



UTM
UNIVERSITI TEKNOLOGI MALAYSIA



**Online
Learning**

JavaScript Fundamentals

Part 1

Sarina Sulaiman
Faculty of Computing

Outline

- Introduction
- Fundamental of JavaScript
- Javascript events management
- DOM and Dynamic HTML (DHTML)

Introduction - Outline

- Introduction
 - What is Javascript?
 - What Javascript can do?
 - Examples usage of Javascript
 - How to use Javascript?

What is Javascript

- Official name: ECMAScript maintain by ECMA organisations
- ECMA 262 - official Javascript standard based on Javascript (Netscape) & Jscript (Microsoft)
- Invented by Brendan Eich at Netscape (with Navigator 2.0)
- Development is still in progress!

What is Javascript

- Java and Javascript is not the SAME - only similar to Java and C++
- The fundamentals of Javascript are similar to Java and/or C++

What is Javascript?

- Was designed to add interactivity to HTML pages
- Is a scripting language
- An interpreted language (means that scripts execute without preliminary compilation)
- Case-sensitive
- Must be embedded into HTML
- Browser dependent
- Execute whenever the HTML doc. which contain the script open by browser.
- Everyone can use JavaScript without purchasing a license

What Javascript can do?

- JavaScript gives HTML designers a programming tool
- JavaScript can put dynamic text into an HTML page
- JavaScript can react to events
- JavaScript can read and write HTML elements
- JavaScript can be used to validate data
- JavaScript can be used to detect the visitor's browser
- JavaScript can be used to create cookies

Examples usage of Javascript - Activity 01

- Tools/plugin
 - Clocks
 - Mouse Trailers (an animation that follows your mouse when you surf a site)
 - Date Picker
- RIA/SPA

Examples usage of Javascript

- **Event management**
- **Form management & verification**
- **Dynamic HTML (DHTML)**
- **Client-Server application - AJAX**

How to use Javascript?

- Activity 02

- Inside the head tag (head section)
- Within the body tag (body section)
- In an external file (external script)

How to use Javascript?

- head section

```
<html>
<head>
<script type="text/javascript">
function message()
{
    alert("This alert box was called with the onload event");
}
</script>
</head>

<body onload="message()">

</body>
</html>
```

How to use Javascript?

- body section

```
<html>
<head>
</head>

<body>

<script type="text/javascript">
document.write("This message is written by JavaScript");
</script>

</body>
</html>
```

How to use Javascript?

- external script

```
<html>
<head>
<script src="myjs.js">
</script>
</head>
<body>
<input type="button" onclick="popup()" value="Click Me!">
</body>
</html>
```

```
// JavaScript Document (myjs.js)
function popup() {
alert("Hello World")
}
```

How to use Javascript?

- All code inside head tag (except in function) will be executed first before the html <body> load
- Put global variables in the <head> section script
- Use document.write/ln to send html output
- Use window.prompt to get user input

Use in alert box

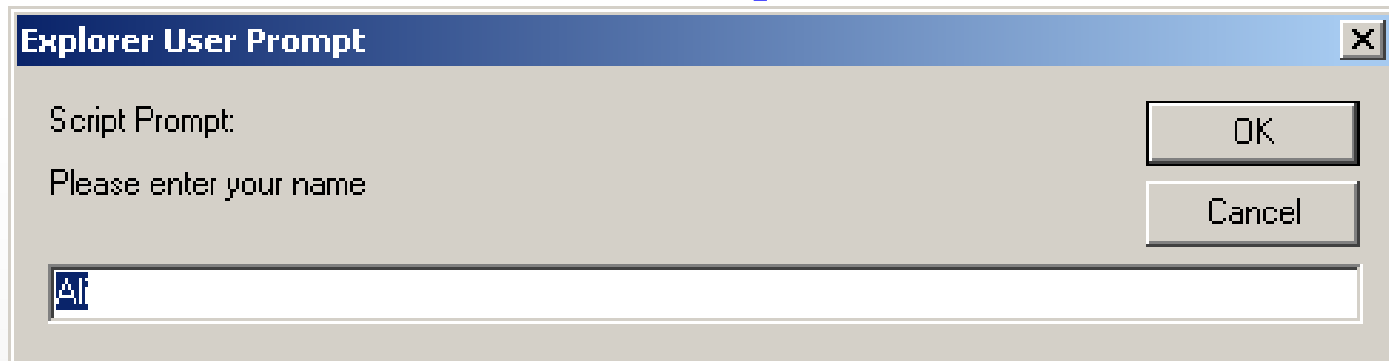
Escape sequence	Description
<code>\n</code>	Newline. Position the screen cursor at the beginning of the next line.
<code>\t</code>	Horizontal tab. Move the screen cursor to the next tab stop.
<code>\r</code>	Carriage return. Position the screen cursor to the beginning of the current line; do not advance to the next line. Any characters output after the carriage return overwrite the characters previously output on that line.
<code>\\</code>	Backslash. Used to represent a backslash character in a string.
<code>\"</code>	Double quote. Used to represent a double quote character in a string contained in double quotes. For example, <pre> window.alert("\"in quotes\""); </pre> displays "in quotes" in an alert dialog.
<code>\'</code>	Single quote. Used to represent a single quote character in a string. For example, <pre> window.alert('\'in quotes\''); </pre> displays 'in quotes' in an alert dialog.

Debugging Errors

- Activity 03

Simple JS application – input/output

– Activity 04



Javascript Fundamental

- A light Java/C++
- All other things are more or less the same:
 - Keyword, variables
 - Operator
 - Conditional statement
 - Looping etc.
- Case sensitive
- No strong typing in JS for variable

Javascript Variables – Activity 05

- Variables name – case sensitive
- No typing!
- Can change type during execution
 - Activity 05 - a
- Use double quote for character and string variable
- Cannot use reserve word for variable name!

Javascript Fundamental

```
<html>
<head>
<script type="text/javascript">
var i;
i = 10;
document.write("<br />" + i);
i = "rosely";
document.write("<br />" + i);
i = 2 + "5";
document.write("<br />" + i);
</script>
</head>
<body>
</body>
</html>
```

OUTPUT:

10
rosely
25

JS – Reserve Word

JavaScript Keywords				
break	case	continue	delete	do
else	false	for	function	if
in	new	null	return	switch
this	true	typeof	var	void
while	with			
<i>Keywords that are reserved, but not used by JavaScript</i>				
catch	class	const	debugger	default
enum	export	extends	finally	import
super	try			

Javascript variables operation

- Arithmetic operations – same as Java/C++
- + operators is overloaded, can be used for string
- Number + string (or vice versa), result string
 - [Activity 05 - b](#)
 - $A = 2 + 5$ (result 7)
 - $A = 2 + \text{"5"}$ (result 25)
 - $A = A + 2$ (result 252)

Javascript variables operation

```
<html>
<head>
<script type="text/javascript">
i = 2;
document.write("<br />" + i);
i = 2 + "5";
document.write("<br />" + i);
i = i + 2;
document.write("<br />" + i);
</script>
</head>
<body>
</body>
</html>
```

OUTPUT:

2
25
252

JS Operators

Arithmetic operators:

JavaScript operation	Arithmetic operator	Algebraic expression	JavaScript expression
Addition	+	$f + 7$	$f + 7$
Subtraction	-	$p - c$	$p - c$
Multiplication	*	bm	$b * m$
Division	/	x / y or $x \div y$	x / y
Modulus	%	$r \bmod s$	$r \% s$

Fig. 7.11 Arithmetic operators.

Relational operators:

Standard algebraic equality operator or relational operator	JavaScript equality or relational operator	Sample JavaScript condition	Meaning of JavaScript condition
<i>Equality operators</i>			
=	==	$x == y$	x is equal to y
	!=	$x != y$	x is not equal to y
<i>Relational operators</i>			
>	>	$x > y$	x is greater than y
<	<	$x < y$	x is less than y
	>=	$x >= y$	x is greater than or equal to y
≤	<=	$x <= y$	x is less than or equal to y

JS Operators

Precedence and Associativity:

Operators	Associativity	Type
()	left to right	parentheses
* / %	left to right	multiplicative
+ -	left to right	additive
< <= > >=	left to right	relational
== !=	left to right	equality
=	right to left	assignment

Fig. 7.16 Precedence and associativity of the operators discussed so far.

Javascript - Conditional expressions - Activity 06

- If else, switch statement – same as C++/Java
- Boolean
 - Value 0, false = false
 - Value 1, true = true
- String comparison, use the quote!
 - if (password == “007”)
- Check the example!

Javascript - Conditional expressions

- Special conditional operator
 - `variablename=(condition)?value1:value2`
 - If true value1 assigned else value2 assigned to variablename

`greeting = (visitor == "PRES") ? "Dear President " : "Dear ";`

Javascript - Loop - Activity 07

- for loop, while loop – same as C++/Java
- Use break statement to exit loop
- JavaScript For...In Statement
 - used to loop (iterate) through the elements of an array or through the properties of an object.
 - `var mycars = new Array();`
 - `for (x in mycars)`
 - `document.write(mycars[x] + "
");`
 - Activity 07

THANK YOU