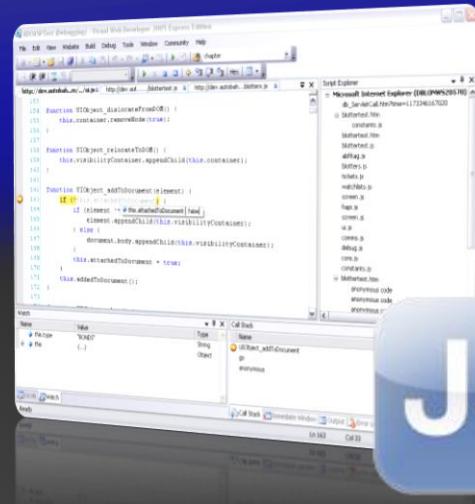


javascript
for
beginner

```
String.prototype.trim =  
function ()  
{  
    return this  
        .replace (^\\s+,"")  
        .replace (\\s+$,"");  
}
```

.js



Introduction to JavaScript

ff javascript

DOM

Style

Events

What is DHTML?

- ◆ Dynamic HTML (DHTML)
 - ◆ Makes possible a Web page to react and change in response to the user's actions



Implementing JavaScript

- ◆ The JavaScript code can be placed in:
 - ◆ <script> tag in the head
 - ◆ <script> tag in the body
 - ◆ External files, linked via <script> tag the head
 - ◆ Files usually have .js extension

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

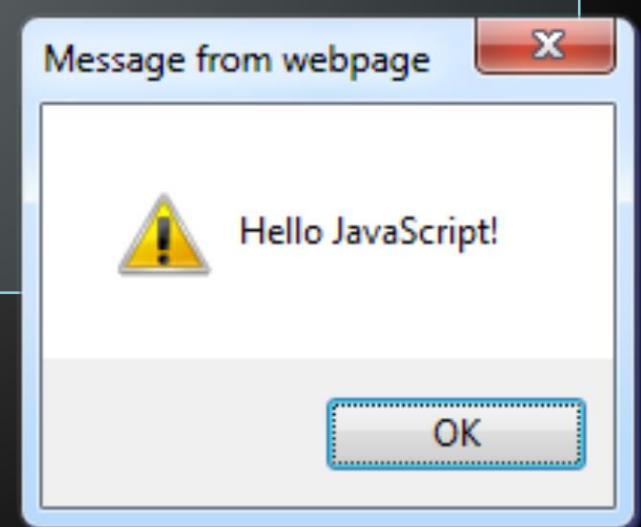
The First Script

first-script.html

```
<html>
<head><title>First script</title>
</head>

<body>
  <script type="text/javascript">
    alert("Hello JavaScript!");
  </script>
</body>

</html>
```



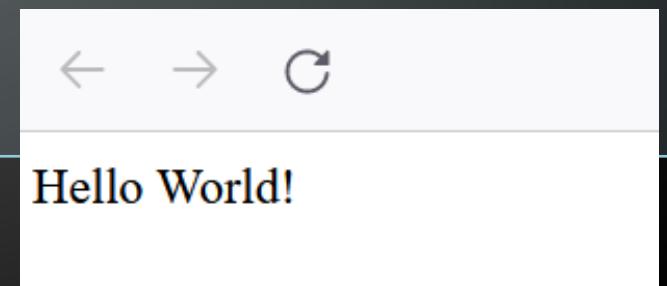
Another Small Example

small-example.html

```
<html>
<head><title>First script</title>
</head>

<body>
  <script type="text/javascript">
    document.write("Hello World!");
  </script>
</body>

</html>
```

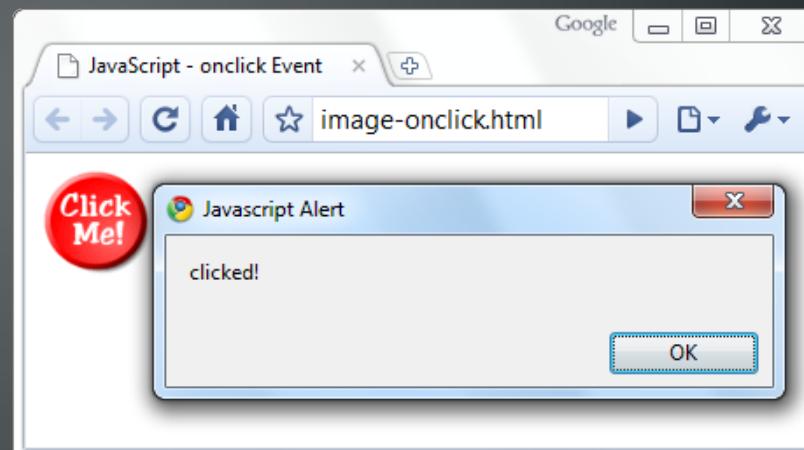


Calling a JavaScript Function from Event Handler – Example

```
<html>
<head>
<script type="text/javascript">
    function test (message) {
        alert(message);
    }
</script>
</head>

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

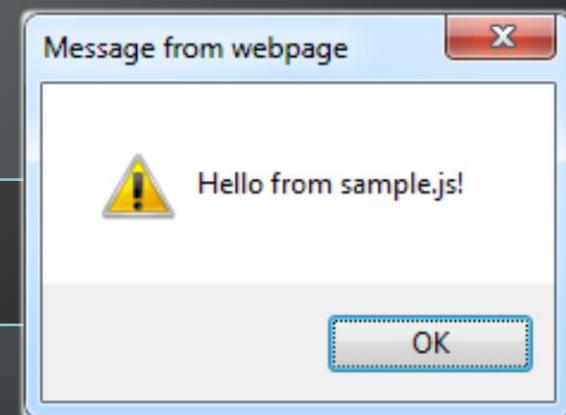
image-onclick.html



Using External Script Files

- ◆ Using external script files:

```
<html>                                external-JavaScript.html
<head>
  <script src="sample.js" type="text/javascript">
  </script>
</head>
<body>
  <input type="button" value="Click here"
        onclick="sample()">
</body>
</html>
```

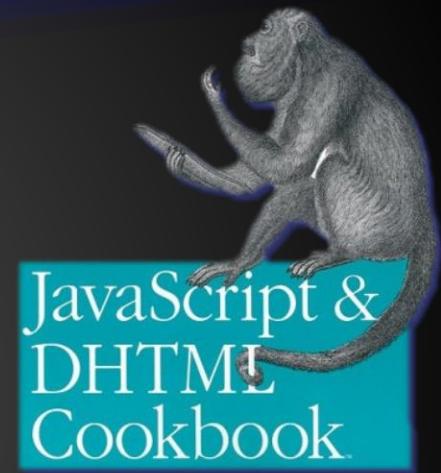


- ◆ External JavaScript file:

```
function sample() {
  alert("Hello from sample.js!")
}
```

The JavaScript Syntax

```
if (pop < 10)
{
    map.graphics.add(features[i].setSymbol(onePopSymbol));
}
else if (pop >= 10 && pop < 95)
{
    map.graphics.add(features[i].setSymbol(twoPopSymbol));
}
else if (pop >= 95 && pop < 365)
{
    map.graphics.add(features[i].setSymbol(threePopSymbol));
}
else if (pop >= 365 && pop < 1100)
{
    map.graphics.add(features[i].setSymbol(fourPopSymbol));
}
else
{
    map.graphics.add(features[i].setSymbol(fivePopSymbol));
}
```



JAVA
SCRIPT

JavaScript Syntax

- Operators (+, *, =, !=, &&, ++, ...)
- Variables (typeless)
- Conditional statements (if, else)
- Functions

- ◆ JavaScript data types:
 - ◆ Numbers (integer, floating-point)
 - ◆ Boolean (true / false)
- ◆ In JavaScript, variables are used to store data. They can be declared using the keywords var, let, or const.

```
var a = 15;  
let b = "Hello";  
const c = 12;
```

parseInt : Used to parse a string and extract an integer.

parseFloat : Used to parse a string and extract a floating-point number.

isNaN : returns true if the value is not a number

Example- parseInt

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

var a = "10.33";
var b = "34 55 66";
var c = "40 years";
var d = "I am 60 years old";

document.write(parseInt(a)); //10
document.write(parseInt(b)); //34
document.write(parseInt(c)); //40
document.write(parseInt(d)); //NaN

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

Example- parseFloat

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

var a = "10.33";
var b = "34 55 66";
var c = "40 years";
var d = "I am 60 years old";

document.write(parseFloat(a)); //10.33
document.write(parseFloat(b)); //34
document.write(parseFloat(c)); //40
document.write(parseFloat(d)); //NaN

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

Example- isNaN

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

document.write(isNaN(123)); //false
document.write(isNaN(-1.23)); //false
document.write(isNaN(5-2)); //false
document.write(isNaN(0)); //false
document.write(isNaN('Hello'))); //true
document.write(isNaN('2005/12/12'))); //true

</script>

</body>
</html>
```

String Operations

- ◆ The + operator joins strings

```
string1 = "hello ";
string2 = "world!";
alert(string1 + string2); // hello world!
```

- ◆ What is "9" + 9?

```
alert("9" + 9); // 99
```

- ◆ Converting string to number:

```
alert(parseInt("9") + 9); // 18
```

Conditional Statement (if)

```
var unitPrice = 1.30;  
var number = 102;  
if (number > 100) {  
    unitPrice = 1.20;  
}
```

Symbol	Meaning
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
==	Equal
!=	Not equal

if(condition)

```
<html>
<head>
  <title>JavaScript Demo</title>
  <script type="text/javascript">
    var number = 5;
    if (number > 0) {
      document.write("The number is positive");
    }
    else {
      document.write("The number is negative");
    }
  </script>
</head>
<body>
</body>
</html>
```

if(condition)

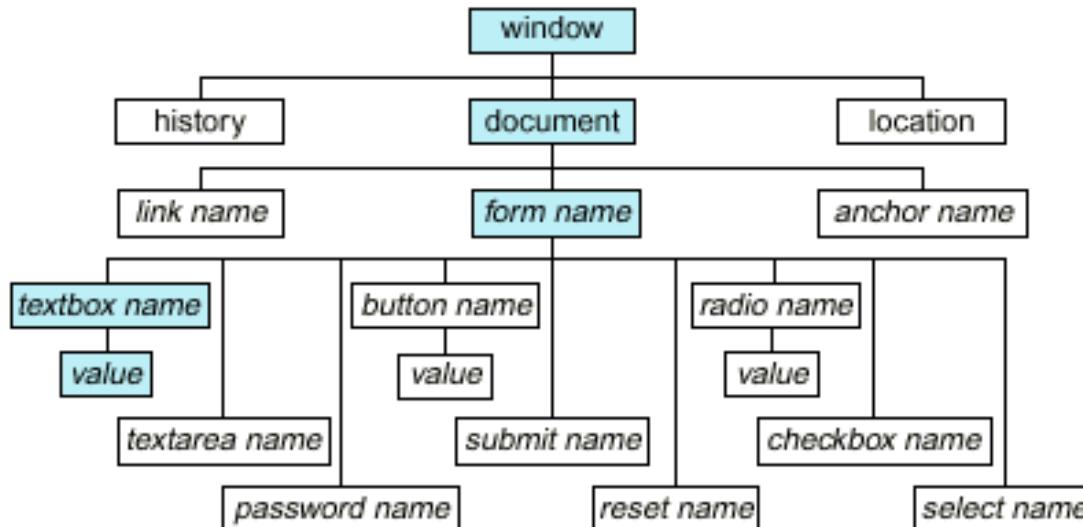
◆ Operator (AND)

```
<html>
<body>
<script>
var x = 5;
var y = 6;
if ( x>0 && y>0 ) {
    document.write("x and y are greater than 0");
} else {
    document.write("x and y are not greater than 0");
}
</script>
</body>
</html>
```

if(condition)

◆ Operator (OR)

```
<html>
<body>
<script>
var x = -5;
var y = 1;
if ( x>0 || y<0 ) {
    document.write("The condition is true");
} else {
    document.write("The condition is false");
}
</script>
</body>
</html>
```



The JavaScript Object Model

Document Object Model (DOM)

getElementById

```
var elem = document.getElementById("mon_id")
```

- ◆ It allows to access and modify the content or attributes of that element.

getElementById - Image

- ◆ Example with an image :

- ◆ In HTML tag :

```

```

- ◆ DOM:

```
var img = document.getElementById("myImage");  
img.onclick = imageClicked;
```

getElementById - Image

```
<html>
<head><title>DOM Image</title>
</head>
<body>


<script>
function imageClicked(){
alert("Hello world!");
}

var img = document.getElementById("myImage");
img.onclick = imageClicked;
</script>
</body>
</html>
```

getElementById – CSS style

- ◆ Change CSS style:

- ◆ To change the style of an HTML element, use this syntax:

```
document.getElementById("id").style.property =new style;
```

```
<html>
<head><title>CSS style</title></head>
<body>
<p id="p2">Hello World!</p>
<script>
document.getElementById("p2").style.color = "blue";
</script>
</body>
</html>
```

getElementById – change content

- ◆ Change HTML content:
 - ◆ To change the content of an HTML element, use this syntax:

```
document.getElementById("id").innerHTML = "new content";
```

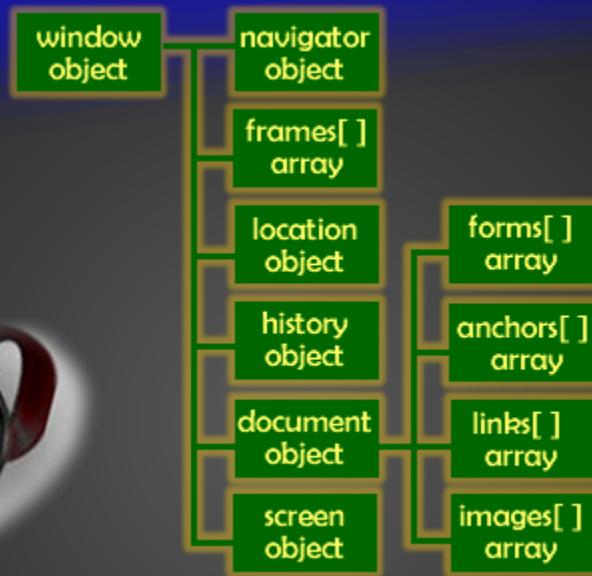
getElementById

```
<html>
<head><title>innerHTML</title></head>
<body>

<p id="demo" onclick="myFunction()">Hello, world!</p>

<script>
function myFunction() {
    document.getElementById("demo").innerHTML = "Hello
everyone!";
}
</script>

</body>
</html>
```



The Built-In Browser Objects

Open new window

◆ **window.open()**

window-open.html

```
<html>
<head><title>New window</title></head>
<body>

<button onclick="myFunction()">click here</button>

<script>
function myFunction() {
    window.open("https://www.google.com"); //add _self
}
</script>

</body>
</html>
```

Open new window

◆ **window.open()**

window-open.html

```
<html>
<body>
<button onclick="myFunction()">click here</button>

<script>
function myFunction() {
    var myWindow = window.open("", "", "width=200,height=100"); //add _self
    myWindow.document.write("Hello!");
}
</script>
</body>
</html>
```

JS

Date in JavaScript



Date and Timer in JavaScript

Start with new Date(), and use the functions below :

- ◆ **getDate()** : Returns the day of the month (1-31) for the specified date.
- ◆ **getFullYear()** : Returns the year of the specified date.
- ◆ **getMonth()** : Returns the month (0-11) for the specified date.
- ◆ **getHours()** : Returns the hour (0-23) for the specified date.
- ◆ **getMinutes()** : Returns the minutes (0-59) in the specified date.
- ◆ **getSeconds()** : Returns the seconds (0-59) in the specified date.

Timer: setTimeout()

- ◆ Make something happen (once) after a fixed delay

```
var timer = setTimeout('bang()', 5000);
```

5 seconds after this statement executes, this function is called

```
clearTimeout(timer);
```

Cancels the timer

setTimeout– Example

timer-demo.html

```
<script type="text/javascript">
    function timerFunc() {
        var now = new Date();
        var hour = now.getHours();
        var min = now.getMinutes();
        var sec = now.getSeconds();
        document.getElementById("clock").value =
            "" + hour + ":" + min + ":" + sec;
    }
    setTimeout('timerFunc()', 5000);
</script>

<input type="text" id="clock" />
```

Timer: setInterval()

- ◆ Make something happen repeatedly at fixed intervals

```
var timer = setInterval('clock()', 1000);
```

This function is called
continuously per 1 second.

```
clearInterval(timer);
```

Stop the timer.

setInterval – Example

timer-demo.html

```
<script type="text/javascript">
    function timerFunc() {
        var now = new Date();
        var hour = now.getHours();
        var min = now.getMinutes();
        var sec = now.getSeconds();
        document.getElementById("clock").value =
            "" + hour + ":" + min + ":" + sec;
    }
    setInterval('timerFunc()', 1000);
</script>

<input type="text" id="clock" />
```