



برنامه نویسی سمت کلاینت JavaScript

M.Madadyar

<http://www.Madadyar.ir>



What is JavaScript?

- JavaScript was designed to add **interactivity** to HTML pages
- A JavaScript consists of lines of **executable** computer code
- A JavaScript is usually embedded directly into HTML pages



What can JavaScript do?

- JavaScript gives HTML designers a programming tool
- JavaScript can put dynamic text into an HTML page
- JavaScript can react to **events**
- JavaScript can read and write HTML elements

JavaScript Sample

```
<html>
```

```
<body>
```

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

محل قرارگیری کدهای جاوا اسکریپت

```
</script>
```

```
</body>
```

```
</html>
```

Where to put the JavaScript in my html code?

- Inside both the **head** or **body**

```
<html>
```

```
  <head>
```

```
    <script type="text/javascript" src=" آدرس فایل js " />
```

```
  </head>
```

```
</html>
```



JavaScript Concepts

1. Variables and Data Types
2. Assignment
3. Input / Output
4. Arithmetic and Relational operators
5. If / Else
6. If / Else if
7. Loops – For
8. Loops – Do
9. Loops – While
10. Arrays
11. Sample Problems / Questions & Answers

Variables

- Purpose is to temporarily store information.
- Think of them as containers of information, the variable name being the label of the container, so that we can refer back to it.

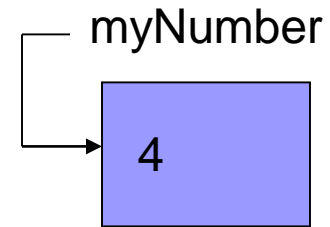


Diagram of an arbitrary variable containing the value 4:

Variables

- Use the **var** keyword to **declare** variables.

```
var myNumber;
```

- Must start with a letter, followed by any sequence of letters and numbers.
- No spaces allowed. The following is wrong:

```
var my Number;
```

- **Initialization** – giving an initial value to a variable:

```
var myNumber = 4;
```

- Must initialize it before you use it. Otherwise your programs have errors (e.g. undefined, NaN, etc.)

Variables – Data types

■ Data Types

□ Numbers

- 1
- 3452
- -33.45

□ Strings

- "a"
- "hello"
- "afg23sd"
- "234"

□ Booleans

- true
- false

Assignment

- Operation which assigns a value to a variable.
- Assignment Operator: =
 - Not to be confused with equality: ==
- Assignment takes the value appearing on the right of the = operator
 - `VariableReceivingNewValue = expressionOrNewValue;`
- Example:
`myNumber = 4;`

Input / Output

- Input – **prompt()**

```
var name = prompt("What is your name?");
```

- Output – **alert()**


```
alert("Your name is " + name);
```

- Output - **document.write()**

```
document.write("Your name is " + name);
```

Prompt()

- Prompt returns a string!
- If you want to get the number use the following function:
 - `var num=parseInt(a string);`
 - `var fnum=parseFloat(a string);`
- Example:
 - `var input=prompt("Enter number:");`
 - `var number=parseInt(input);`

- 
- `<script type="text/javascript">`
 - `document.write("<h1>This is a header</h1>");`
 - `document.write("<p>This is a paragraph</p>");`
 - `document.write("<p style='color:red'>test ! </p>");`
 - `</script>`

Operators

■ Arithmetic Operators → result is a Number

- ☐ Addition +
- ☐ Subtraction −
- ☐ Multiplication *
- ☐ Division /

■ Relational Operators → result is a Boolean

- ☐ Greater than >
- ☐ Greater than or equal >=
- ☐ Less than <
- ☐ Less than or equal <=
- ☐ Equal ==
- ☐ Not equal !=
- ☐ Not !
- ☐ And &&
- ☐ OR ||

if / else conditional statement

- Executes a set of statements if the condition is true; it may optionally execute another set of statements if the condition is false.

```
if (<condition to check>)  
{  
    <statement(s) to execute if condition  
    is true>  
}  
else // this part is optional  
{  
    <statement(s) to execute if condition  
    is false>  
}
```

if / else example

- Display the appropriate message if the password is “123”

```
var password = prompt("Enter password", "");  
if (password == "123")  
{  
    alert("Access Granted!");  
}  
else  
{  
    alert("Access Denied!");  
}
```


if / else if

- A form of a conditional statement - checks multiple sets of conditions.
- Only the statements in the first true condition found will be executed.

```
if (<condition1 to check>)
{
    <statement(s) to execute if condition 1 is true>
}
else if (<condition2 to check>)
{
    <statement(s) to execute if condition 1 is false and
    condition 2 is true>
}
else if (<condition3 to check>)
{
    <statement(s) to execute if conditions 1 and 2 are false
    and condition 3 is true>
}
.
.
.
```

if / else if Example

- Checks the average grade of a student to determine their letter grade:

```
if (avg >= 90)
{
    document.write("The grade obtained is A.");
}
else if (avg >= 75 && avg < 90)
{
    document.write("The grade obtained is B.");
}
else if (avg >= 65 && avg < 75)
{
    document.write("The grade obtained is C.");
}
else if (avg >= 50 && avg < 65)
{
    document.write("The grade obtained is D.");
}
else
{
    document.write("The grade obtained is F.");
}
```

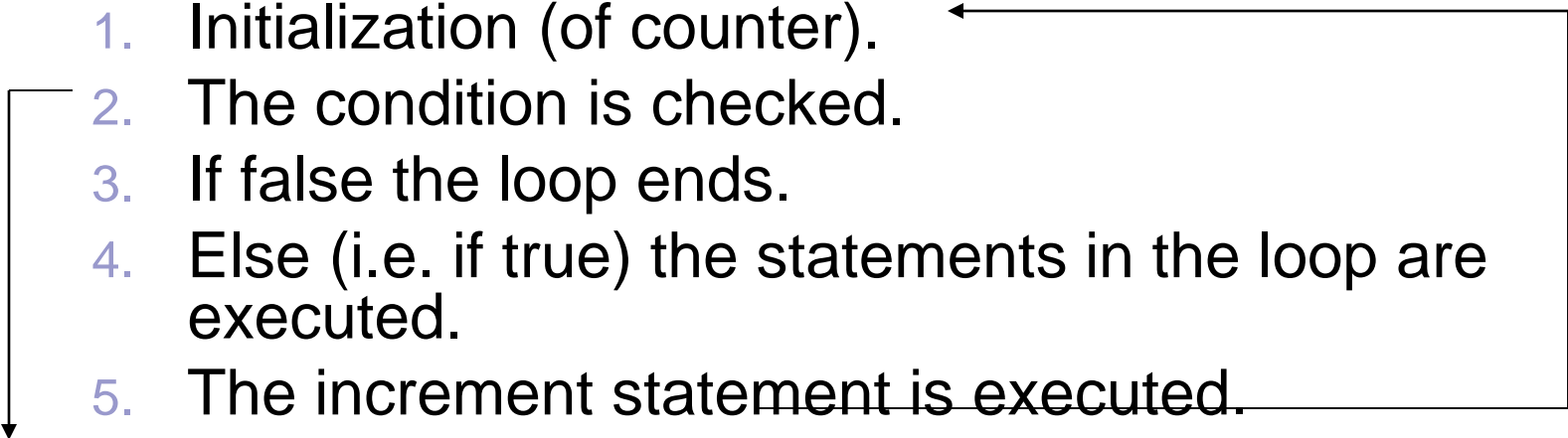
Loops – for loop

- Loops repeat execution a portion of a Java script based on a given condition.

```
for (<initialization>; <condition/continuation>;  
    <next iteration /increment>)  
{  
    <set of commands to be repeated>  
}
```

- Steps

1. Initialization (of counter).
2. The condition is checked.
3. If false the loop ends.
4. Else (i.e. if true) the statements in the loop are executed.
5. The increment statement is executed.
6. Go back to step 2.



For loop - example

- Display “Hello” 3 times:

```
for (var i=0; i<3; i++)  
{  
    document.write("<br>Hello!");  
}
```

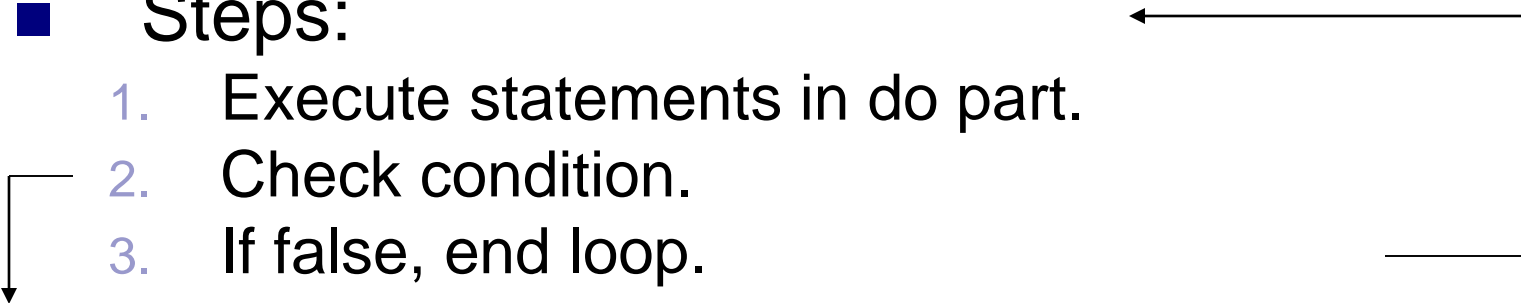
Do loop

- This loop first executes the statements in the loop, before checking the loop condition.
- Executes **UNTIL** loop condition is false.
- Do loops execute at least ONCE.

```
do
{
    <statement(s) to repeat while loop
    condition is true>
}while(<loop condition>);
```

■ Steps:

1. Execute statements in do part.
2. Check condition.
3. If false, end loop.
4. Else (i.e. If condition is true) go back to 1.



Do loop example

- Displays a confirm window repeatedly, **until** the user clicks Cancel.

```
do
{
    var c=confirm("Repeat again?");
}while (c==true);
```

While loop

- This loop first checks the condition and then executes the statements in the loop if the condition is true.
- Executes **WHILE** loop condition is true.
- While loops don't have to execute at least once.

```
while(<loop condition>)  
{  
    <statement(s) to repeat while loop condition is true>  
}
```

- Steps:
 1. Check condition.
 2. If false, end loop.
 3. Else (i.e. If condition is true) execute statements in the loop.
 4. go back to 1.
-

While loop example

- Same example as the do loop example:
 - Will keep displaying a confirm window **WHILE** the user clicks on OK:

```
// c is initialized to true to force
// the confirm window the first time
var c = true;
while(c==true)
{
    c=confirm("Repeat?") ;
}
```


Arrays

- Indexed group of variables

- Groups together similar information (e.g. names of salespersons, number of cars sold by a salesperson)
- Still a variable
 - Declared with **var** and **new Array()**

```
var <array name> = new Array(<optionalSize>);  
var colorArray = new Array(3);  
var anotherArray = new Array();
```
- Use the index to refer to a location within the array
 - Indexes start at ZERO.
- Allows convenient access via loop counters.

Arrays example

- Create an array of 3 elements and assign 3 colors as values to each array element:

```
var colorArray = new Array(3);  
colorArray[0] = "red";  
colorArray[1] = "green";  
colorArray[2] = "blue";
```

| colorArray | |
|------------|-------|
| 0 | red |
| 1 | green |
| 2 | blue |

index

Array elements / values

Array example 2

- Get the names of 3 people and display them. Uses 1 loop to collect the info, and 1 loop to display it.

```
var names = new Array(3);
```

```
for (var i=0; i<3; i++)
```

```
    names[i] = prompt("What is the name of person " + i, "");
```

```
for (var i=0; i<3; i++)
```

```
    document.write("<br>Person " + i + " is called " + names[i]);
```

تعمیرین

* برنامه ای بنویسید که اسامی و نمرات تعدادی دانشجو را دریافت کرده و میانگین نمرات ، لیست اسامی و نمرات را چاپ کند.

* در پایان تعداد مقبولین و مردودین را نیز نمایش دهد.

JavaScript Functions

- **function** *functionname*(*var1,var2,...,varX*)
{
 some code
 return value;
}

```
function add(a,b) {  
  var x=a+b ;  
  return x;  
}
```

- Call function:
 a=add(2,3);

Function Example

```
<html>
<head>
  <script type="text/javascript">
    function displaymessage()
      { alert("Hello World!"); }
  </script>
</head>
<body>
  <p onclick="displaymessage()"> test function </p>
  <script type="text/javascript">
    displaymessage();
  </script>
</body>
</html>
```

فراخوانی به روش غیرمستقیم

فراخوانی به روش مستقیم

Events

| | | | |
|----------|--|-------------|---|
| onAbort | Occurs when the user aborts the loading of an image | onLoad | Occurs when the page (or an image) finishes loading |
| onBlur | Occurs when an object on the page loses focus | onMouseOver | Occurs when the mouse pointer moves over an item |
| onChange | Occurs when a text field is changed by the user | onMouseOut | Occurs when the mouse pointer moves off an item |
| onClick | Occurs when the user clicks on an item | onSelect | Occurs when the user selects text in a text area |
| onError | Occurs when a document or image can't load correctly | onSubmit | Occurs when a submit button is pressed |
| onFocus | Occurs when an item gains focus | onUnload | Occurs when the user leaves the document or exits |

Event handling

- When an event occurs you call a function to handle that event.

- Example:

```

```


Timer

- `var timername=setTimeout ('js code',milliseconds);`

تعریف تایمر

- `clearTimeout (timername);`

حذف و غیرفعالسازی تایمر

- 'js Code' usually is a Function called recursively.

معمولا فراخوانی تابع است!

JS Functions

■ `var d = new Date();`

| | |
|---------------------------------------|--|
| <u><code>getDate()</code></u> | Returns the day of the month (from 1-31) |
| <u><code>getDay()</code></u> | Returns the day of the week (from 0-6) |
| <u><code>getFullYear()</code></u> | Returns the year (four digits) |
| <u><code>getHours()</code></u> | Returns the hour (from 0-23) |
| <u><code>getMilliseconds()</code></u> | Returns the milliseconds (from 0-999) |
| <u><code>getMinutes()</code></u> | Returns the minutes (from 0-59) |
| <u><code>getMonth()</code></u> | Returns the month (from 0-11) |
| <u><code>getSeconds()</code></u> | Returns the seconds (from 0-59) |

EVAL function

- The **eval()** function **evaluates** or **executes** an argument.
 - If the argument is an expression, eval() evaluates the expression.
 - If the argument is one or more JavaScript statements, eval() executes the statements.
- **eval (string)**
 - A JavaScript expression, variable, statement, or sequence of statements

eval() - Example


- Evaluate/Execute JavaScript code/expressions:
- `<script>`
 eval("x=10;y=20;document.write(x*y)");
 document.write("
" + **eval**("2+2"));
 document.write("
" + **eval**(x+17));
`</script>`
- The output of the code above will be:
 - **200**
 - 4**
 - 27**

Try...Catch

```
■ try
{
    //Some codes here
}
catch(err)
{
    //Handle errors here
}
```

کدهای اصلی برنامه

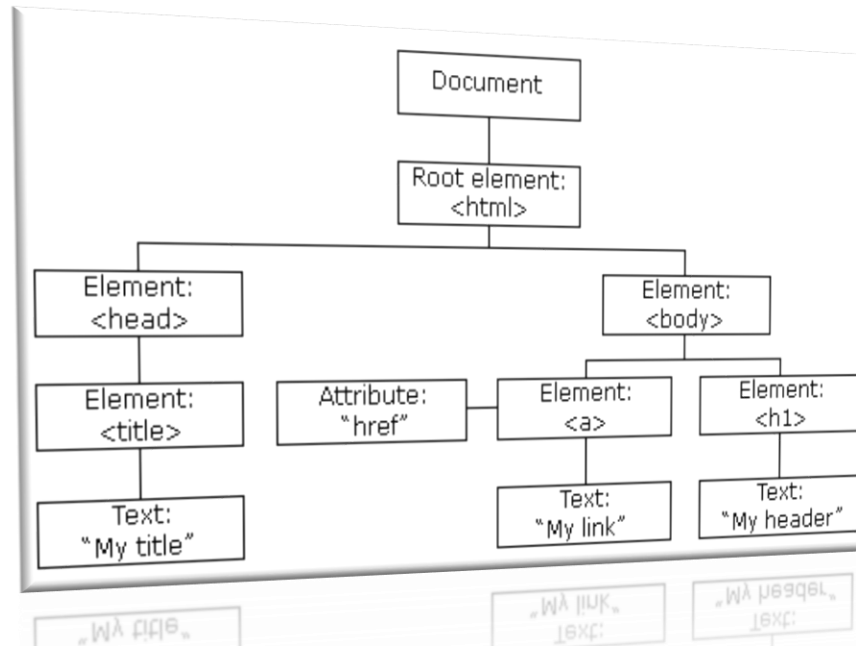
کدهایی که در صورت بروز خطا اجرا
خواهند شد



```
<script type="text/javascript">
function message()
{
try
{
addalert("Welcome guest!");
}
catch(err)
{
txt="Error description: " + err.description;
alert(txt);
}
}
</script>
```

The DOM

- DOM = “Document Object Model”.
- HTML is a Tree!
- The DOM is a way for you to “Walk” the tree.



Document Object Methods

| Methods | |
|-------------------------------|--|
| getElementById() | دسترسی به المان توسط شناسه یا همان ID. |
| getElementsByTagName() | دسترسی به المان توسط نام یا همان Name. |
| write() | دسترسی به المان توسط نام تگ یک المان. |
| writeln() | برای نوشتن یک رشته در صفحه. |
| | برای نوشتن یک رشته در صفحه در یک خط جدید |

Access to Elements

■ **document.getElementById('an id in HTML')**

- ☐ **.style**
- ☐ **.innerHTML**
- ☐ **.value**
- ☐ **.setAttribute(Name, Value)**
- ☐ **.getAttribute(Name)**

Exercises

1. Alert an Element inside Content by ID.
2. Hide an element in HTML by ID.
3. Change Color of an element by ID.
4. Set Attribute an Element.
5. Get All Paragraphs Content.

JavaScript's Objects

| Object | |
|-----------|---|
| Window | بالاترین سطح آبجکت به لحاظ سلسله مراتب در جاوا اسکریپت می باشد که نشان دهنده پنجره مرورگری است که تمام صفحه داخل آن قرار می گیرد. |
| Navigator | حاوی اطلاعات مرورگر کاربر می باشد. |
| Screen | حاوی اطلاعاتی در خصوص صفحه نمایش کاربر می باشد. |
| History | حاوی اطلاعات صفحات مشاهده شده در مرورگر می باشد. |
| Location | نشان دهنده URL جاری در حال نمایش است. |

■ navigator

- appName نام مرورگر
- appVersion پلتفرم و نسخه مرورگر

■ window

- close() بستن پنجره جاری
- moveTo() تغییر مکان پنجره به مکان مشخص

■ screen

- availHeight (pixel) ارتفاع صفحه نمایش
- availWidth (pixel) عرض صفحه نمایش

■ Location

- assign() انتقال صفحه جاری به آدرس مشخص
- reload() بارگذاری مجدد صفحه

■ History

- back() رفتن به صفحه قبلی تاریخچه
- forward() رفتن به صفحه بعدی تاریخچه



Examples

Of JavaScript Sources

1

```
■ <script type="text/javascript">  
function newDoc()  
{  
  window.location.assign("http://www.w3schools.com")  
}  
</script>
```

2

```
<html>
  <head>
    <script type="text/javascript">
      function changecolor()
      {
        document.getElementById('header').style.color="red"
      }
    </script>
  </head>
  <body>
    <h1 id="header" onclick="changecolor()">
      Click on this text
    </h1>
  </body>
</html>
```

3

```
<html>
  <head>
    <script type="text/javascript">
      function changestatus(arg)
      {
        window.status = arg;
      }
    </script>
  </head>
  <body>
    <h1 onclick="changestatus('You are click on the text')">
      Click on this text
    </h1>
  </body>
</html>
```


4

```
<html>
  <head>
    <script type="text/javascript">
      function disable()
      {
        if (event.button == 2)
        {
          alert("Sorry no rightclick on this page. ")
        }
      }
    </script>
  </head>

  <body onmousedown="disable()">

    <p>Right-click on this page to trigger the event.</p>

  </body>
</html>
```

5

```
<html>
<head>
  <script>
    function startEQ()
    {
      richter=5
      parent.moveBy(0,richter)
      parent.moveBy(0,-richter)
      parent.moveBy(richter,0)
      parent.moveBy(-richter,0)
      timer=setTimeout("startEQ()",10)
    }
    function stopEQ()
    {
      clearTimeout(timer)
    }
  </script>
</head>
<body>
<h1 onclick="startEQ()"> Start an earthquake </h1>
<br/>
<h1 onclick="stopEQ()"> Stop the earthquake </h1>
</body>
</html>
```

6

```
<html>
<head>
<script type="text/javascript">
    function shakeleft()
    {
        document.getElementById('myLink').style.position="relative"
        document.getElementById('myLink').style.left="3"
        timer=setTimeout("shakeright()",10)
    }
    function shakeright()
    {
        document.getElementById('myLink').style.left="0"
        timer=setTimeout("shakeleft()",10)
    }
    function stoptimer()
    {
        clearTimeout(timer)
    }
</script>
</head>
<body>
    <a id="myLink" href="http://www.w3schools.com/"
        onmouseover="shakeleft()" onmouseout="stoptimer()">Mouse over this link</a>
</body>
</html>
```

7

```
<html>
<head>
<script type="text/javascript">
  function cursor(event)
  {
    document.getElementById('trail').style.visibility="visible"
    document.getElementById('trail').style.position="absolute"
    document.getElementById('trail').style.left=event.clientX+10
    document.getElementById('trail').style.top=event.clientY
  }
</script>
</head>
<body onmousemove="cursor(event)">
<h1>Move the cursor over this page</h1>

</body>
</html>
```



more info & samples just in:

<http://www.w3schools.com>