JAVASCRIPT
- Introduction -

Week VIII

Static Web Programming
Semester 1

Javascript Introduction

- The origin of JavaScript was **LiveScript,** developed in1995 in Netscape Communications.
- Collaboration from **Netscape** and **Sun** (*"Java"* programming language developer) named *"JavaScript"* on 4 December1995.
- The language was recognized on the browser of Netscape Navigator 2.0 version or above, while Microsoft completed Internet Explorer with JavaScript on 3.0 version above.

Javascript Introduction

- **JavaScript** is **a scripting language integrated to HTML code and will be proceed client side (Browser)**, thus an ability of HTML document was enhanced.
    - JavaScript is possible to validate the input in a form before sent to server
    - JavaScript is able to implement the dynamic web and interactive
- Javascript depends on the browser (navigator) which called a webpage contained scripts from Javascript inside the HTML document.
- Javascript does not need a specific compiler or interpreter to execute.
- Javascript is a programming language applied into HTML, Web, computer applications, servers, laptop, tablet, smart phones, etc.

Javascript Introduction

- Javascript is often used to manipulate HTML elements and add Style automatically or simply to create more interactive HTML documents.
- To add a javascript in HTML document, we can use script tag and put in head element or wherever in the body tag area.

Example of JavaScript

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

  //…Script javascript is written here…
  //example:

   document.write("Hello world");
  </script>
</body>
```

Location of JavaScript in HTML Document

- Using <*SCRIPT*> tag
  - <*SCRIPT*> tag is written between tag of <*HEAD*> and </*HEAD*>. To call JavaScript function (a.k.a *event*), put the caller in the body of HTML document between tag of <*BODY*> *and* </*BODY*>.
  - Additional information in the <*SCRIPT*> *tag* shows the language type used and its version, e.g. *"JavaScript", "JavaScript1.1", "JavaScript1.2"* for JavaScript language.

Location of JavaScript in HTML Document

- Example:

```
<HTML>
<HEAD>
       <TITLE>Example of Javascript</TITLE>
</HEAD>
<BODY>
       <SCRIPT language="Javascript">
       <!--
              alert("Hello !");
       // -->
       </SCRIPT>
</BODY>
</HTML>
```

Location of JavaScript in HTML Document

- Using external file
  - It use a JavaScript code writing in a textfile and will be called from an HTML document (specific for Netscape 3.0 above).

```
<SCRIPT LANGUAGE="Javascript"
src="url/file.js"> </SCRIPT>
```

Where url/file.js is the filename contains JavaScript code, if the src attribute is not included then the Script tag will find the code inside the tag of Script.

Location of JavaScript in HTML Document

- Using particular event
  - Event is an action done by user, e.g. mouse click.

```
<tag eventHandler="Javascript
code written">
```

Where **eventHandler** is the name of the event.

JavaScript as OOP

- JavaScript treats the elements as an **object**, means that the element:
  - Classified based on the hierarchy thus we know where the specific location of the object.
  - Associated with the condition or the properties

JavaScript as OOP

- Illustration:

  Garden

  - □Tree
    - Branch
      - o Leaves
      - o Bird Nest
        - □ Length = 20 cm
        - □ Color= Yellow
        - □ Height = 4 cm
    - Trunk
    - Root
  - □Cage
    - Chicken
    - Duck
- Bird's nest which is on the tree can be written as follows :

  Garden.Tree.Branch.Bird_Nest
- If we want to paint or change the color of the bird nest on the tree, the instruction will be:

  Garden.Tree.Branch.Bird_Nest.Color=green

Property

- Property is an attribute of an object.
- Syntax (separated by ".") :

  object_name . property_name
- Property may have a value, syntax:

  object . property = value

Property

- Example:

```
<HTML>
<HEAD>
<TITLE>Property defaultStatus</TITLE>
</HEAD>
<BODY>
<H1>Test of defaultStatus</H1>
<SCRIPT LANGUAGE = "JavaScript">
<!--
  window.defaultStatus = “Studying javascript “ ;
//-->
</SCRIPT>
</BODY>
</HTML>
```

Property name

Value

Object name

Method

- Method is a set of code used to take an action for an object.
- Syntax (separated by “.”) :

  object_name . method_name(“parameter”)

- Example:

```
<HTML>
<HEAD>
<TITLE>Javascript</TITLE>
</HEAD>
<BODY>
<SCRIPT language="Javascript">
<!--
   document.write(“Trying a JavaScript <BR>”):
   document.write(“Good Luck !”);
// -->
</SCRIPT>
</BODY>
</HTML>
```

Parameter

Object name

Method name

JavaScript Can Change HTML Element

- HTML DOM (**D**ocument **O**bject **M**odel) is a W3C standart to access the HTML elements.
- JavaScript can manipulate DOM (change the content of HTML).
- The following code is an example to change the content (innerHTML) of an HTML element defined by id="demo":

```
document.getElementById("demo").innerHTML =
"Hello JavaScript";
```

Example

```
<h1>JavaScript</h1>
<p>JavaScript can change the HTML element:</p>
<button type="button"
  onclick="myFunction()">Click Me!</button>
<p id="demo">this is just a demonstration.</p>

<script>
function myFunction() {
   document.getElementById("demo").innerHTML =
  "Welcome to Javascript!";
}
</script>
```

JavaScript Can Change HTML Element

- Method **document.getElementById()** is one of many methods in HTML DOM.
- JavaScript can also be used to:
  - Change HTML elements
  - Remove HTML elements
  - Create new HTML elements
  - Copy and duplicate HTML elements
  - etc…

JavaScript to Change HTML Attribute

- The following code is used to change the value of *source* (src) attribute from <img> HTML element :

Web Sample

JavaScript to Change Style HTML (CSS)

- Changing the style of an HTML element, is a variant of changing an HTML attribute.
- Using JavaScript, almost all CSS value could be changed.

Web Sample

JavaScript to Validate Data

- JavaScript is often used to validate input.

Web Sample

JavaScript Syntax

- JavaScript is a **scripting** language.
- JavaScript **syntax** is the set of rules, how JavaScript programs are constructed.
- Scripting language is a lightweight programming language.
- In a programming language, the program instructions are called **statements**.

JavaScript Statements

- JavaScript statements are composed of:
  - Values,
  - Expressions,
  - Operators,
  - Variables,
  - Keywords, and
  - Comments.

JavaScript Values (Literals)

- In computer science, **fixed values** called as **literal**.
- Using JavaScript, the most important rules to write **fixed values** (constant) are:
  - **Number Literal** is written with or without decimals, or written with scientific notation (e):
  - Example:

    3.14, 1001, 123e5

JavaScript Values (Literals)

- In computer science, **fixed values** called as **literal**.
- Using JavaScript, the most important rules to write **fixed values** (constant) are:
  - **String Literal** is written within double or single quotes:
  - Example:

    “Informatics”, ‘Informatics’

JavaScript Values (Literals)

- In computer science, **fixed values** called as **literal**.
- Using JavaScript, the most important rules to write **fixed values** (constant) are:
  - **Expression Literal i**s a combination of values, variables, and operators, which computes to a value.
  - Used to evaluate value.
  - The computation is called an evaluation.
  - Example:

    5 + 6, 5 * 10

JavaScript Variable

- In a programming language, **variables** are used to **store** data values.
- JavaScript uses the **var** keyword to **define** variables.
- An **equal sign** is used to **assign values** to variables (as in algebra).

JavaScript Variable

- In this example, x is defined as a variable.
- Then, x is assigned (given) the value 6:

```
var x;
x = 6;
```

- A literal is a **fixed** value.
- A variable is a **name**.
- A variable may have **variabel value**.

JavaScript Variable (Identifier)

- All JavaScript **variables** must be **identified** with **unique names**.
- These unique names are called **identifiers**.
- Identifiers can be short names (like x and y), or more descriptive names (age, sum, totalVolume).
- The general rules for constructing names for variables (unique identifiers) are:
  - Names can contain letters, digits, underscores, and dollar signs.
  - Names must begin with a letter
  - Names can also begin with $ and _
  - Names are case sensitive (y and Y are different variables)
  - Reserved words (like JavaScript keywords) cannot be used as names

JavaScript Operators

- JavaScript used 4 arithmetic operators and a modulus operator to calculate numbers:

No	Operator	Symbol	Example
1	Multiply	*	5 * 4 = 20
2	Divide	/	20 / 5 = 4
3	Sum	+	5 + 4 = 9
4	Substract	-	5 – 4 = 1
+	Modulus	%	10 % 4 = 2

Rules of JavaScript Syntax

- All instructions in JavaScript are **case sensitive**.
- Variable **lastName** and **lastname** are 2 different variables.
- JavaScript does not read **VAR** or **Var** as an instruction word **var**.

JavaScript Keywords

Keyword	Description
break	Terminates a switch or a loop
continue	Jumps out of a loop and starts at the top
debugger	Stops the execution of JavaScript, and calls (if available) the debugging function
do ... while	Executes a block of statements, and repeats the block, while a condition is true
for	Marks a block of statements to be executed, as long as a condition is true
break	Terminates a switch or a loop
for ... in	Marks a block of statements to execute for each element in an object (or array)

JavaScript Keywords

Keyword	Description
function	Declares a function
if ... else	Marks a block of statements to be executed, depending on a condition
return	Exits a function
switch	Marks a block of statements to be executed, depending on different cases
try ... catch	Implements error handling to a block of statements
var	Declares a variable
function	Declares a function
while	Marks a block of statements to be executed when the condition is true

Assignment Operators

- In JavaScript, an equal (=) is an assignment operator, not "equal to" operator
- This is different with algebra.
- The following code will not change in algebra:

```
x = x + 5
```

Assignment Operators

- In JavaScript, this is significantly influence and will change the value, means to assign the operation result of x + 5 and stored to variable x.
- Operator of "equal to" in JavaScript is written using == or ===.

Data Type in JavaScript

- Variables in JavaScript are able to handle many **data types**: numbers, strings, arrays, objects etc:

```
var length = 16; // number literal

var points = x * 10; // expression literal

var lastName = "Johnson"; // string literal

var cars = ["Saab", "Volvo", "BMW"]; // array
                    literal

var x = {firstName:"John", lastName:"Doe"}; //
                    object literal
```