіндегі мәтін көлемді болса, оны «+» (біріктіру операциясы) таңбасы арқылы бірнеше жолға жазуға болады, мысалы:

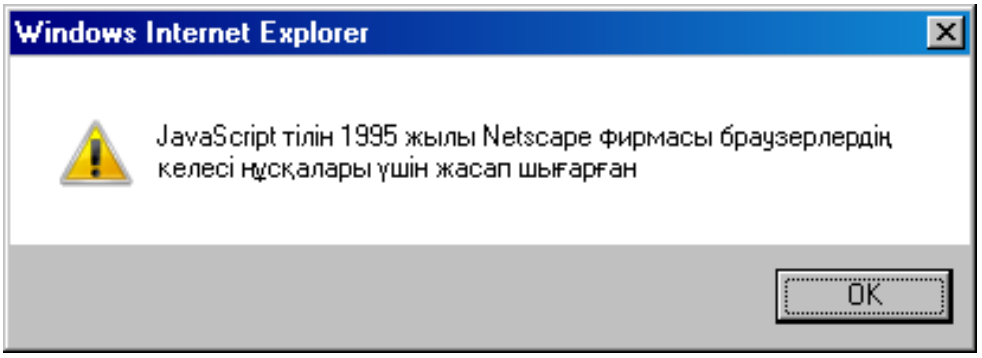
```
alert("JavaScript тілін 1995 жылы Netscape фирмасы " +
    "браузерлердің келесі нұсқалары үшін жасап
     шығарған");
```

Мұндай функция нәтижесі келесідей болады:

Егер alert функциясының аргументі ретіндегі мәтін көлемді болса, оны «+» (біріктіру операциясы) таңбасы арқылы бірнеше жолға жазуға болады, мысалы:

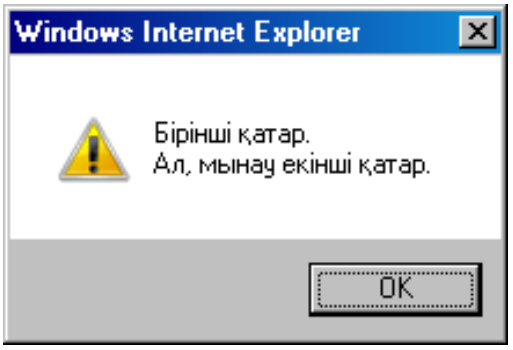
```
alert("JavaScript тілін 1995 жылы Netscape фирмасы " +
    "браузерлердің келесі нұсқалары үшін жасап
     шығарған");
```

Мұндай функция нәтижесі келесідей болады:



alert() функциясында, қажет болғанда мәліметті «\n» символы арқылы екі-үш жолға бөліп шығару мүмкіндігі де қарастырылған (сурет):

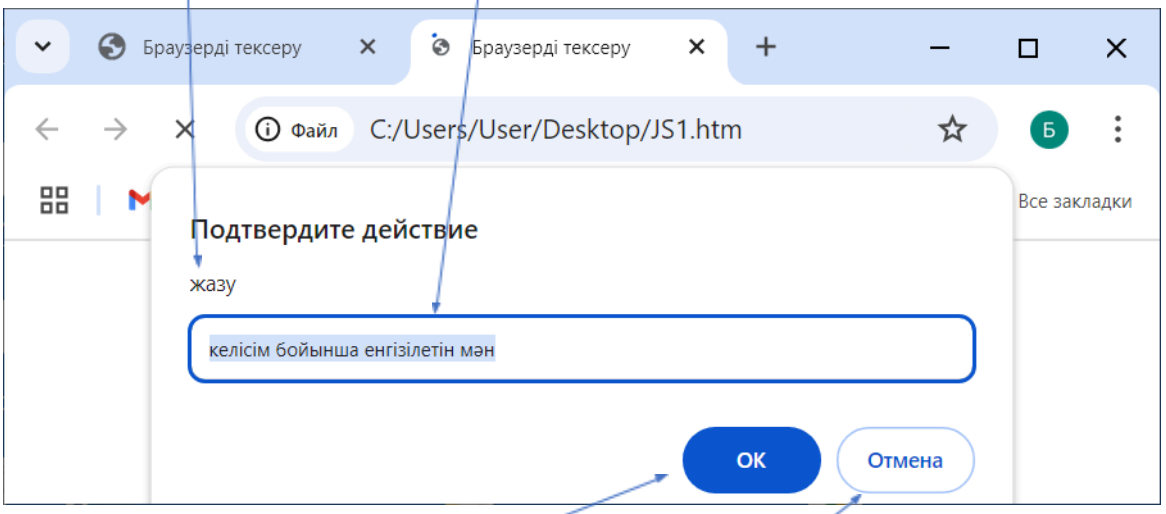
```
alert("Бірінші қатар.\nАл, мынау екінші қатар.");
```



Prompt() функциясы

prompt("жазу", "келісім бойынша енгізілетін мән");

мұнда экранға ішінде екі батырмасы бар терезе шығады. Біз жазба мәліметті енгізу жолына жазамыз да, ОК басамыз. Сонда терезе жоқ болады да, терезеге енгізілген мәліметтер шығады.



мұнда экранға ішінде екі батырмасы бар терезе шығады. Біз жазба мәліметті енгізу жолына жазамыз да, **ОК** басамыз. Сонда терезе жоқ болады да, терезеге енгізілген мәліметтер шығады.

Ол мәнді мысалы, айнымалыға меншіктеуге, артынан басқа командаларда пайдалануға болады. Егер **Cancel (Отмена)** батырмасын шертетін болсақ, онда **prompt** функциясы арнайы **null** мәнін қайтарады (бұл "null" сөз тіркесі емес, яғни бос жол ("") емес, мәліметтің арнайы мәні).

prompt функциясының экрандағы көрінісі

Мысалы:

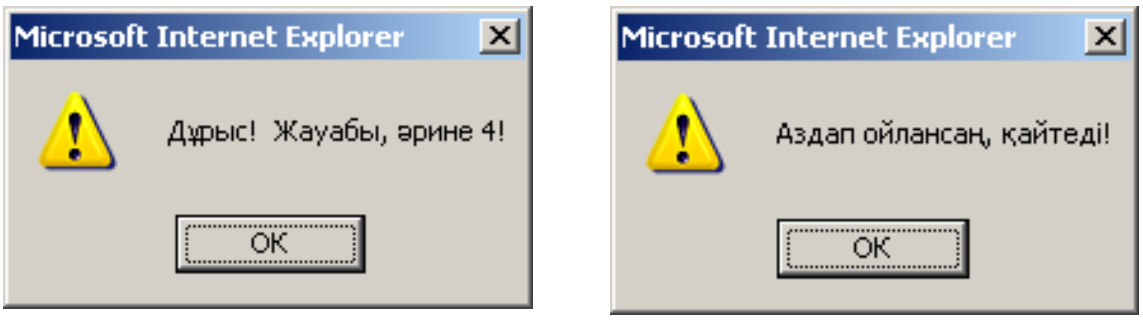
```
var str = prompt("2 х 2 қанша болады?", "5");
if (str == "4") alert ("Дұрыс!  Жауабы, әрине 4!");
else alert("Аздап ойлансаң, қайтеді!");
```

Осы скриптіні іске қосқанда экранға 12.2-суреттегі терезе шығады.

Скриптіні орындау нәтижесі

Егер 4 санын енгізіп, ОК батырмасын шертетін болсақ, скрипт жұмысы төмендегі суреттегідей болып жалғасады.

Егер енгізу өрісінде 5 санын қалдырсақ (не 4-тен басқа кез келген сан енгізсек), экранға төменгі суретте көрсетілген хабарлама шығады:



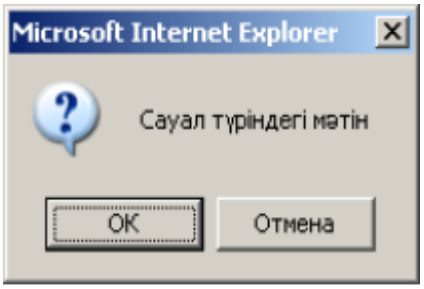
Мысалға 4 және 5 енгізу нәтижелері

confirm() функциясы

Бірақ бұдан гөрі қарапайым әрі жеңіл тәсіл бар, ол – **confirm** функциясын пайдалану:

```
confirm("жазу");
```

Бұл функция былай жұмыс істейді. Экранға екі батырмасы бар терезе шығады (12.7-сурет):



12.7-сурет. confirm функциясының экраны

Егер **ОК** батырмасын шертсек, **confirm** функциясы true мәнін қайтарады, ал егер **Отмена** (Cancel –

Болдырмау) батырмасы шертілсе – false мәні қайтарылады.

Олардың орнына соларға парапар (эквивалентті) <**Enter**> және <**Esc**> пернелерін де пайдалануға болады.

Мысалы:

```
if (confirm("Бізге дән де және ән де керек емес пе?"))
    alert ("Дұрыс ойлайсың!");
 else   alert("Дән азық болса,\n" + "ән көңіл көтермей ме!");
```

Осы кодты орындағанда, экранға оң жақ суреттегі терезе шығады. Содан соң **ОК** батырмасы шертілсе, төменгі сол жақ терезедегі мәнді шығарады, ал **Отмена (Болдырмау)** басылса, оң жақ терезедегі мән шығады.

confirm функциясын пайдалану мысалы

Логикалық тип мәндері

JavaScript тілінде айнымалылар логикалық типтегі мәндерді қабылдай алады. Мұндай тип **true** және **false** тәрізді екі түрлі мән қабылдайды.

Құрамына арифметикалық және тіркестік (строковые) өрнектер кіретін шарттар алғашқы кездерде қиындықтар туғызады. Оларды түсіну үшін алдымен шарттар құрамындағы арифметикалық және тіркестік өрнектер мәнін есептеп алу керек, содан соң барып олардың мәні логикалық true және false мәндерімен салыстырылады.

«**Қасқыр**» және «**крокодил**» сияқты **тіркестік константалар** және де **25** және **3.14** тәрізді **сандық константалардың барлығы да логикалық true мәніне сәйкес келеді**. Сәйкесінше, **солардың логикалық терістеу мәндері false болады**.

Төменгі скриптерден соң, **z** айнымалысы нешеге тең болады?

1-мысал	2-мысал
var x = "каша"; var y = "лот"; var z = "акула"; if (x) z = x + y;	var x = "каша"; var y = "лот"; var z = "акула"; if(!x) z = x + y;

3–мысал	4-мысал
var x = "каша"; var y = "лот"; var z = "акула"; if(x && y) z = x + y;	var x = "каша"; var y = "лот"; var z = "акула"; if(!x && y) z = x + y;

5-мысал	6-мысал
var x = "каша"; var y = "лот"; var z = "акула"; if(!x || y) z = x + y;	var x = "каша"; var y = "лот"; var z = "акула"; if(!x || y) z = x + y;

7-мысал	8-мысал

var x = -10;
var y = 10;
var *z* = x + y;
if(z) z = 20;
	var x = -10;
	var y = 10;
	var *z* = x + y;
	if(!z) z = y - x;

9-мысал	10-мысал

var x = -10;
var y = 10;
var *z* = x + y;
if(x < y) z = y - x;
	var x = -10;
	var y = 10;
	var *z* = x + y;
	if(x <= y) z = y - x;

11-мысал	12-мысал

var x = -10;
var y = 10;
var *z* = x / y;
if(x + y || x < 0) z = y - x;
	var x = -10;
	var y = 10;
	var *z* = x / y;
	if(x + y || x < 0) z = y - x;

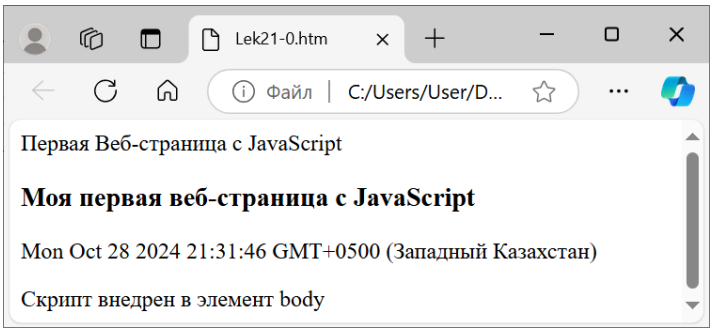
13-мысал	14-мысал

`var x = -10;`
`var y = 10;`
`var z = x / y;`
`if(x + y && x < 0) z = y - x;`
	`var x = -10;`
	`var y = 10;`
	`var z = x / y;`
	`if(x + y && y < 0) z = y - x;`
	`else z = x + y;`

Скриптерді құжат ішіне енгізу

Құжатқа скрипттерді кірістіру үшін <script> элементі пайдаланылады. <script> элементі скрипт командалары бар бар блоктың басы мен соңын білдіреді. Оның скрипт жазылған тілдің атауын көрсететін type атрибуты бар (біздің мысалымызда, келісім бойынша, ол JavaScript болып саналады). <body> элементіне енгізілген JavaScript скрипті бар программа үлгісін қарастырайық (Lek21-0.htm):

```
<html><head>Первая Веб-страница с JavaScript</head>
<body>
  <h3>Моя первая веб-страница с JavaScript</h3>
  <script type="text/javascript">
    document.write("<p>" + Date()  + "</p>");
    // Бұл қысқаша комментарий </script>
    Скрипт внедрен в body элементі іішіне енгізілген.
</body> </html>
```



Бұл мысалда JavaScript тілінің ең маңызды командаларының бірі - document.write() пайдаланылды. document.write() командасы ағымдағы құжатта жазба жасау үшін қолданылады. Бұдан басқа, бұл мысалда *уақытпен* және *даталармен* (күн-ай мерзімдерімен) жұмыс істеуге мүмкіндік беретін Date объектісі пайдаланылды.

JavaScript комментарийлері кодты түсінікті етуге мүмкіндік береді. қарапайым комментарий жол соңындағы // таңбаларынан басталады.

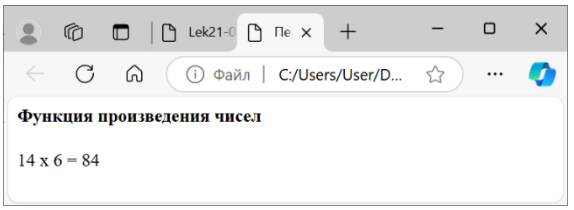
Ал көп жолдық көлемді комментарийлер /* таңбаларынан басталып, */ таңбаларымен аяқталады.

Құжат тақырыбына скрипт қосу

Бұл бөлімде скриптті құжат тақырыбына енгізу мысалы келтірілген. Әдетте, функциялары бар скрипт веб-құжаттың <head> элементі ішінде орналасады, ал функциялар бар болса, олар <body> элементінің ішінде шақырылады. Функциялар (пайдаланушы функциялары немесе әдістер) – логикалық жағынан тәуелсіз программалар бөліктері – JavaScript тілінің негізгі бір элементтері болып табылады. Параметрлері бар функцияны пайдалану мысалы (Lek21-1.htm):

```
<html> <head> Алғашқы функция
<script type="text/javascript">
  function product(a,b)
   {  return a*b;  }
       /* Бұл мысалда көпжолдық комментарий және сандар көбейтіндісін
    есептейтін функция жұмысы көрсетілген  */
</script> </head>

<body>
<h4>Сандар көбейтіндісінің функциясы
</h4>
14 x 6 =
```



```
<script type="text/javascript">
    document.write (product(14,6));
</script></body></html>
```

Шартты оператор

Шартты оператор тармақталған командаларды ұйымдастыру үшін пайдаланылады. Шартты операторлар төмендегідей командалар түрінде іске асырылған:

- **if** <операторлар>;
- **if...else** <операторлар>;
- if...else if...else <операторлар>;
- **switch** <операторлар>.

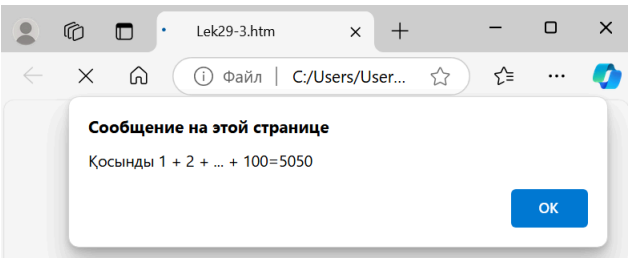
if логикалық командасы салыстырудың қалай аяқталатынын тексереді. Егер **if** командасын орындау нәтижесі **true** болса, онда операциялардың бір блогы орындалатын болады, ал егер нәтижесі **false** болса, онда операциялардың басқа блогы (тізбегі) орындалатын болады. Мысалы:

```
<html><head> Шартты оператор </head>
<body>
   <script type="text/javascript">
        var  r = Math.random();
        if (r>0.5)
         {  document.write("Кездейсоқ сандар алу");  }
        else
         {  document.write("Math объектісі бар программа мысалы"); }
  </script>
</body> </html>
```

Бұл мысалда кездейсоқ сандар туындататын Math.random() объектісі пайдаланылды. Егер ол 0,5-тен артық кездейсоқ сан шығарса, онда экранда Кездейсоқ сандар алу деген жазу пайда болады, егер кездейсоқ сан 0,5-тен кем немесе тең болса, онда экранға Math объектісі бар программа мысалы деген жазба шығады.

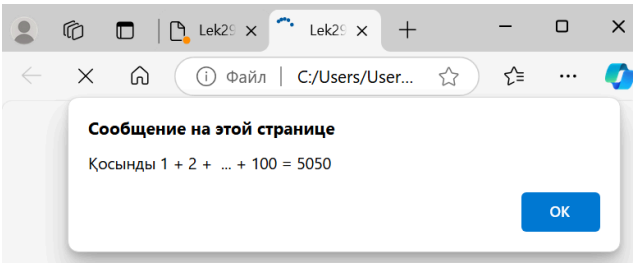
29-3 `while` циклін пайдалану мысалы:

```
<html> <head> while циклдік операторы </head>
<body>
   <script type="text/javascript">
      var i = 1;
      var sum = 0;
      while (i <= 100)
      {   sum += i;
          i ++;
       }
       alert("Қосынды 1 + 2 + ... + 100=" + sum);
   </script>
</body> </html>
```



29-4 `for` циклін пайдалану мысалы:

```
<html><head> for циклдік операторы </head>
<body>
   <script type="text/javascript">
      var i = 1;
      var sum = 0;
      for(i=1; i<=100; i ++)   sum += i;
```



```
    alert ("Қосынды 1 + 2 +  ... + 100 = " + sum);
  </script>
</body> </html>
```

Бұл мысалдың да нәтижесі алдыңғы суреттегі мысалдағыдай болады. Цикл жұмысы келесі түрде атқарылады: циклдегі алғашқы теңдік жүзеге асырылып (мысалдағы i=1; командасы), содан соң келесі әрекеттер орындалады:

- шартты тексеру (мысалдағы i<=100);
- цикл тұлғасын орындау (мысалдағы sum += i);
- қадамды көрсету командасын орындау, (мысалдағы i++).

Егер шарт бірден жалған болса, while командасындағы сияқты цикл операторлары бір де бір рет орындалмауы да мүмкін. Мұндайда қадам енгізу командасы орындалмайды. Ал циклдің алғашқы командасы әрқашанда кем дегенде бір рет орындалады.

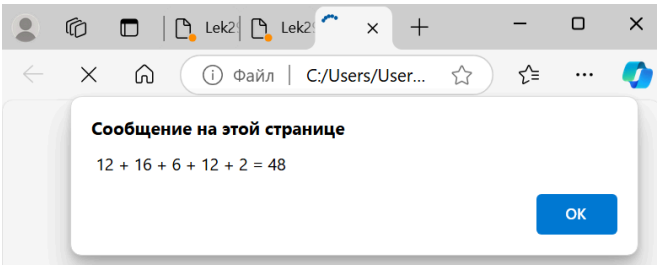
Break және continue командалары

Бұл командалар цикл командаларының орындалу ретін өзгерту үшін қолданылады.

continue командасы циклдің онан кейінгі тұрған барлық командаларын аттап өтіп, цикл параметрінің келесі мәніне көшіреді. Ал break командасы жалпы цикл орындалуын аяқтап, одан кейінгі келесі командаларға көшіреді.

29-5 *continue* (*while) мысалы.* [1, 20] аралығынан кездейсоқ түрде алынған 5 жұп санның қосындысын табу керек.

```
<html><head> while циклiндегi continue операторы </head>
<body>
  <script type="text/javascript">
    var  len = 5;    // Сандардың қанша екендігі.
    var  a = 1;      // Аралықтың сол жақ шекарасы.
    var  b = 20;     // Аралықтың оң жақ шекарасы.
    var  sum = 0;  // Қосқыш.
    var  counter = 0;  // сандардың санауышы.
    var  number;      // Кездейсоқ сан.
    var  str = " ";      // Экранға шығаруға арналған қатар.
    while (counter < len)
    {  number = Math.floor(a + (b-a+1)*Math.random());  // кездейсоқ сан алу
       if (number % 2)  continue;
       sum += number;  str += number;
       if (counter < len-1)  str += " + ";
       else  str += " = ";
       counter++;
    }
    str += sum;  alert(str);
  </script>
</body> </html>
```

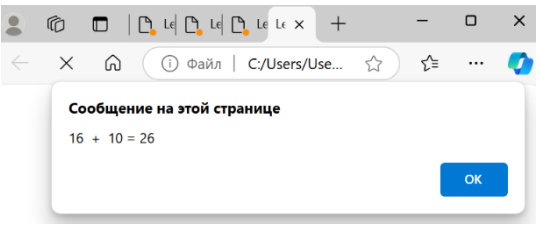


Math.random() стандартты функциясы [0,1] аралығынан кез келген кездейсоқ сан береді. Math.floor(num) стандартты функциясы аргументке тең немесе одан кіші бүтін сан мәнін береді.

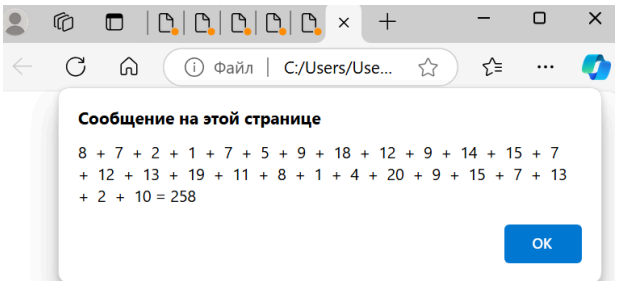
29-6 *break (for) мысалы.* Керекті сандар [1,20] аралығынан кездейсоқ түрде алынып отырады. Осы сандардың кезекті келесісі 10-ға тең болғанша, олардың қосындысын табу керек.

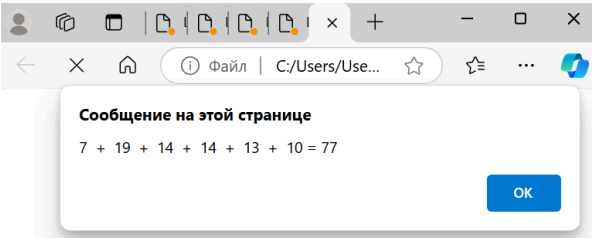
```
...
var a = 1;            // аралықтың сол жақ шекарасы.
var b = 20;           // аралықтың оң жақ шекарасы.
var special = 10;  // кездейсоқ санның өзекті мәні
var sum = 0;       // қосынды, басында 0-ге тең
```

```
var number;      // кездейсоқ сан айнымалысы
var str = "";    // экранға мәні шығарылатын айнымалы.
for ( ; ; )      // шексіз цикл.
{  number = Math.floor(a + (b-a+1)*Math.random( ));
   sum += number;   str += number ;
   if (number == special)  break;
   str  +=  "  +  ";
}
str += " = " + sum;   alert(str);
...
```



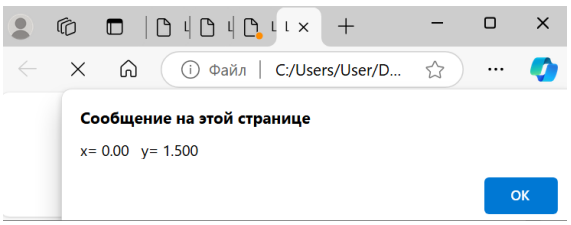
Бұл мысалда да сандар кездейсоқ алынатын болғандықтан нәтижесі әртүрлі бола береді.

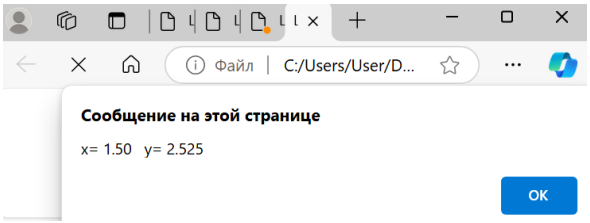


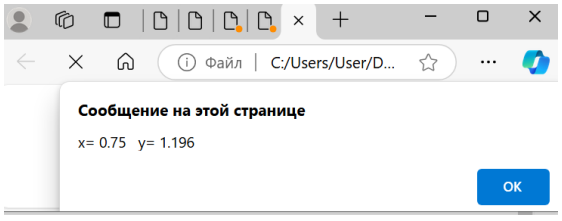


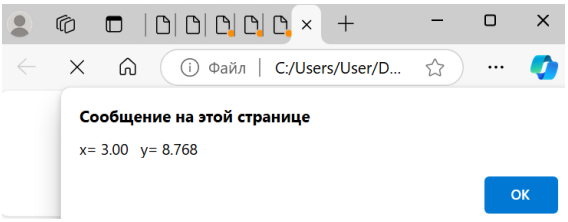
29-7 *мысал*. $y = x^2 - \sqrt{x} + 1.5$ функциясы берілген. Мұндағы *x* аргументі 0-ден 3-ке дейін, қадамы 0,75 болып өзгергенде, *y* функциясының мәндерін табу керек.

```
...
var  x0 = 0;     // х-тің бастапқы мәні
var  xk = 3;     // х-тің соңғы мәні
var  dx = 0.5;   // х-тің өзгеру қадамы
var  x=x0;       // х-ке алғашқы мәнді меншіктеу
var y;           // y айнымалысын сипаттау
while (x =< xk)
{  y = x*x - Math.sqrt(x)+1.5;
   alert("x= " + x.toFixed(2) + " y= " + y.toFixed(3));
   x+=dx ;
} ...
```





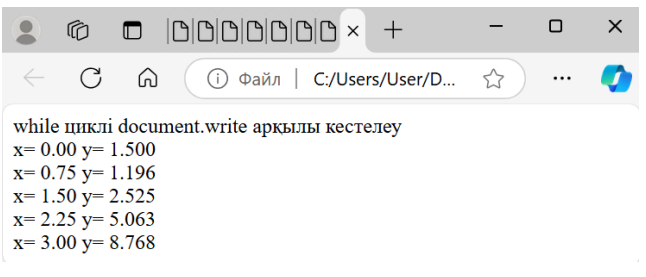




Осы мысалдың document.write командасының көмегімен орындалуының нәтижесі төменде көрсетілген.

Енді осы функцияның мәнін, х-тің мәні 0-ден 3-ке дейін, 0,75 қадаммен өзгерген кезде анықтау төмендегідей болады (29-8 мысал).

```
...
var  x0 = 0;     // х-тің бастапқы мәні
var  xk = 3;     // х-тің соңғы мәні
```



```
var  dx = 0.75;   // x-тің өзгеру қадамы
var  x = x0;      // x-ке алғашқы мәнді меншіктеу
var y;            // y-ті сипаттау
while (x =< xk)
 {  y = x*x - Math.sqrt(x) +1.5;
    document.write ("x= " + x.toFixed(2) + " y= " + y.toFixed(3));
    x+=dx ;
 }
...
```