Technical University of Sofia
Faculty of Computer Systems and Control

Web Programming

Lecture 4

JavaScript

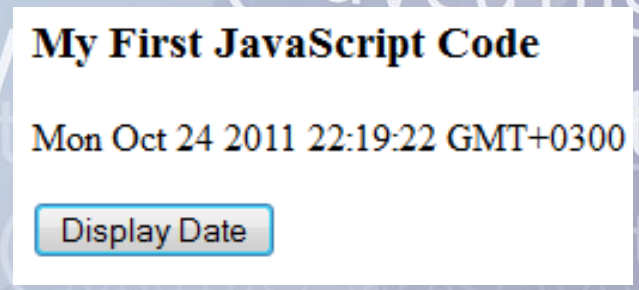
JavaScript basics

- JavaScript is scripting language for Web.
- JavaScript is used in billions of Web pages to add functionality, validate forms, communicate with the server, and much more.
- JavaScript is easy to learn. You will enjoy it.
- Works in all major browsers, such as Internet Explorer, Firefox, Chrome, Opera, and Safari.

Example

```
<html>
<head>
<script type="text/javascript">
function displayDate() {

document.getElementById("demo").innerHTML=Date();
}
</script>
</head>
<body>
  <h3>My First JavaScript Code</h3>
  <p id="demo">This is a paragraph.</p>
  <button type="button" onclick="displayDate()">
     Display Date
  </button>
</body>
</html>
```



What is JavaScript?

- A scripting language is a lightweight programming language
- JavaScript is an interpreted language (means that scripts execute without preliminary compilation)
- JavaScript was designed to add interactivity to HTML pages and is usually embedded directly into them
- Everyone can use JavaScript without purchasing a license

Java and JavaScript

- Java and JavaScript are two completely different languages in both concept and design!
- Java is developed by Sun Microsystems.
- Java is a powerful and much more complex programming language - in the same category as C and C++.

What Can JavaScript do?

- **JavaScript gives HTML designers a programming tool**

  - HTML authors are normally not programmers, but JavaScript is a scripting language with a very simple syntax! Almost anyone can put small "snippets" of code into their HTML pages.

What Can JavaScript do?

- **JavaScript can react to events**

  - A JavaScript can be set to execute when something happens, like when a page has finished loading or when a user clicks on an HTML element

What Can JavaScript do?

- **JavaScript can read and write HTML elements**
  - A JavaScript can read and change the content of an HTML element.
- **JavaScript can be used to validate data**
  - A JavaScript can be used to validate form data before it is submitted to a server. This saves the server from extra processing

What Can JavaScript do?

- **JavaScript can be used to detect the visitor's browser**

  - A JavaScript can be used to detect the visitor's browser, and - depending on the browser - load another page specifically designed for that browser.

- **JavaScript can be used to create cookies**

  - A JavaScript can be used to store and retrieve information on the visitor's computer

How to add JS?

- JavaScripts can be put in the <body> and in the <head> sections of an HTML page.
- Note that the JavaScript should be placed at the bottom of the page to make sure it is not executed before the elements manipulated in it are created.

When it executes?

- JavaScripts in an HTML page will be executed when the page loads. This is not always what we want.
- Sometimes we want to execute a JavaScript when an **event** occurs, such as when a user clicks a button. When this is the case we can put the script inside a **function**.

Scripts in <head> and <body>

- You can place an unlimited number of scripts in your document, and you can have scripts in both the body and the head section at the same time.
- It is a common practice to put all functions in the head section, or at the bottom of the page. This way they are all in one place and do not interfere with page content.

External JavaScript

- JavaScript can also be placed in external files.
- External JavaScript files often contain code to be used on several different web pages.
- External JavaScript files have the file extension .js
- External script cannot contain the <script></script> tags!

Import external JavaScript

- To use an external script, point to the .js file in the "src" attribute of the <script> tag:

```
<html>
<head>
<script type="text/javascript“
        src=“name.js"></script>
</head>
<body>
</body>
</html>
```

JavaScript Statements

- JavaScript is a sequence of statements to be executed by the browser.
- A JavaScript statement is a command to a browser. The purpose of the command is to tell the browser what to do.
- JavaScript is case sensitive - therefore watch your capitalization closely when you write JavaScript statements, create or call variables, objects and functions.

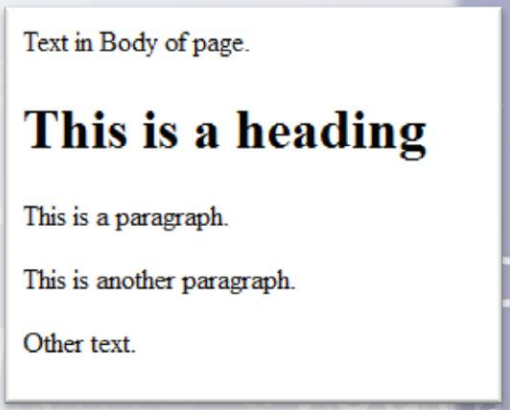
JavaScript Code

- JavaScript code (or just JavaScript) is a sequence of JavaScript statements.
- Each statement is executed by the browser in the sequence they are written.
- This example will write a heading and two paragraphs to a web page:

JavaScript Code - Example

- This example will write a heading and two paragraphs to a web page:

```
<html>
<body>
Text in Body of page.
<script type="text/javascript">
document.write("<h1>This is a heading</h1>");
document.write("<p>This is a paragraph.</p>");
document.write("<p>This is another paragraph.</p>");
</script>
Other text.
</body>
</html>
```



Examples

- Example 1
- Example 2
- Example 3

JavaScript Variables

- Variables are used to store data.
- A variable is a "container" for information you want to store. A variable's value can change during the script. You can refer to a variable by name to see its value or to change its value.
- Rules for variable names:
  - Variable names are case sensitive
  - They must begin with a letter or the underscore character
    - strname – STRNAME (not same)

JavaScript Variables - Types

- There is Dynamic Typing – variables can hold any valid type of value:
  - Number … `var myInt = 7;`
  - Boolean … `var myBool = true;`
  - Function … (for later)
  - Object … (for later)
  - Array … `var myArr = new Array();`
  - String … `var myString = “abc”;`
  - … and can hold values of different types at different times during execution.

JavaScript Operators

Arithmetic Operators

Operator	Description	Example	Result
+	Addition	x=2 y=2 x+y	4
-	Subtraction	x=5 y=2 x-y	3
*	Multiplication	x=5 y=4 x*y	20
/	Division	15/5 5/2	3 2,5
%	Modulus (division remainder)	5%2 10%8 10%2	1 2 0
++	Increment	x=5 x++	x=6
--	Decrement	x=5 x--	x=4

JavaScript Operators

Assignment Operators

Operator	Example	Is The Same As
=	x=y	x=y
+=	x+=y	x=x+y
-=	x-=y	x=x-y
*=	x*=y	x=x*y
/=	x/=y	x=x/y
%=	x%=y	x=x%y

JavaScript Operators

Comparison Operators

Operator	Description	Example
==	is equal to	5==8 returns false
===	is equal to (checks for both value and type)	x=5 y="5" x==y returns true x===y returns false
!=	is not equal	5!=8 returns true
>	is greater than	5>8 returns false
<	is less than	5<8 returns true
>=	is greater than or equal to	5>=8 returns false
<=	is less than or equal to	5<=8 returns true

JavaScript Operators

Logical Operators

Operator	Description	Example
&&	and	x=6 y=3 (x < 10 && y > 1) returns true
||	or	x=6 y=3 (x==5 || y==5) returns false
!	not	x=6 y=3 !(x==y) returns true

Example

```
<html>
<body>
<script>
x="Hello World!"
document.write(x)
document.write("<br>")
</script>

<script>
x="GOOD BYE"
document.write("CLASS" +x)
</script>

</body>
</html>
```

JavaScript Popup Boxes

Alert Box

- An alert box is often used if you want to make sure information comes through to the user.
- When an alert box pops up, the user will have to click "OK" to proceed.

```
<script>
alert("Hello World!")
</script>
```

JavaScript Popup Boxes

Confirm Box

- A confirm box is often used if you want the user to verify or accept something.
- When a confirm box pops up, the user will have to click either "OK" or "Cancel" to proceed.
- If the user clicks "OK", the box returns true. If the user clicks "Cancel", the box returns false.

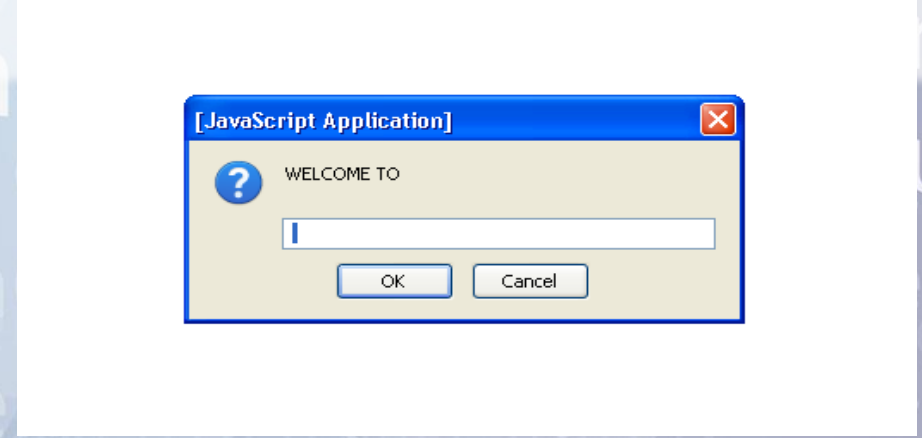
JavaScript Popup Boxes

Prompt Box

- A prompt box is often used if you want the user to input a value before entering a page.
- When a prompt box pops up, the user will have to click either "OK" or "Cancel" to proceed after entering an input value.
- If the user clicks "OK“, the box returns the input value. If the user clicks "Cancel“, the box returns null.

Prompt Box Example

```
<script>
x=prompt ("WELCOME TO", " ")
document.write("CLASS <br>",+x)
</script>
```



JS Examples

Y=20x+12 ;where x=3

```
<script>
x=3;
y=20*x+12;
alert(y);
</script>
```

The answer should be?

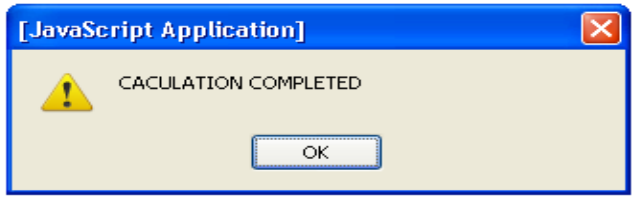
JS Examples

```
<script>
s1=12;
s2=28;
sum=s1+s2;
document.write(“Sum of numbers is: "+sum);
</script>
```

JS Examples

```
<script>
    s1=12; s2=6;
    add=s1+s2;
    sub=s1-s2;
    mul=s1*s2;
    divl=s1/s2;
    document.write(add+"<br>");
    document.write(sub+"<br>");
    document.write(mul+"<br>");
    document.write(divl+"<br>");
    alert("CACULATION COMPLETED");
</script >
```

```
18
6
72
2
```



Conditional Statements

- Very often when you write code, you want to perform different actions for different decisions. You can use conditional statements in your code to do this.

In JavaScript we have the following conditional statements:

- **if statement** - use this statement if you want to execute some code only if a specified condition is true
- **if...else statement** - use this statement if you want to execute some code if the condition is true and another code if the condition is false
- **if...else if....else statement** - use this statement if you want to select one of many blocks of code to be executed
- **switch statement** - use this statement if you want to select one of many blocks of code to be executed

Conditional Statements

```
if (condition) {
    code to be executed if condition is true
}
```

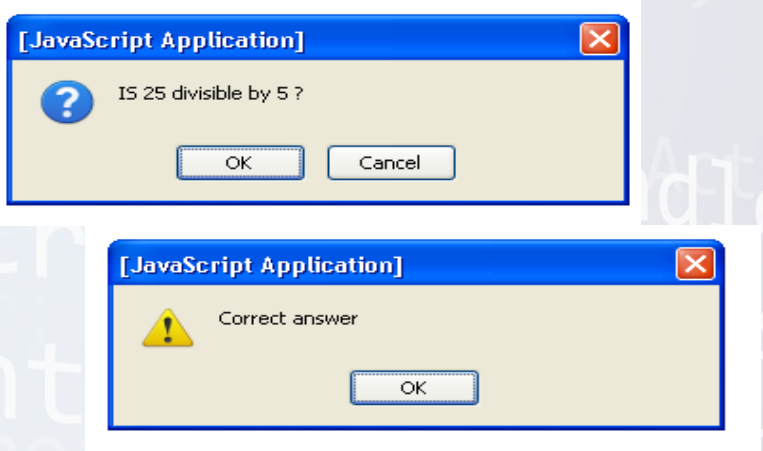
```
if (condition) {
    code to be executed if condition is true
}
else {
    code to be executed if condition is not true
}
```

Conditional Statements Examples

```
<script>
x=3;
if(x<0)
{
    alert ("negative");
}
else
{
    alert ("pozitive");
}
</script>
```

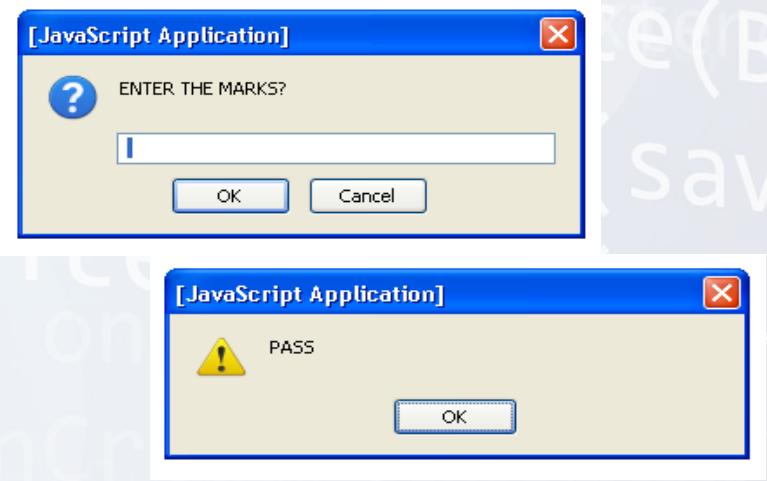
Conditional Statements Examples

```
<script>
c=confirm("IS 25 divisible by 5 ?");
if(c)
{
    alert ("Correct answer");
}
else
{
    alert ("wrong answer");
}
</script>
```



Conditional Statements Examples

```
<script>
p=prompt(“ENTER THE MARK? (2 to 6)", " ");
if(p>=“3") {
alert(“PASS");
}
else {
alert(“FAIL");
}
</script>
```



Conditional Statements – “switch”

```
switch (myVar) {
  case 1:
      // if myVar is equal to 1 this is executed
  case “two”:
      // if myVar is equal to “two” this is executed
  case default:
      // if none of the cases above are satisfied OR if
      // no “break” statement is used in the cases
      // above, this will be executed
}
```

JS Functions

- Declaring a function by using the “function” keyword
  - No return type, nor typing of arguments

```
function add (numOne, numTwo) {
   return numOne + numTwo;
}
```

- Calling a function – myFunc (arg1, arg2, …);

```
add (4,5);
```

JS Functions Example

```
<html> <head>
<script Language="JavaScript">
function getMax(n1, n2){
if (n1 < n2) return n2;
else return n1;
}
</script> </head>
<body> <script Language="JavaScript">
document.write("<p>") ;
document.write(“Largest number of 5 or 6 is: " + getMax(5, 6));
 document.write("</p>")
</script> </body>
</html>
```

Thank you for your attention!