INTRODUCTION TO JAVASCRIPT

Lecture 6.

Senior- Lecturer: Sarsenova Zhibek

JAVASCRIPT

- JavaScript is a simplified programming language designed to beef up web pages with interactive features.
- JavaScript is perfect for creating pop-up windows, embedding animated effects, or modifying the content that appears on your web page.

JAVASCRIPT

- You can display a personalized message to your visitors(“Hello, Joe!”) or make title grow and shrink perpetually
- Gather information about date, your visitors browser, or the content your visitor types into a form.
- React to events that take place in a browser. For example, you can add JavaScript code that runs when a page finishes loading or when a visitor clicks a picture.

- JavaScript is one of the 3 language all web developers must learn:
- HTML to define the content of web pages
- CSS to specify the layout of web pages.

SERVER-SIDE AND CLIENT-SIDE PROGRAMMING

- **Server-side** applications rule the web world. However, they're difficult to program. Not only do developers need to worry about getting the program to generate HTML for a browser, they also need to make sure the program can run all kinds of complex code and tap giant databases—and they need to do it so that the site performs just as well when millions of people view it as it does when only one person visits it.

CLIENT-SIDE

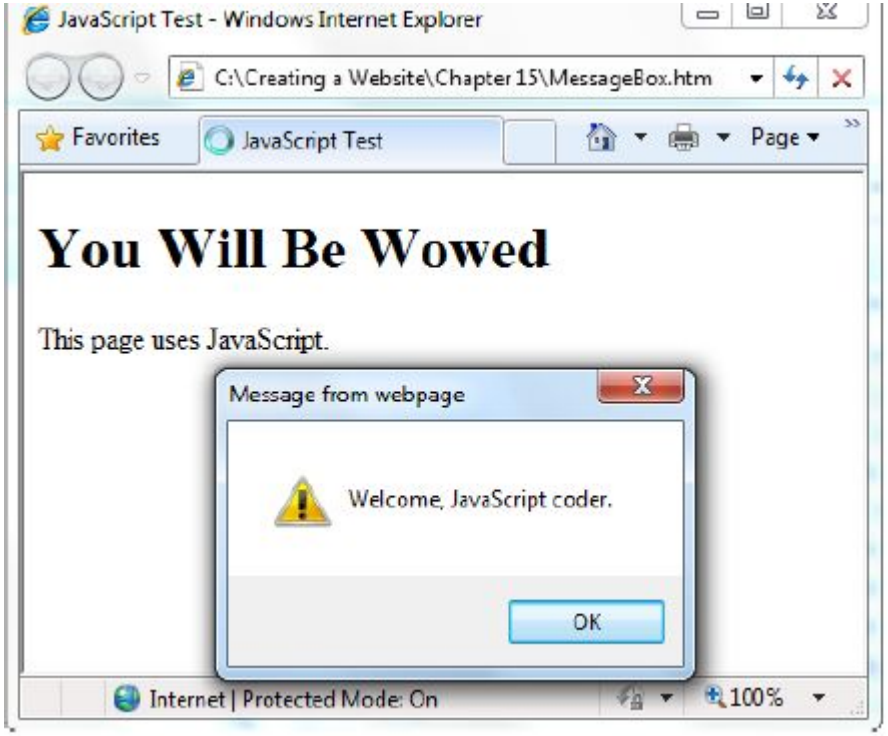
- *Client-side applications, on the other hand, use a completely different model. They* embed small, lightweight programs inside an ordinary HTML page. When a browser downloads the page, the browser itself runs the program (assuming your security settings or compatibility issues haven't disabled the program). Client-side programs are much less powerful than those on the server side—they can't reliably poll the huge databases stored on web servers, for example, and for security reasons they can't directly change most things on your computer. However, they're much simpler to create.

THE SCRIPT ELEMENT

- **The <body> section**. Put script you want your browser to run right away in the <body> section of your HTML. The browser runs your script as soon as it reaches the <script> element. If you put your script at the beginning of the <body> section, your browser process the script before it displays the page.
- **The <head> section.** It runs immediately, before the browser processes any part of the markup.

EXAMPLE

```
<!DOCTYPE html>

<html>

<head>

  <title>JavaScript Test</title>
</head>

<body>
  <h1>You Will Be Wowed</h1>
  <p>This page uses JavaScript.</p>
  <script type="text/javascript">
    alert("Welcome, JavaScript coder.")
  </script>
</body>

</html>
```



VARIABLES

- Temporary containers that store important information. Variables can store numbers, objects, or pieces of text.

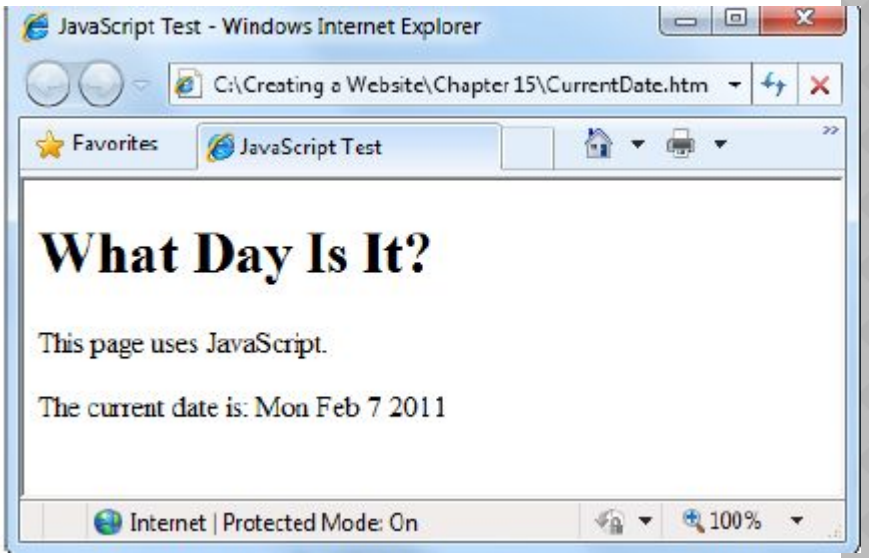
DECLARING VARIABLES

- var , followed by the name of the variable.
- var myMessage
- To store (=), copies the data on the right side of the equal sign into the variable on the left.
- myMessage = “Everybody loves variables”

EXAMPLE

- 1. Creates a new variable named currentDate. It fills the currentDate variable with a new Date object. New keyword is crate an object .
- 2. creates a new variable named message and fills it with the beginning of a sentence that announces the date.
- 3.This line adds information created in line 2.currentDate object comes with a built-in function. toDateString(), that converts the date information it gets from your computer into a piece of text suitable for display in a browser.
- 4. uses document object, which has a function named write(), displays a piece of text on a web page at the current location. The final result is a page that shows your welcome message.

```
1   var currentDate = new Date()
2   var message = "The current date is: "
3   message = message + currentDate.toDateString()
4   document.write(message)
```



JAVASCRIPT DISPLAY POSSIBILITIES

- JavaScript can "display" data in different ways:
- 1. Writing into an alert box, using **window.alert()**.
- 2. Writing into the HTML output using **document.write()**.
- 3. Writing into an HTML element, using **innerHTML**.
- 4. Writing into the browser console, using **console.log()**.

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

<h1>My First Web Page</h1>
<p>My first paragraph.</p>

<script>
window.alert(5 + 6);
</script>

</body>
</html>
```

USING DOCUMENT.WRITE()

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

<h1>My First Web Page</h1>
<p>My first paragraph.</p>

<script>
document.write(5 + 6);
</script>

</body>
</html>
```


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

<h1>My First Web Page</h1>
<p>My first paragraph.</p>

<button type="button"
onclick="document.write(5 + 6)">Try
it</button>

</body>
</html>
```

USING INNERHTLM

- The id attribute defines the HTML element. The innerHTML property defines the HTML content:

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

<h1>My First Web Page</h1>
<p>My First Paragraph.</p>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML =
5 + 6;
</script>

</body>
</html>
```

USING CONSOLE.LOG()

- In your browser, you can use the console,log() method to display data.
- Activate the browser with F12, and select “Console” in the menu.

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

<h1>My First Web Page</h1>
<p>My first paragraph.</p>

<p>
Activate debugging in your browser (Chrome,
IE, Firefox) with F12, and select "Console"
in the debugger menu.
</p>

<script>
console.log(5 + 6);
</script>

</body>
</html>
```

JAVASCRIPT PROGRAMS

- The program instructions are called **statements.**
- JavaScript is a programming language.
- JavaScript statements are separated by semicolons.

```
<script>
var x = 5;
var y = 6;
var z = x + y;
document.getElementById("demo").innerHTML =
z;
</script>
```

JAVASCRIPT STATEMENTS

- javaScript statements are composed of:
- Values, Operators, Expressions, Keywords and Comments.
- **JavaScript Values**
- Defines two types of values:
- Fixed values and variable values
- Fixed values are called literals.
- Variable values called variables.

JAVASCRIPT LITERALS

- Numbers are written with or without decimals:
- 10.50
- Strings are test, written within double or single quotes:
- “Hello” or ‘Hello’

JAVASCRIPT VARIABLES

- Variables are used to store data values
- JavaScript uses the **var** keyword to declare variables
- An **equal sign** is used to assign values to variables.
- var a;
- a= 8;

OPERATORS

- Assignment operator (=) to assign values to variables:
- var a=5;
- Var b = 8;
- **Arithmetic operators** (+ - * /) to compute values:
- (5+8)*10;
- The values can be of various types, such as numbers and strings
- “John”+ “ ”+ “Doe”

ASSIGNMENT OPERATORS

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulus
++	Increment
--	Decrement

Operator	Example	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

COMPARISON AND LOGICAL OPERATORS

Operator	Description
==	equal to
===	equal value and equal type
!=	not equal
!==	not equal value or not equal type
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to
?	ternary operator

JOINING MULTIPLE WORDS

- Historically, programmers have used three ways of joining multiple words into one variable name:
- **Hyphens:** first-name, last-name, master-cars, inter-city
- **Underscore:** first_name, last_name, master_cars,inter_city
- **Camel Case:** FirstName, LastName, MasterCard, InterCity
- **In JavaScript, camel case often starts with a lowercase letter:**
- **firstName, lastName, masterCard, interCity**

DATA TYPES

- In programming, text values are called text string.
- String are written inside double or single quotes. Numbers are without quote.
- If you put a number in quotes, it will be treated as a text string.
- var a= “Hello", carName = “World”; //string
- var price = 200; // number
- var x={firstName: “John”, lastName: “Doe”} // Objects

JAVASCRIPT COMMENTS

- Comments are ignored, and will not be exacuted:

  var a =5; //I will be executed

  //var a = 7; I will not be executed

- /* and */
- JavaScript is case sensitive
- All JavaScript identifiers are case sensitive:
- The variables **lastName** and **lastname,** are two different variables.
- lastName= “Zhanerke”;
- Lastname = “Serik” ;

FUNCTION

- function is a series of code instructions you group together and give a name. They can perform a series of operations.
- You need to create them only once, and you can reuse them over and over again.

FUNCTION

- A function is a programming routine consisting of one or more lines of code that performs a certain task.
- alert () function requires one piece of information, known as an argument in programmer-speak. In this case, that piece of information is the text you want the alert text box to display.
- To display some information you put text inside single apostrophe quotes (‘) or double quotation mark(“).

DECLARING A FUNCTION

- Start by deciding what your function is going to do(like display an alert message) then choose a suitable name for it.
- Names can’t have any spaces or special characters.
- Start with the word **function**
- Parentheses is to send extra information to your function

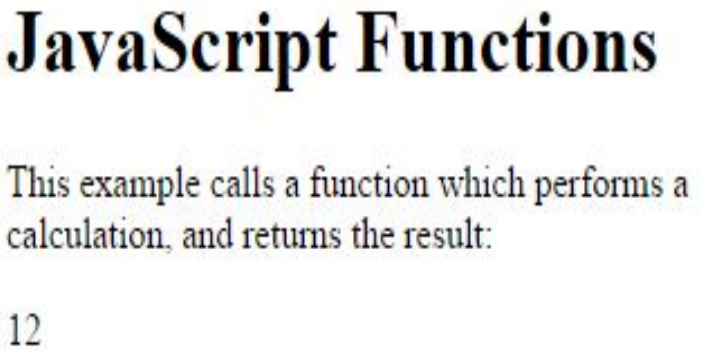
```
<script>
  function ShowAlertBox() {
  alert("I'm a function.")
  }
</script>
```

CALLING A FUNCTION

- To call the function, use the function name followed by parentheses:
- ShowAlertBox()

EXAMPLE

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

<h1>JavaScript Functions</h1>

<p>This example calls a function which
performs a calculation, and returns the
result:</p>

<p id="demo"></p>

<script>
function myFunction(p1, p2) {
    return p1 * p2;
}
document.getElementById("demo").innerHTML =
myFunction(3, 4);
</script>

</body>
</html>
```



FUNCTIONS WITH PARAMETERS

- The code to be executed, by the function, is placed inside curly brackets: {}

THANK YOU!