Pemrograman Web

5. Javascript (lanjut)

M. Udin Harun Al Rasyid, S.Kom, Ph.D
http://lecturer.eepis-its.edu/~udinharun
udinharun@eepis-its.edu

Table of Contents

Comparison Operators

- Given that x=5;

Operator	Description	Comparing	Returns
==	is equal to	x==8	*false*
		x==5	*true*
===	is exactly equal to (value and type)	x==="5"	*false*
		x===5	*true*
!=	is not equal	x!=8	*true*
!==	is not equal (neither value or type)	x!=="5"	*true*
		x!==5	*false*
>	is greater than	x>8	*false*
<	is less than	x<8	*true*
>=	is greater than or equal to	x>=8	*false*
<=	is less than or equal to	x<=8	*true*

Logical Operators

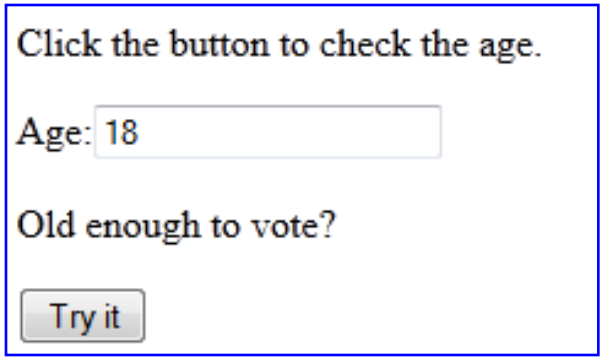
- Given that x=6 and y=3

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

Conditional Operator

- Syntax:
  variablename=(*condition*)?*value1*:*value2*

```
<html>
<body>

<p>Click the button to check the age.</p>

Age:<input id="age" value="18" />
<p>Old enough to vote?</p>
<button onclick="myFunction()">Try it</button>

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

<script>
function myFunction()
{
var age,voteable;
age=document.getElementById("age").value;
voteable=(age<18)?"Too young":"Old enough";
document.getElementById("demo").innerHTML=voteable;
}
</script>

</body>
</html>
```



if ... else statement

- JavaScript supports conditional statements which are used to perform different actions based on different conditions.

- JavaScript supports following forms of if..else statement:
  - if statement
  - if...else statement
  - if...else if... statement.

if statement:

- Syntax:

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

```
<html>
<body>

<script type="text/javascript">
<!--
var age = 20;
if( age > 18 ){
   document.write("<b>Qualifies for driving</b>");
}
//-->
</script>
</body>
</html>
```

Qualifies for driving

if...else statement:

- Syntax:

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

```
<html>
<body>
<script type="text/javascript">
<!--
var age = 15;

if( age > 18 ){
   document.write("<b>Qualifies for driving</b>");
}else{
   document.write("<b>Does not qualify for driving</b>");
}
//-->
</script>
</body>
</html>
```

Does not qualify for driving

if...else if... statement:

- 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
}
```

```
<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>

</body>
</html>
```

Maths Book

Switch Case

- The basic syntax of the **switch** statement is to give an expression to evaluate and several different statements to execute based on the value of the expression. The interpreter checks each **case** against the value of the expression until a match is found. If nothing matches, a **default** condition will be used.
- The **break** statements indicate to the interpreter the end of that particular case.

```
switch (expression)
{
  case condition 1: statement(s)
                    break;
  case condition 2: statement(s)
                    break;
   ...
  case condition n: statement(s)
                    break;
  default: statement(s)
}
```

```
<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>


</body>
</html>
```

Entering switch block
Good job
Exiting switch block

```
<html>
<body>

<script type="text/javascript">
<!--
var grade='A';
document.write("Entering switch block<br />");
switch (grade)
{
  case 'A': document.write("Good job<br />");
  case 'B': document.write("Pretty good<br />");
  case 'C': document.write("Passed<br />");
  case 'D': document.write("Not so good<br />");
  case 'F': document.write("Failed<br />");
  default:  document.write("Unknown grade<br />")
}
document.write("Exiting switch block");
//-->
</script>

</body>
</html>
```

Entering switch block
Good job
Pretty good
Passed
Not so good
Failed
Unknown grade
Exiting switch block

while Loops

- The purpose of a **while** loop is to execute a statement or code block repeatedly as long as *expression* is true. Once expression becomes *false*, the loop will be exited.

- **Syntax:**

  while (expression){

  Statement(s) to be executed if expression is true

  }

```
<html>
<body>

<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>

</body>
</html>
```

Starting LoopCurrent Count : 0
Current Count : 1
Current Count : 2
Current Count : 3
Current Count : 4
Current Count : 5
Current Count : 6
Current Count : 7
Current Count : 8
Current Count : 9
Loop stopped!

Math Object

- The Math object allows you to perform mathematical tasks.
- Syntax:

  var x = Math.PI; // Returns PI
  var y = Math.sqrt(16); // Returns the square root of 16

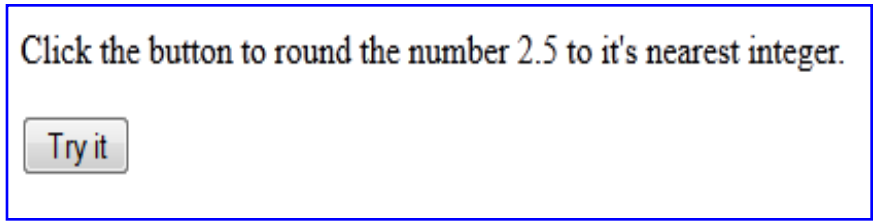
- Math Object Properties

Property	Description
E	Returns Euler's number (approx. 2.718)
LN2	Returns the natural logarithm of 2 (approx. 0.693)
LN10	Returns the natural logarithm of 10 (approx. 2.302)
LOG2E	Returns the base-2 logarithm of E (approx. 1.442)
LOG10E	Returns the base-10 logarithm of E (approx. 0.434)
PI	Returns PI (approx. 3.14)
SQRT1_2	Returns the square root of 1/2 (approx. 0.707)
SQRT2	Returns the square root of 2 (approx. 1.414)

Math Object Methods

Method	Description
abs(x)	Returns the absolute value of x
acos(x)	Returns the arccosine of x, in radians
asin(x)	Returns the arcsine of x, in radians
atan(x)	Returns the arctangent of x as a numeric value between -PI/2 and PI/2 radians
atan2(y,x)	Returns the arctangent of the quotient of its arguments
ceil(x)	Returns x, rounded upwards to the nearest integer
cos(x)	Returns the cosine of x (x is in radians)
exp(x)	Returns the value of E^x
floor(x)	Returns x, rounded downwards to the nearest integer
log(x)	Returns the natural logarithm (base E) of x
max(x,y,z,...,n)	Returns the number with the highest value
min(x,y,z,...,n)	Returns the number with the lowest value
pow(x,y)	Returns the value of x to the power of y
random()	Returns a random number between 0 and 1
round(x)	Rounds x to the nearest integer
sin(x)	Returns the sine of x (x is in radians)
sqrt(x)	Returns the square root of x
tan(x)	Returns the tangent of an angle

```
<html>
<body>

<p id="demo">Click the button to round the number 2.5 to it's nearest integer.</p>

<button onclick="myFunction()">Try it</button>

<script>
function myFunction()
{
document.getElementById("demo").innerHTML=Math.round(2.5);
}
</script>

</body>
</html>
```



Cookies

- Web Browser and Server use HTTP protocol to communicate and HTTP is a stateless protocol.

- But for a commercial website it is required to maintain session information among different pages. For example one user registration ends after completing many pages.

- Using cookies is the most efficient method of remembering and tracking preferences, purchases, commissions, and other information required for better visitor experience or site statistics.

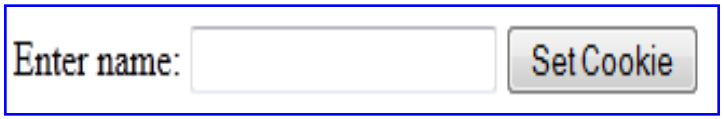
Storing Cookies:

- The simplest way to create a cookie is to assign a string value to the *document.cookie* object, which looks like this:
- Syntax:

  document.cookie = "key1=value1;key2=value2;expires=date";

- JavaScript *escape()* function to encode the *value* before storing it in the cookie.
- *unescape()* function when you read the cookie value.

```
<html>
<head>
<script type="text/javascript">
<!--
function WriteCookie()
{
   if( document.myform.customer.value == "" ){
      alert("Enter some value!");
      return;
   }

   cookievalue= escape(document.myform.customer.value) + ";";
   document.cookie="name=" + cookievalue;
   alert("Setting Cookies : " + "name=" + cookievalue );
}
//-->
</script>
</head>
<body>
<form name="myform" action="">
Enter name: <input type="text" name="customer"/>
<input type="button" value="Set Cookie" onclick="WriteCookie();"/>
</form>
</body>
</html>
```



Browser Detection

- The Navigator object contains all information about the visitor's browser

```
<html>
<body>
<div id="example"></div>

<script>

txt = "<p>Browser CodeName: " + navigator.appCodeName + "</p>";
txt+= "<p>Browser Name: " + navigator.appName + "</p>";
txt+= "<p>Browser Version: " + navigator.appVersion + "</p>";
txt+= "<p>Cookies Enabled: " + navigator.cookieEnabled + "</p>";
txt+= "<p>Platform: " + navigator.platform + "</p>";
txt+= "<p>User-agent header: " + navigator.userAgent + "</p>";

document.getElementById("example").innerHTML=txt;

</script>

</body>
</html>
```

Browser CodeName: Mozilla

Browser Name: Netscape

Browser Version: 5.0 (Windows)

Cookies Enabled: true

Platform: Win32

User-agent header: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:15.0) Gecko/20100101 Firefox/15.0.1

Form Validation

- JavaScript can be used to validate data in HTML forms before sending off the content to a server.
  - has the user left required fields empty?
  - has the user entered a valid e-mail address?
  - has the user entered a valid date?
  - has the user entered text in a numeric field?

Required Fields

```
<html>
<head>
<script>
function validateForm()
{
var x=document.forms["myForm"]["fname"].value;
if (x==null || x=="")
  {
  alert("First name must be filled out");
  return false;
  }
}
</script>
</head>

<body>
<form name="myForm" action="demo_form.asp" onsubmit="return validateForm()" method="post">
First name: <input type="text" name="fname">
<input type="submit" value="Submit">
</form>
</body>

</html>
```

E-mail Validation

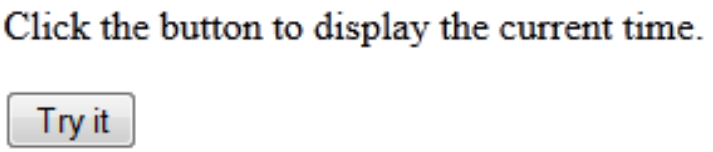
- The input data must contain an @ sign and at least one dot (.). Also, the @ must not be the first character of the email address, and the last dot must be present after the @ sign, and minimum 2 characters before the end.

```
<html>
<head>
<script>
function validateForm()
{
var x=document.forms["myForm"]["email"].value;
var atpos=x.indexOf("@");
var dotpos=x.lastIndexOf(".");
if (atpos<1 || dotpos<atpos+2 || dotpos+2>=x.length)
  {
  alert("Not a valid e-mail address");
  return false;
  }
}
</script>
</head>

<body>
<form name="myForm" action="demo_form.asp" onsubmit="return validateForm();" method="post">
Email: <input type="text" name="email">
<input type="submit" value="Submit">
</form>
</body>

</html>
```

Timing Events

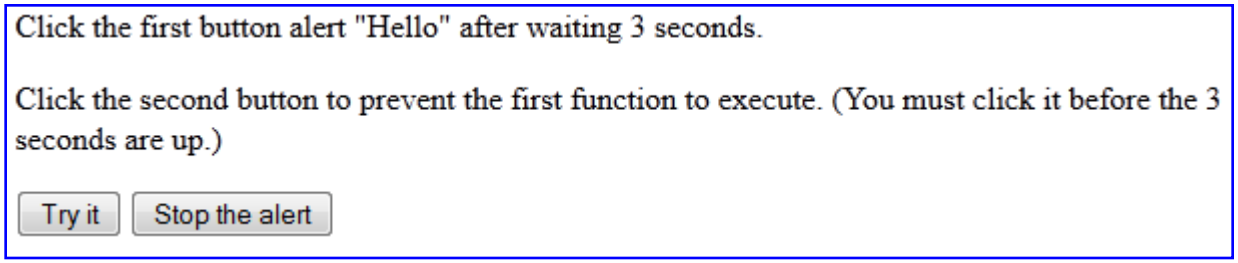
- To execute some code at specified time-intervals.
  - setInterval() - executes a function, over and over again, at specified time intervals
  - setTimeout() - executes a function, once, after waiting a specified number of milliseconds

```
<html>
<body>

<p>Click the button to display the current time.</p>
<button onclick="myFunction()">Try it</button>

<script>
function myFunction()
{
setInterval(function(){myTimer()},1000);
}

function myTimer()
{
var d=new Date();
var t=d.toLocaleTimeString();
document.getElementById("demo").innerHTML=t;
}
</script>

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

</body>
</html>
```



Stop the Execution

```
<html>
<body>

<p>Click the first button alert "Hello" after waiting 3 seconds.</p>
<p>Click the second button to prevent the first function to execute. (You must click it before the 3 seconds are up.)</p>
<button onclick="myFunction()">Try it</button>
<button onclick="myStopFunction()">Stop the alert</button>

<script>
var myVar;

function myFunction()
{
myVar=setTimeout(function(){alert("Hello")},3000);
}

function myStopFunction()
{
clearTimeout(myVar);
}
</script>

</body>
</html>
```



Javascript animation

- You can use JavaScript to create a complex animation such as Object movements, etc.

- JavaScript can be used to move a number of DOM elements (<img />, or any other HTML element) around the page according to some sort of pattern determined by a logical equation or function.

- JavaScript can also set a number of attributes of a DOM object including its position on the screen. You can set *top* and *left* attribute of an object to position it anywhere on the screen. Here is the simple syntax:

```
// Set distance from left edge of the screen.
object.style.left = distance in pixels or points;

or
// Set distance from top edge of the screen.
object.style.top = distance in pixels or points;
```

```
<html>
<head>
<title>JavaScript Animation</title>
<script type="text/javascript">
<!--
var imgObj = null;
function init(){
   imgObj = document.getElementById('myImage');
   imgObj.style.position= 'relative';
   imgObj.style.left = '0px';
}
function moveRight(){
   imgObj.style.left = parseInt(imgObj.style.left) + 10 + 'px';
}
window.onload =init;
//-->
</script>
</head>
<body>
<form>
<img id="myImage" src="/images/html.gif" />
<p>Click button below to move the image to right</p>
<input type="button" value="Click Me" onclick="moveRight();" />
</form>
</body>
</html>
```

Manual Animation:

- Using JavaScript function *getElementById()* to get a DOM object and then assigning it to a global variable *imgObj*.
- Defined an initialization function *init()* to initialize *imgObj* where we have set its *position* and *left* attributes.
- Calling initialization function at the time of window load.
- Finally, Calling *moveRight()* function to increase left distance by 10 pixels. You could also set it to a negative value to move it to the left side.

```
<html>
<head>
<title>JavaScript Animation</title>
<script type="text/javascript">
<!--
var imgObj = null;
var animate ;
function init(){
   imgObj = document.getElementById('myImage');
   imgObj.style.position= 'relative';
   imgObj.style.left = '0px';
}
function moveRight(){
   imgObj.style.left = parseInt(imgObj.style.left) + 10 + 'px';
   animate = setTimeout(moveRight,20); // call moveRight in 20msec
}
function stop(){
   clearTimeout(animate);
   imgObj.style.left = '0px';
}
window.onload =init;
//-->
</script>
</head>
<body>
<form>
<img id="myImage" src="/images/html.gif" />
<p>Click the buttons below to handle animation</p>
<input type="button" value="Start" onclick="moveRight();" />
<input type="button" value="Stop" onclick="stop();" />
</form>
</body>
</html>
```

Automated Animation

- Automate how an image moves to right by using JavaScript function *setTimeout()*
- The *moveRight()* function is calling *setTimeout()* function to set the position of *imgObj*.
- Added a new function *stop()* to clear the timer set by *setTimeout()* function and to set the object at its initial position.

```
<html>
<head>
<title>Rollover with a Mouse Events</title>
<script type="text/javascript">
<!--
if(document.images){
    var image1 = new Image();      // Preload an image
    image1.src = "/images/html.gif";
    var image2 = new Image();      // Preload second image
    image2.src = "/images/http.gif";
}
//-->
</script>
</head>
<body>
<p>Move your mouse over the image to see the result</p>
<a href="#" onMouseOver="document.myImage.src=image2.src;"
            onMouseOut="document.myImage.src=image1.src;">
<img name="myImage" src="/images/html.gif" />
</a>
</body>
</html>
```

Rollover with a Mouse Event

- The *onMouseOver* event handler is triggered when the user's mouse moves onto the link, and the *onMouseOut* event handler is triggered when the user's mouse moves away from the link (image).

- When the mouse is moved away from the link, the initial image html.gif will reappear on the screen.

Attractive Hover Menu using CSS & JavaScript

```
<html>
<style type="text/css">

#coolmenu{
border: 1px solid black;
width: 170px;
background-color: #E6E6E6;
}

#coolmenu a{
font: bold 13px Verdana;
padding: 2px;
padding-left: 4px;
display: block;
width: 100%;
color: black;
text-decoration: none;
border-bottom: 1px solid black;
}

html>body #coolmenu a{ /*Non IE rule*/
width: auto;
}

#coolmenu a:hover{
background-color: black;
color: white;
}

</style>

<body>

<div id="coolmenu">
<a href="http://www.eepis-its.edu">Website EEPIS EDU</a>
<a href="http://lecturer.eepis-its.edu">Web Dosen</a>
<a href="http://student.eepis-its.edu">Web mahasiswa</a>
<a href="http://jas.eepis-its.edu">Job Arrangement System</a>
<a href="http://it.eepis-its.edu">Web IT</a>
</div>
</body>
</html>
```

- 100% CSS based hover menu

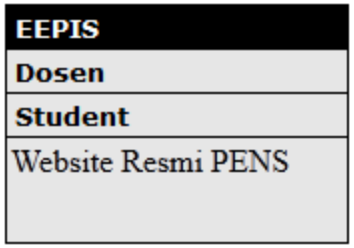
Website EEPIS EDU
Web Dosen
Web mahasiswa
Job Arrangement System
Web IT

Attractive Hover Menu using CSS & JavaScript

```
<html>
<head>
<style type="text/css">

#coolmenu{
border: 1px solid black;
width: 170px;
background-color: #E6E6E6;
}

#coolmenu a{
font: bold 13px Verdana;
padding: 2px;
padding-left: 4px;
display: block;
width: 100%;
color: black;
text-decoration: none;
border-bottom: 1px solid black;
}

html>body #coolmenu a{ /*Non IE rule*/
width: auto;
}

#coolmenu a:hover{
background-color: black;
color: white;
}

#tabledescription{
width: 100%;
height: 3em;
padding: 2px;
filter:alpha(opacity=0);
-moz-opacity:0;
}

</style>
```

```
</style>

<script type="text/javascript">

var baseopacity=0

function showtext(thetext){
if (!document.getElementById)
return
textcontainerobj=document.getElementById("tabledescription")
browserdetect=textcontainerobj.filters? "ie" : typeof textcontainerobj.style.MozOpacity=="string"?
"mozilla" : ""
instantset(baseopacity)
document.getElementById("tabledescription").innerHTML=thetext
highlighting=setInterval("gradualfade(textcontainerobj)",50)
}

function hidetext(){
cleartimer()
instantset(baseopacity)
}

function instantset(degree){
if (browserdetect=="mozilla")
textcontainerobj.style.MozOpacity=degree/100
else if (browserdetect=="ie")
textcontainerobj.filters.alpha.opacity=degree
else if (document.getElementById && baseopacity==0)
document.getElementById("tabledescription").innerHTML=""
}

function cleartimer(){
if (window.highlighting) clearInterval(highlighting)
}

function gradualfade(cur2){
if (browserdetect=="mozilla" && cur2.style.MozOpacity<1)
cur2.style.MozOpacity=Math.min(parseFloat(cur2.style.MozOpacity)+0.2, 0.99)
else if (browserdetect=="ie" && cur2.filters.alpha.opacity<100)
cur2.filters.alpha.opacity+=20
else if (window.highlighting)
clearInterval(highlighting)
}

</script>
```

```
</head>

<body>

<div id="coolmenu">
<a href="http://www.eepis-its.edu" onMouseover="showtext('Website Resmi PENS')" onMouseout="hidetext()">
EEPIS</a>
<a href="http://lecturer.eepis-its.edu" onMouseover="showtext('Website Dosen')" onMouseout="hidetext()">
Dosen</a>
<a href="http://student.eepis-its.edu" onMouseover="showtext('Website Student')" onMouseout="hidetext()">
Student</a>

<div id="tabledescription"></div>
</div>
</body>
</html>
```



Finish