UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Centre de la Imatge i la Tecnologia Multimèdia

JavaScript for Beginners

Bachelor's Degree in Video Game Design and Development

Summary

- Introduction
- JS and HTML
- Syntax
- Basics
- Operators
- Flow control
- Functions
- Events
- Object oriented programming
- Arrays
- Additional resources

Introduction

- **JavaScript programming language**
  - **Appears in 1995 with the name *LiveScript***
  - **Lightweight and interpreted programming language with object oriented capabilities**
  - **Designed for creating network-centric applications, and commonly used as a part of web pages**
    - **The core of the language is embedded in the most popular web browsers**
  - **Complementary to and integrated with Java and HTML**
  - **Open and cross-platform language**

https://www.javascript.com/

JS and HTML

- JavaScript code within <script> … </script> HTML tags in web pages
- The script tag takes 2 attributes
  - Language: specifies the scripting language (recent versions of HTML do not use this attribute anymore)
  - Type: it is the attribute now recommended to indicate the scripting language in use

```
<html>
   <body>
      <script language="javascript" type="text/javascript">
         <!--
            document.write("Hello World!")
         //-->
      </script>
   </body>
</html>
```

JS and HTML

- **Can be placed within**
  - **Header of the HTML file <head>…</head>**
    - **When you want to have the script running on some event (for instance, when the user clicks somewhere)**

```
<html>

   <head>

      <script type="text/javascript">
         <!--
            function sayHello() {
               alert("Hello World")
            }
         //-->
      </script>

   </head>

   <body>
      <input type="button" onclick="sayHello()" value="Say Hello" />
   </body>

</html>
```

JS and HTML

- **Can be placed within**
  - **The body section <body>…</body>**
    - **When you want to have the script running when the webpage loads (in this case you would not have any function)**

```
<html>

   <head>
   </head>

   <body>

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

      <p>This is web page body </p>

   </body>
</html>
```

JS and HTML

- **Can be placed within**
  - **Both, the header and the body section**

```
<html>
   <head>
      <script type="text/javascript">
         <!--
            function sayHello() {
               alert("Hello World")
            }
         //-->
      </script>
   </head>

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

      <input type="button" onclick="sayHello()" value="Say Hello" />

   </body>
</html>
```


JS and HTML

- **Can be placed within**
  - **An external file (.js) that loads the HTML file**

```
<html>

   <head>
      <script type="text/javascript" src="filename.js" ></script>
   </head>

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

example.html

```
function sayHello() {
   alert("Hello World")
}
```

file.js

Syntax

- JavaScript ignores spaces, tabs, newlines… indent your code as you want!

- Semicolons after each statement are optional
  - But you must use them for multiple instructions in a single line

```
<script language="javascript" type="text/javascript">
   <!--
      var1 = 10
      var2 = 20
   //-->
</script>
```

```
<script language="javascript" type="text/javascript">
   <!--
      var1 = 10; var2 = 20;
   //-->
</script>
```

Syntax

- JavaScript is a case-sensitive language
  - Be careful with capital letters (Time and time are different names!!)
- Comments
  - C and C++ style comments are supported
  - JavaScript code can be put between html comments
    <!-- … -->, just in case the browser does not support JavaScript.
    In this case, the closing tag must be preceded by //

```
<script language="javascript" type="text/javascript">
   <!--

      // This is a comment. It is similar to comments in C++

      /*
      * This is a multiline comment in JavaScript
      * It is very similar to comments in C Programming
      */

   //-->
</script>
```

Basics

- **Data types**
  - **Numbers: 123, 120.5**
    - No distinction between integers and floats, all numbers represented as floats
  - **Strings: “This is a string”**
  - **Booleans: true or false**
- **Variables**
  - **Declared with the *var* keyword**
  - **No need to tell the type of the variable (as in python)**
  - **There are global and local variables**
  - **Names**
    - First character must be a letter or an underscore
    - The other characters must be letters, underscores of digits
    - They are case-sensitive

```
<script type="text/javascript">
   <!--
      var name = "Ali";
      var money;
      money = 2000.50;
   //-->
</script>
```

```
<body onload = checkscope();>
   <script type = "text/javascript">
      <!--
         var myVar = "global"; // Declare a global variable
         function checkscope( ) {
            var myVar = "local";  // Declare a local variable
            document.write(myVar);
         }
      //-->
   </script>
</body>
```

Basics

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

```
<html>
   <body>
      <script language="javascript" type="text/javascript">
         <!--
            document.write("Hello World!")
         //-->
      </script>
   </body>
</html>
```

- **Input**
  - **Using HTML forms**
- **Other features**
  - **Scopes specified with { }**

Input example

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

          if( document.myForm.Name.value == "" )
          {
             alert( "Please provide your name!" );
             return false;
          }
          else
          {
              document.write(document.myForm.Name.value)
              return true;
          }
        }
      //-->
    </script>

  </head>

  <body>
    <form name="myForm" onsubmit="return(validate());">
          <tr>
            <td align="right">Name</td>
            <td><input type="text" name="Name" /></td>
          </tr>
    </form>

  </body>
</html>
```

Operators

Arithmetic ops.	+, -, *, /, %, ++, --
Relational ops.	==, !=, <, >, <=, >=
Boolean ops.	!, &&, ||
Bitwise ops.	&, |, ^, ~, <<, >>, >>> (right shift with zero)
Conditional op.	?: (example (a < b) ? a : b)

- **There are also shortcuts for arithmetic and bitwise operators**
  - **+=, -=, *=, /= and %=**
  - **&=, |=, ^=, <<= and >>=**


Flow control

- **If, if…else and if…else if…else**

Syntax

```
if (expression 1){
   Statement(s) to be executed if expression 1 is true
}

else if (expression 2){
   Statement(s) to be executed if expression 2 is true
}

else if (expression 3){
   Statement(s) to be executed if expression 3 is true
}

else{
   Statement(s) to be executed if no expression is true
}
```

Example

```
<html>
   <body>

      <script type="text/javascript">
         <!--
            var book = "maths";
            if( book == "history" ){
               document.write("<b>History Book</b>");
            }

            else if( book == "maths" ){
               document.write("<b>Maths Book</b>");
            }

            else if( book == "economics" ){
               document.write("<b>Economics Book</b>");
            }

            else{
               document.write("<b>Unknown Book</b>");
            }
         //-->
      </script>

      <p>Set the variable to different value and then try...</p>
   </body>
<html>
```

Flow control

- **Switch**

Syntax

```
switch (expression)
{
   case condition 1: statement(s)
   break;

   case condition 2: statement(s)
   break;
   ...

   case condition n: statement(s)
   break;

   default: statement(s)
}
```

Example

```
<html>
   <body>

      <script type="text/javascript">
         <!--
            var grade='A';
            document.write("Entering switch block<br />");
            switch (grade)
            {
               case 'A': document.write("Good job<br />");
               break;

               case 'B': document.write("Pretty good<br />");
               break;

               case 'C': document.write("Passed<br />");
               break;

               case 'D': document.write("Not so good<br />");
               break;

               case 'F': document.write("Failed<br />");
               break;

               default:  document.write("Unknown grade<br />")
            }
            document.write("Exiting switch block");
         //-->
      </script>

      <p>Set the variable to different value and then try...</p>
   </body>
</html>
```

Flow control

- **While**

Syntax

```
while (expression){
   Statement(s) to be executed if expression is true
}
```

```
<script type="text/javascript">
   <!--
      var count = 0;
      document.write("Starting Loop ");

      while (count < 10){
         document.write("Current Count : " + count + "<br />");
         count++;
      }

      document.write("Loop stopped!");
   //-->
</script>
```

Example

- **Do… while**

Syntax

```
do{
   Statement(s) to be executed;
} while (expression);
```

```
<script type="text/javascript">
   <!--
      var count = 0;

      document.write("Starting Loop" + "<br />");
      do{
         document.write("Current Count : " + count + "<br />");
         count++;
      }

      while (count < 5);
      document.write ("Loop stopped!");
   //-->
</script>
```

Example

Flow control

- **For**

Syntax

```
for (initialization; test condition; iteration statement){
   Statement(s) to be executed if test condition is true
}
```

```
<script type="text/javascript">
   <!--
      var count;
      document.write("Starting Loop" + "<br />");

      for(count = 0; count < 10; count++){
         document.write("Current Count : " + count );
         document.write("<br />");
      }

      document.write("Loop stopped!");
   //-->
</script>
```

Example

- **For… in**

Syntax

```
for (variablename in object){
   statement or block to execute
}
```

```
<script type="text/javascript">
   <!--
      var aProperty;
      document.write("Navigator Object Properties<br /> ");

      for (aProperty in navigator) {
         document.write(aProperty);
         document.write("<br />");
      }
      document.write ("Exiting from the loop!");
   //-->
</script>
```

Example

Flow control

- **Loop control**
  - **As in C and C++, JavaScript includes the commands *break* and *continue***

```
<script type="text/javascript">
   <!--
   var x = 1;
   document.write("Entering the loop<br /> ");

   while (x < 20)
   {
      if (x == 5){
         break; // breaks out of loop completely
      }
      x = x + 1;
      document.write( x + "<br />");
   }

   document.write("Exiting the loop!<br /> ");
   //-->
</script>
```

Example break

```
<script type="text/javascript">
   <!--
      var x = 1;
      document.write("Entering the loop<br /> ");

      while (x < 10)
      {
         x = x + 1;

         if (x == 5){
            continue; // skip rest of the loop body
         }
         document.write( x + "<br />");
      }

      document.write("Exiting the loop!<br /> ");
   //-->
</script>
```

Example continue

Functions

- **Functions are reusable pieces of code**
  - Defined using *function* keyword
  - May receive parameters
    - Between parenthesis
    - **Type NOT NEEDED!!**
  - May return one value
    - Using the keyword *return*

```
<html>
   <head>

      <script type="text/javascript">
         function concatenate(first, last)
         {
            var full;
            full = first + last;
            return full;
         }

         function secondFunction()
         {
            var result;
            result = concatenate('Zara', 'Ali');
            document.write (result );
         }
      </script>

   </head>

   <body>
      <p>Click the following button to call the function</p>

      <form>
         <input type="button" onclick="secondFunction()" value="Call Function">
      </form>

      <p>Use different parameters inside the function and then try...</p>

  </body>
</html>
```

Events

- **JavaScript interaction with HTML is handled through events**
  - **Events occur when the user or the browser manipulates a page**
  - **Developers can execute JavaScript code when events occur**
    - ***onclick* event: mouse left button clicked**
    - ***onmouseover*: mouse cursor over any element**
    - ***onmouseout*: mouse moved out of that element**
    - **…**

```
<html>
   <head>

      <script type="text/javascript">
         <!--
            function sayHello() {
               alert("Hello World")
            }
         //-->
      </script>

   </head>

   <body>
      <p>Click the following button and see result</p>

      <form>
         <input type="button" onclick="sayHello()" value="Say Hello" />
      </form>

   </body>
</html>
```

Example

Object oriented prog.

- **JavaScript works directly with objects**
  - **Created by using the *new* keyword followed by the constructor method**
  - **User-defined objects inherit from a generic *Object***

```
<html>
   <head>

   <title>User-defined objects</title>

      <script type="text/javascript">
         function book(title, author){
            this.title = title;
            this.author  = author;
         }
      </script>

   </head>
   <body>

      <script type="text/javascript">
         var myBook = new book("Perl", "Mohtashim");
         document.write("Book title is : " + myBook.title + "<br>");
         document.write("Book author is : " + myBook.author + "<br>");
      </script>

   </body>
</html>
```

```
<html>
   <head>
      <title>User-defined objects</title>

      <script type="text/javascript">
         var book = new Object();   // Create the object
         book.subject = "Perl"; // Assign properties to the object
         book.author  = "Mohtashim";
      </script>

   </head>

   <body>

      <script type="text/javascript">
         document.write("Book name is : " + book.subject + "<br>");
         document.write("Book author is : " + book.author + "<br>");
      </script>

   </body>
</html>
```

Object oriented prog.

- **JavaScript works directly with objects**
  - **Properties or attributes are assigned "directly"**
  - **The keyword *this* is used in object methods to refer the object itself**

```
<html>
   <head>

   <title>User-defined objects</title>

      <script type="text/javascript">
         function book(title, author){
            this.title = title;
            this.author  = author;
         }
      </script>

   </head>
   <body>

      <script type="text/javascript">
         var myBook = new book("Perl", "Mohtashim");
         document.write("Book title is : " + myBook.title + "<br>");
         document.write("Book author is : " + myBook.author + "<br>");
      </script>

   </body>
</html>
```

```
<html>
   <head>
      <title>User-defined objects</title>

      <script type="text/javascript">
         var book = new Object();   // Create the object
         book.subject = "Perl"; // Assign properties to the object
         book.author  = "Mohtashim";
      </script>

   </head>

   <body>

      <script type="text/javascript">
         document.write("Book name is : " + book.subject + "<br>");
         document.write("Book author is : " + book.author + "<br>");
      </script>

   </body>
</html>
```

Object oriented prog.

- JavaScript works directly with objects
  - User-defined methods have to be assigned as properties of the object
- JavaScript native objects
  - Number
  - Boolean
  - String
  - Array
  - Date
  - Math
  - RegExp

```
<html>
   <head>
   <title>User-defined objects</title>

      <script type="text/javascript">
         // Define a function which will work as a method
         function addPrice(amount){
            this.price = amount;
         }

         function book(title, author){
            this.title = title;
            this.author  = author;
            this.addPrice = addPrice; // Assign that method as property.
         }
      </script>

   </head>
   <body>

      <script type="text/javascript">
         var myBook = new book("Perl", "Mohtashim");
         myBook.addPrice(100);

         document.write("Book title is : " + myBook.title + "<br>");
         document.write("Book author is : " + myBook.author + "<br>");
         document.write("Book price is : " + myBook.price + "<br>");
      </script>

   </body>
</html>
```


Arrays

- The Array object is a list of strings or integers
- Arrays are created in two different (but equivalent) ways

```
var fruits = new Array( "apple", "orange", "mango" );
```

```
var fruits = [ "apple", "orange", "mango" ];
```

- The elements of the array go from 0 to length-1
- To access the elements of an array [ ] is used

Arrays

- The property length returns the size of the array

```
<html>
   <head>
      <title>JavaScript Array length Property</title>
   </head>

   <body>

      <script type="text/javascript">
         var arr = new Array( 10, 20, 30 );
         document.write("arr.length is : " + arr.length);
      </script>

   </body>
</html>
```

- The object array also offers other useful methods
  - push(), pop(), reverse(), forEach(), indexOf(), sort()…

Additional resources

- **JavaScript webpage**

https://www.javascript.com/

- **JavaScript tutorials**

https://www.javascript.com/learn/javascript/

https://www.w3schools.com/js/

https://www.tutorialspoint.com/javascript

Questions?

www.citm.upc.edu