

ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО

Вовед во скриптиње и Javascript

Скриптирање

Скриптите (програми напишани во скриптирачки јазик)
се користат на
клиентската страна во рамки на пребарувачот,
и на **серверска страна**,
како програми што се извршуваат на серверот и генерираат
резултати што се испраќаат на пребарувачот.

Скриптирачки јазици

- едноставни (вообичаено) интерпретирани јазици
- Видови
 - ☐ Клиентска страна
 - VBScript - скриптите се извршуваат (само) на IE
 - JavaScript - скриптите може да се извршуваат на поголем број прелистувачи
 - ☐ Серверска страна
 - Active Server Pages (ASPs)
 - PERL

Скриптирање и скриптирачки јазици

- ги разгледуваме само скриптите што се извршуваат на клиентската страна
- скриптите се вгнездуваат во веб-страниците
- се користат за додавање на интерактивност на веб-страниците
 - извршување и контрола на HTML форми
 - валидација на кориснички внесени податоци во HTML формите
 - креирање напредни својства на веб страниците, вклучувајќи:
 - отворање на нов пребарувачки прозорец да се прикаже алтернативна веб-страница,
 - креирање на колачиња („cookies”) што сместуваат податоци на корисничкиот компјутер за неговите (нејзините) активности при пребарување на веб-страницата ...



Зошто JavaScript?

- Скриптен јазик со кој се зголемува функционалноста на веб-страниците
- Денес е еден од најшироко распространетите програмски јазици
 - го подржуваат сите (?) пребарувачи
 - не е напoлно платформски независен
 - различните пребарувачи се однесуваат различно
 - Се користи дури и на мобилните уреди
 - Постојат и проекти за Arduino, Tessel, Es- prduino, и NodeBots кои предвидуваат дека јазикот ќе се користи и во ваквите вгнездени системи и роботиката
- претставува програмски јазик со објектно-ориентирани карактеристики
 - Објектно-базиран (верзиите пред JavaScript 2.0), настански воден јазик
 - комплетен, комплексен јазик
- едноставен е за учење
- достапни се голем број скрипти без надомест!!!!

Зошто JavaScript?

- Креирањето на програма во JavaScript е едноставно и се сведува на едитирање на текстуална датотека која потоа се отвора во прелистувач
- Нема потреба од комплексна развојна околина што треба да се инсталира ниту да се совлада
- Основната синтакса е позната на секој програмер што имал искуство со синтаксата на програмскиот јазик C.

Историја

- На почеток овозможил проверка на точноста на податоците на клиентската страна без истите да се праќаат на сервер што за тогашните брзини на интернетот била значајна придобивка (28.8 kbps)
- Brendan Eich, кој работел за Netscape во тоа време, започнал развој на скриптирачки јазик кој прво го нарекол Mocha, а потоа LiveScript, за верзијата Netscape Navigator 2 во 1995, со намера да се користи и на клиентска и на серверска страна
- Netscape се здружува со Sun Microsystems да ја заврши имплементацијата на LiveScript
- Пред да излезе официјално Netscape Navigator 2, Netscape го сменил името на LiveScript во JavaScript.
- Поради популарноста на JavaScript 1.0, Netscape ја издава верзијата 1.1 за Netscape Navigator 3, со што Netscape се позиционира како водечка компанија на пазарот.
- Во тоа време, Microsoft вложува средства за неговиот прелистувач Internet Explorer, и набргу по излегувањето на Netscape Navigator 3, Microsoft во воведува Internet Explorer 3 со имплементација на JavaScript наречена JScript (се одбегнуваат прашања поврзани со лиценца со Netscape). Ова е значаен чекор за Microsoft во реалноста на прелистувачите, но и значаен чекор кон развој на JavaScript.

Стандардизација

- Имплементацијата на Microsoft на JavaScript значи дека има две различни верзии на JavaScript: JavaScript во Netscape Navigator и JScript во Internet Explorer.
- За разлика од C и другите програмски јазици, JavaScript немала стандард по кој би се развивала. Со развојот на индустријата одлучено е јазикот да се стандардизира.
- Во 1997, JavaScript 1.1 е предложена на European Computer Manufacturers Association (Ecma).
 - На техничкиот комитет #39 (TC39) му била доделена задачата “да ја стандардизира синтаксата и семантиката на скриптирачкиот јазик со генерална намена, достапен за многу платформи и неутрален во однос на производителите” (www.ecma-international.org/memento/TC39.htm).
 - Сочинет од програмери на Netscape, Sun, Microsoft, Borland, NOMBAS, и други компании, TC39 го дефинира ECMA-262, стандардот за новиот скриптен јазик наречен ECMAScript (“ek-ma-script”).
- Следната година, International Organization for Standardization and International Electrotechnical Commission (ISO/IEC) го прифаќа ECMAScript како стандард (ISO/IEC-16262).
 - Од тогаш, компаниите со различно ниво на успех го користат ECMAScript како основа на нивните JavaScript имплементации.

Изглед на едноставна скрипта напишана во JavaScript?

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Barely a script at all</title>
</head>
<body bgcolor="#FFFFFF">
<h1>
  <script language="Javascript" type="text/javascript">
    document.write("Zdravo na site!")
  </script>
</h1>
</body>
</html>
```

ОСНОВНИ КОНЦЕПТИ

- JavaScript ретко се користи за пишување на комплетни програми
- скрипта напишана во JavaScript претставува ASCII текст
 - програмите се интерпретираат
 - интерпретерот ги чита и извршува скриптите
- дефинирани се објекти и едноставни податочни видови
- имплементирани се концептите на настани и исклучоци
- (?) наредбите завршуваат со интерпункцискиот знак ;
- за пристап до елементите на објектите се користи синтаксата на точка
 - пример:
document.write("Hello World")
- дефинирани се нови наредби: with и for

Основни концепти

- Кога се извршуваат скриптите?
 - дел од програмскиот код се извршува при првото полнење на страницата
 - дел од програмскиот код се извршува кога корисникот ќе предизвика настан поврзан со акциите од кодот



Користење на JavaScript во пребарувач

- JavaScript кодот се вметнува во рамките на `<script>` тагот:

```
<script type="text/javascript">  
    //Javascript kod  
</script>
```

```
<html>  
<head>  
    <script type = "text/javascript">  
        <!--  
            document.write("<h1 style=\"color: red\">");  
            document.write(" Hello world");  
            document.write("</h1>");  
        // -->  
    </script>  
</head>  
<body> <p>And here is more text</p> </body>  
</html>
```



Внимание

- Кога се вметнува JavaScript кодот треба да се внимава дека “</script>” не може да се најде каде и да е во кодот.
- На пример, следниот код предизвикува грешка при полнење во прегледувачот:

```
<script type="text/javascript">
function sayScript(){
    alert("</script>"); }
</script>
```

- Бидејќи и вгнездените скрипти се парсираат, прелистувачот го гледат текстот “</script>” како ознака за затворање на </script>.
- Овој проблем може да се елиминира со користење на знакот “/” character, како во примерот:

```
<script type="text/javascript">
function sayScript(){
    alert("<\script>"); }
</script>
```



Каде се вметнува JavaScript

- може да се вметне во
 - заглавјето (<head>)
 - телото (<body>) на еден HTML документ
- функциите треба да бидат дефинирани во заглавјето (<head>)
 - тоа обезбедува функцијата да биде наполнета пред да биде употребена

Каде се вметнува JavaScript

- може да се смести во одделна .js датотека
`<script src="myJavaScriptFile.js"></script>`
 - надворешната .js датотека овозможува користење на иста скрипта на повеќе HTML страници
 - надворешната .js датотека не може да содржи `<script>` таг
- пример:
`<script src="myjavascript.js"
 language="JavaScript1.2"
 type="text/javascript">
</script>`



Работа со пребарувачи што не поддржуваат скриптирачки јазици

- некои стари пребарувачи не ги препознаваат script таговите
- овие пребарувачи ќе ги игнорираат script таговите но, ќе го прикажат кодот на вметнатиот JavaScript
- да се овозможи старите пребарувачи да го игнорираат целиот код, се користат тагови за коментари од HTML што ја кријат скриптата од пребарувачот и тој ја игнорира
- синтакса
 - <!--
script here
// -->
 - ☐ <!-- воведува HTML коментар (почеток)
 - ☐ за JavaScript да ја игнорира ознаката за крај на HTML коментарот (-->), се користи JavaScript коментар (//), кој важи до крајот на линијата



пример

```
<html >
<head>
  <title>My first script</title>
</head>
<body bgcolor="#FFFFFF">
<h1>
  <script language="Javascript" type="text/javascript">
    <!-- Hide script from old browsers

      document.write("Hello, world!")

  // End hiding script from old browsers -->
  </script>
</h1>
</body>
</html>
```

«NOSCRIPT» таг

- Друг начин да се реши проблемот со скрипти што се извршуваат на прелистувачи што не подржуваат JavaScript е да се користи `<noscript>` елементот.

```
<!DOCTYPE html> <html>
```

```
<head>
```

```
<title>Example HTML Page</title>
```

```
<script type="text/javascript" defer="defer" src="example1.js"></script>
```

```
<script type="text/javascript" defer="defer" src="example2.js"></script>
```

```
</head> <body>
```

```
<noscript>
```

```
<p>This page requires a JavaScript-enabled browser.</p>
```

```
</noscript>
```

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

Едноставни податочни видови

- JavaScript има три „едноставни“ податочни видови:
 - ☐ нумерички вредности (number),
 - ☐ текстуални низи (string), и
 - ☐ логички вредности (boolean)
 - ☐ се останато е објект
- Нумеричките вредности секогаш се сместуваат во формат со подвижна точка
 - ☐ хексадецималните броеви започнуваат со 0x
 - ☐ некои платформи бројот 0123 го сметаат како октален, другите го сметаат како децимален

Едноставни податочни видови

- Текстуалните низи може да се ограничени во единечни или двојни наводници
 - тие може да содржат и контролни знаци - `\n` (newline), `\"` (double quote), итн.
 - Пример:
`strFirst = "John";`
`strLast = "Kennedy";`
`strFull = strFirst + "F." + strLast;`
- Логичките податоци може да имаат вредност `true` или `false`
 - `0`, `"0"`, празни текстуални низи, `undefined`, `null`, и `NaN` се логички `false`
 - сите останати вредности се логички `true`
- Недефинирана податочна вредност
 - вредност на неиницијализирана променлива
 - вредност **`undefined`**

Предефинирани објекти

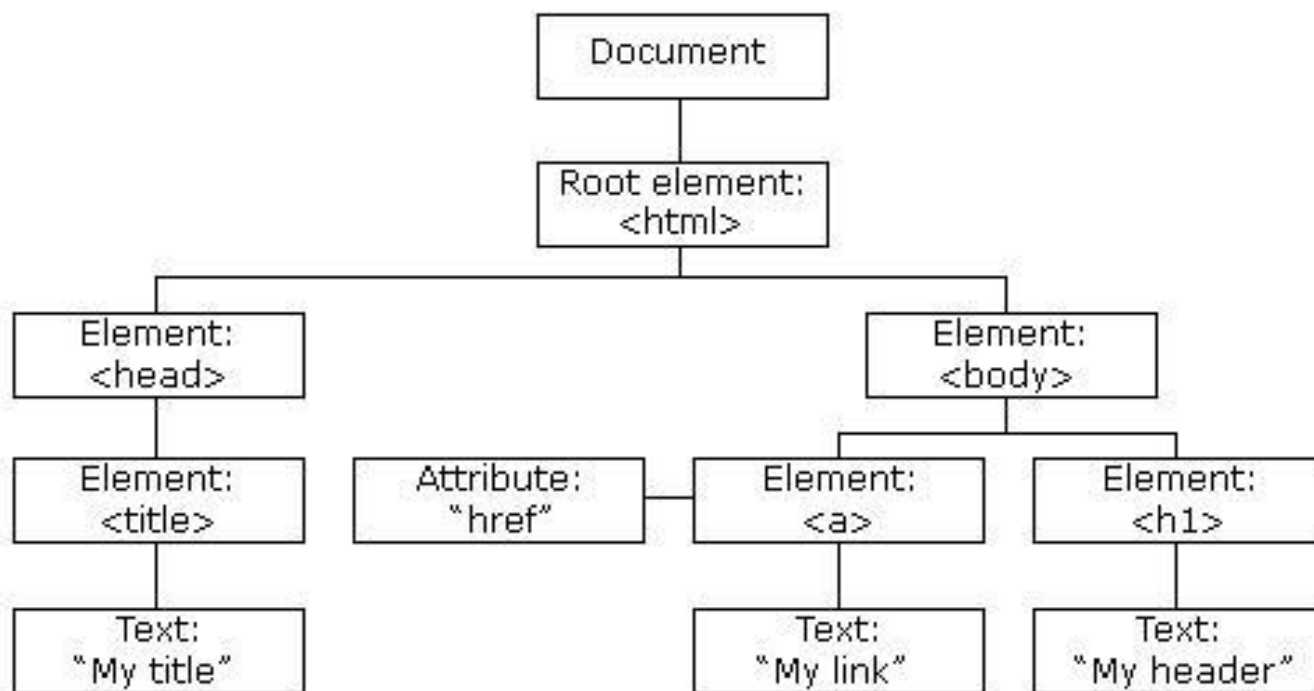
- Во JavaScript се дефинирани следните објекти (корисникот може да ги користи без да ги дефинира)
 - ☐ document
 - ☐ navigator
 - ☐ screen
 - ☐ window

HTML DOM

- Означува **Document Object Model** за HTML
- W3C стандард
- Платформски и јазички е независен
- Дефинира стандард
 - Множество објекти за HTML
 - Пристап и манипулација со HTML

HTML DOM

- Ги дефинира HTML документите како структура на дрво со елементи



HTML DOM

- Дозволува некој скриптирачки јазик, како на пример JavaScript
 - да пристапи и ажурира различни делови од HTML документ
 - да креира динамички веб страници на пример, напиши “Hello World!” на веб-страница

```
document.write("<h1>Hello World!</h1>")
```

Објект
document

Метод во
објектот
document

Дијалог прозорци

□ Прозорец за сигнализација

```
alert("Input field is empty.");
```



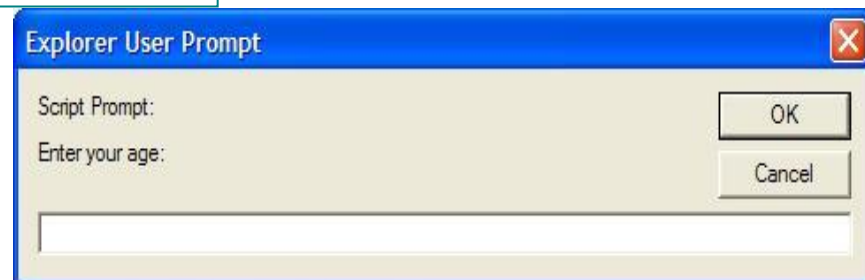
□ Прозорец за потврда

```
answer = confirm("Do you want to quit?");
```



□ „Prompt” прозорец

```
age = prompt("Enter your age:", "");
```



Променливи

- променливите се декларираат со наредбата `var`
 - зборот `var` е опционален (добар програмерски стил е негова употреба)
 - податочниот вид на променливите не мора да биде деклариран и се определува во време на извршување на скриптата
 - променливите може да содржат вредност од кој и да е вид
 - `Var pi;`
 - `Var ime: Integer;` `//(Javascript 2.0)`
- Променливите може да се иницијализираат со користење на знакот `=`.
 - `var pi = 3.1416, x, y, name = "Dr. Dave" ;`
 - имињата на променливите мора да започнуваат со буква или долна цртичка (`_`)
 - се прави разлика меѓу големи и мали букви во имињата на променливите

Оператори

- операторите се користат за процесирање на вредностите

- видови оператори

- ☐ аритметички оператори:

+ - * / % ++ --

- ☐ релациски оператори:

< <= == != >= >

- ☐ логички оператори:

&& || !

- ☐ битови оператори:

& | ^ ~ << >> >>>

- ☐ оператори за доделување:

+= -= *= /= %= <<= >>= >>>= &=

^= |=

- ☐ оператори за текстуални низи:

+

- условен оператор:

condition ? value_if_true : value_if_false

- Дополнителни оператори:

new typeof void delete



Оператори

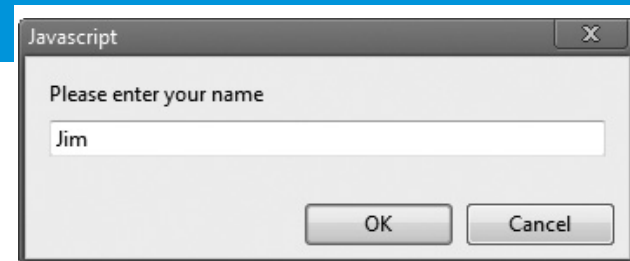
- специјални релациски операции за тестирање еднаквост:
 - == и != се обидуваат да ги конвертираат нивните операнди во ист вид пред да го извршат тестот
 - === и !== претпоставуваат дека нивните операнди се нееднакви ако истите се од различен вид

Оператори

- Системот ќе се обиде да ги кастира вредностите да може да ја изврши операцијата
 - се може да се претвори во текстуална низа
 - некои текстуални низи може да се претворат во броеви
 - дополнување за логичките вредности
 - во нумерички контекст, **true** се кастира во 1 и **false** се кастира во 0
 - во логички контекст, дефинираните вредности се кастираат во **true**, а недефинираните (**undefined**) се кастираат во **false**
 - во контекст на текстуални низи, **true** се кастира во “true” и **false** се кастира во “false”
 - ништо освен функција не може да се кастира во функција
- Операторите се извршуваат во контекст на операндите
 - $a * b$ => обезбедува нумерички контекст
 - $e(x)$ => контекст на функција (за e)
 - $a + b$ => неопределено (текстуална низа или број)



Коментари



■ Коментарите се идентични како и коментарите во С или во Java:

- од // до крајот на линијата
- меѓу /* и */



■ Пример

```
<!DOCTYPE html> <!-- Fig. 6.5: welcome4.html -->
<!-- Prompt box used on a welcome screen --> <html>
<head>
<meta charset = "utf-8">
<title>Using Prompt and Alert Boxes</title> <script type = "text/javascript">
<!--
var name; // string entered by the user
// read the name from the prompt box as a string
name = window.prompt( "Please enter your name" );
document.writeln( "<h1>Hello " + name + ", welcome to JavaScript programming!</h1>" );
// -->
</script> </head><body></body> </html>
```

Наредби

- наредба за доделување:

```
greeting = "Hello, " + name;
```

```
nNum -= 3; nNum = nNum - 3;
```

```
nNum *= 3; nNum = nNum * 3;
```

```
nNum /= 3; nNum = nNum / 3;
```

```
nNum %= 3; nNum = nNum % 3;
```

- блок од наредби:

```
{ statement; ...; statement }
```

- празна наредба: `;;` или `{ }`

Сложени наредби

- Наредби за избор
 - If, switch
- Наредби за повторување
 - for, do-while, while

If Наредби

■ Синтакса

`if (condition) statement;`

`if (condition) statement; else statement;`

`if (condition) {statement;}`

`if (condition) {statement;} else {statement;}`

```
if (i > 25) {  
    alert("Greater than 25.");  
} else if (i < 0) { alert("Less than 0.");  
} else {  
    alert("Between 0 and 25, inclusive.");  
}
```



Проверка дали е овозможена Java

```
<html>
<head>
  <title>Do you have Java?</title>
</head>
<body bgcolor="#FFFFFF">
<h1>
<script language="Javascript" type="text/javascript">
  <!-- Hide script from old browsers
  document.write("Java is ")
  if (!navigator.javaEnabled()) {
    document.write("not ")
  }
  document.write("enabled.")
  // End hiding script from old browsers -->
</script>
<noscript> JavaScript is not enabled. </noscript>
</h1>
</body>
</html>
```

Java-enabled.html

Switch Наредба

■ switch наредба:

```
switch (expression){
    case label :    statement; break;
    case label :    statement; break;
    ...
    default : statement;
}
```

```
switch ( k ) {
    case 0:    weekDay = "Monday";    break;
    case 1:    weekDay = "Tuesday";    break;
    case 2:    weekDay = "Wednesday"; break;
    case 3:    weekDay = "Thursday";   break;
    case 4:    weekDay = "Friday";     break;
    default: break;
}
```

Пример

```
<html >
<head>
  <title>Switch/Case handling</title>
  <script language="Javascript" type="text/javascript">
    <!-- Hide script from older browsers
    function saySomething(buttonName) {
      switch(buttonName.value) {
        case "Lincoln": alert("Four score and seven years ago..."); break ;
        case "Kennedy": alert("Ask not what your country can do for you..."); break;
        case "Nixon":   alert("I am not a crook!"); break;
        default:
      }
    }
    // End hiding script from older browsers -->
  </script>
</head>
<body bgcolor="#FFFFFF">
<h2>Famous Presidential Quotes</h2>
<form action="#">
  <input type="button" value="Lincoln" onclick="saySomething(this)" />
  <input type="button" value="Kennedy" onclick="saySomething(this)" />
  <input type="button" value="Nixon" onclick="saySomething(this)" />
</form>
</body>
</html>
```

Switch.html

Наредби за повторување

■ СИНТАКСА:

while (condition) statement;

```
var i = 0;  
while (i < 10) { i += 2; }
```

do statement while (condition);

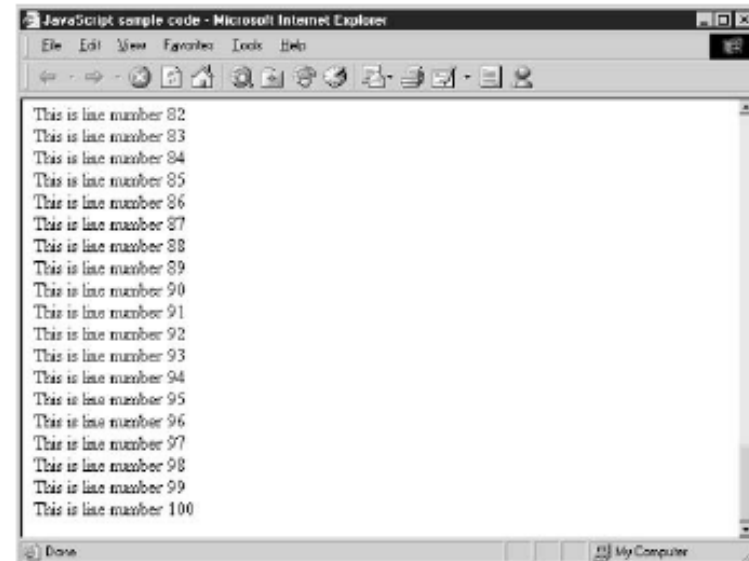
```
var i = 0;  
do {  
    i += 2;  
} while (i < 10);
```

for (initialization; condition; increment) statement;

```
var count = 10;  
for (var i=0; i < count; i++){ alert(i); }
```

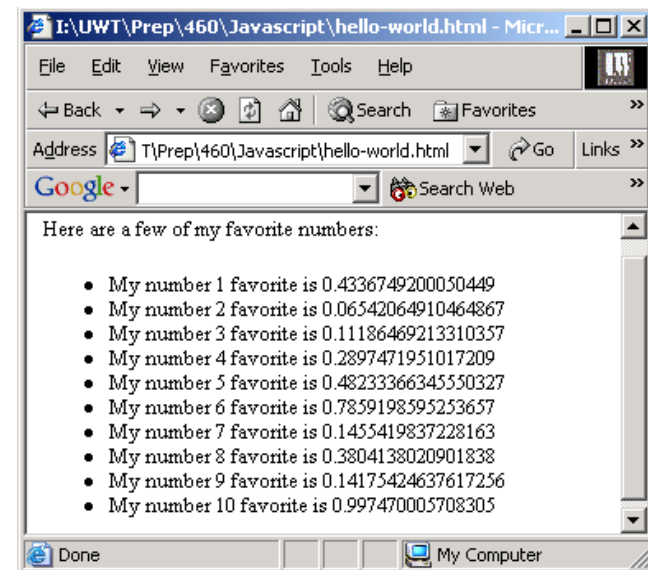
Пример за while

```
<html>
<head>
<title>JavaScript sample code</title>
</head>
<body>
<h1>My Sample Code</h1>
<script language="JavaScript" type="text/javascript">
<!-- // Begin
    var counter = 1;
    while (counter < 101) {
        document.write ("This is line number " + counter + "\n");
        counter++;
    }
// End -->
</script>
</body>
</html>
```



Пример for наредба

```
<html> <head>
  <script type = "text/javascript">
    document.write("Here are a few of my favorite numbers:");
    document.write("<ul>");
    for (i=0; i<10; i++) { n = Math.random(); j = i+1;
      document.write("<li>My number " + j + " favorite is " + n + "</li>");
    }
    document.write("</ul>");
  </script>
</head>
<body></body>
</html>
```





do-while пример

```
<html>
<head>
  <title>Force a response</title>
  <script language="Javascript" type="text/javascript">
    <!-- Hide script from older browsers
    do {
      ans = prompt("Tell me your name","")

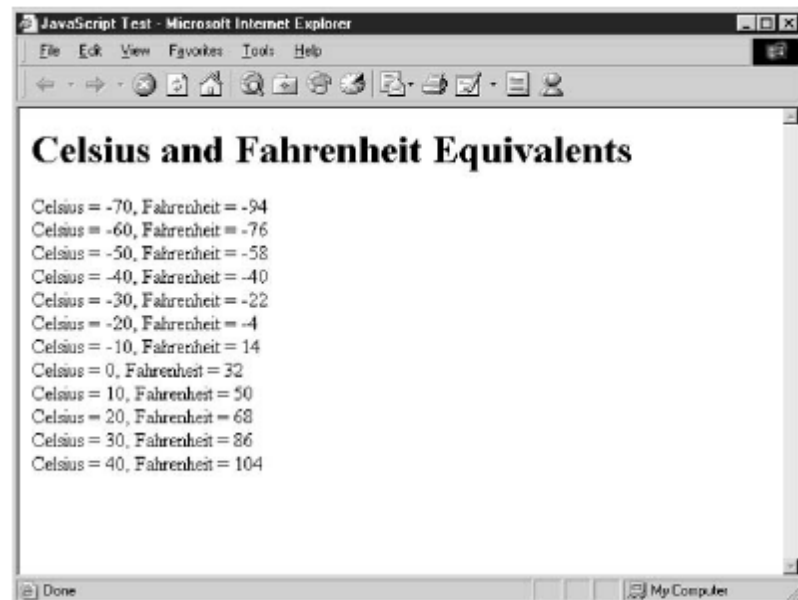
    }
    while (!ans)
    alert("Hello, " + ans)
  // End hiding script from older browsers -->
</script>
</head>
<body bgcolor="#FFFFFF">
</body>
</html>
```

■ Do-while.html

Наредби за скок - break;

```
var Fahrenheit, Celsius;

for (Celsius = -70; Celsius <= 70; Celsius += 10) {
    Fahrenheit = (Celsius * 9/5) + 32;
    document.write ("Celsius = " + Celsius);
    document.write (" , Fahrenheit = " + Fahrenheit + "<br>");
    if (Fahrenheit > 100) { // It's getting hot in here
        break;
    }
}
```



Наредби за скок

■ continue;

// broevi delivi so 23?

var iNumber;

for (iNumber = 1; iNumber <= 1000; iNumber++) {

if (iNumber % 23 > 0) { // ne e deliv so 23

continue;

}

document.write("" + iNumber + "");

document.write(" is evenly divisible by 23!
");

}

