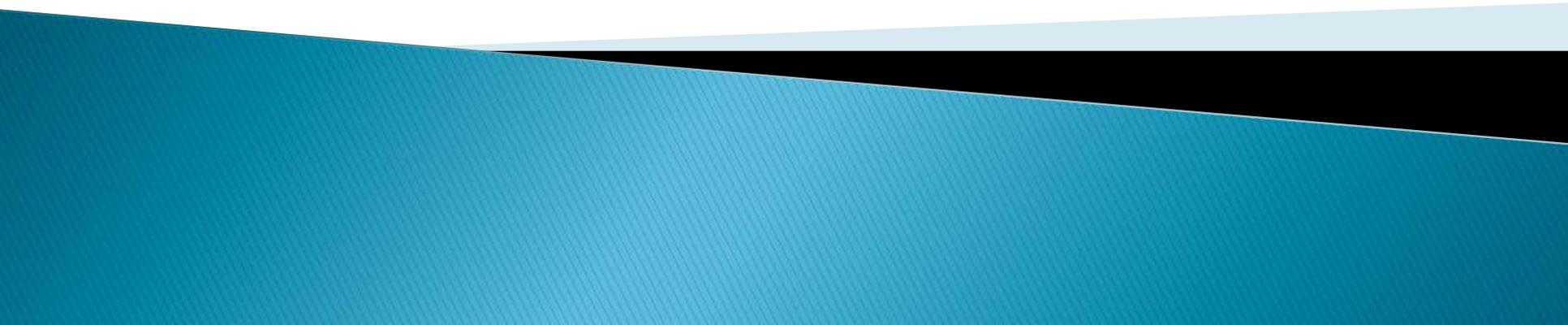


Introduction to Javascript

Chris Panayiotou



Introduction

- ▶ JavaScript's role on the Web
 - JavaScript Programming Language
 - Developed by Netscape for use in Navigator Web Browsers
 - Purpose → make web pages (documents) more dynamic and interactive
 - Change contents of document, provide forms and controls, animation, control web browser window, etc.

Introduction

- ▶ JavaScript was designed to add interactivity to HTML pages
- ▶ JavaScript is a scripting language i.e. lightweight programming language
- ▶ A JavaScript is usually embedded directly in HTML pages
- ▶ JavaScript is an interpreted language (means that scripts execute without preliminary compilation)
 - Converts code to executable format **each time it runs**
 - Converted when browser **loads a web document**
- ▶ JavaScript is supported by all major browsers
- ▶ Java and Javascript are **not the same** although they share some common syntax

What can a JavaScript Do?

- ▶ Gives HTML designers a **programming tool**
 - HTML authors are normally not programmers, but JavaScript is a scripting language with a *very simple syntax!*
 - Almost anyone can put small "snippets" of code into their HTML pages
- ▶ Puts **dynamic text** into an HTML page
 - A JavaScript statement like this:
document.write("<h1>" + name + "</h1>")
can write a variable text into an HTML page
- ▶ React to **events**
 - A JavaScript can be set to execute when something happens, like when a page has finished loading or when a user clicks on an HTML element
- ▶ Can **read and write HTML elements**
- ▶ Can be used to **validate data** (e.g. form data) before it is submitted to a server

Elements of JavaScript

- ▶ Typical programming elements:
 - variables, operators, expressions, statements, definition of functions
- ▶ Connection to Web
 - built-in functions (e.g., Date())
 - **Document Object Model** = names of “things”, properties, methods (e.g., document.write)
 - events (e.g., onSubmit, onMouseOver)

JavaScript and HTML pages

- ▶ Inserting JavaScript into an HTML page

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

- ▶ Note that semicolons are **optional**

JavaScript and HTML pages

- ▶ JavaScript is usually placed in the HEAD section
 - Make sure that the function is loaded before used

```
<html>  
  <head>  
    <script type="text/javascript">  
      some statements  
    </script>  
  </head>  
  <body>  
    Some HTML  
  </body>
```

- ▶ You can place any number of scripts in one page
- ▶ Can write code to a file and load the file

```
<html>  
  <head>  
    <script src="xxx.js"></script>  
  </head>  
  <body> </body>  
</html>
```

JavaScript Variables

- ▶ Used to store information
- ▶ Variable name are case sensitive
- ▶ Must begin with a letter or the underscore
 - Example:
`var strname = "Chris"`
Or
`strname = "Chris"`
- ▶ The scope of variables is as usual
 - Local within the functions
 - Global (page level) if declared outside the functions

JavaScript Operators

Operator	Description	Example	Result
+	Addition	x=2 x+2	4
-	Subtraction	x=2 5-x	3
*	Multiplication	x=4 x*5	20
/	Division	15/5 5/2	3 2.5
%	Modulus (division remainder)	5%2 10%8 10%2	1 2 0
++	Increment	x=5 x++	x=6
--	Decrement	x=5 x--	x=4

Arithmetic operators

Operator	Example	Is The 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

Assignment Operators

JavaScript Operators

Operator	Description	Example
==	is equal to	5==8 returns false
!=	is not equal	5!=8 returns true
>	is greater than	5>8 returns false
<	is less than	5<8 returns true
>=	is greater than or equal to	5>=8 returns false
<=	is less than or equal to	5<=8 returns true

Comparison

Operator	Description	Example
&&	and	x=6, y=3 (x < 10 && y > 1) returns true
	or	x=6, y=3 (x==5 y==5) returns false
!	not	x=6, y=3 !(x==y) returns true

Logical

JavaScript General Syntax

- ▶ Writing formatted text

```
<html>
```

```
<body>
```

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

```
document.write("<h1>Hello World!</h1>")
```

```
</script>
```

```
</body>
```

```
</html>
```

JavaScript General Syntax

- ▶ As in Java

- ▶ Example:

```
txt1="What a very"  
txt2="nice day!"  
txt3=txt1 + " " + txt2
```

- ▶ txt3 contains "What a very nice day!"

- ▶ Creating functions

```
function myfunction(argument1,argument2,etc) {  
    some statements  
}
```

- ▶ Functions with returning values

```
function result(a,b) {  
    c=a+b  
    return c  
}
```

- ▶ Calling Functions

```
myfunction(argument1,argument2,etc)
```

Concatenation

Functions

JavaScript Flow Control

```
<script type="text/javascript">  
// If the time on your browser  
// is less than 12, you will get  
// a "Good morning" greeting.
```

```
var d=new Date()  
var time=d.getHours()  
if (time<10) {  
    document.write("<b>Good  
morning</b>")  
} else {  
    document.write("<b>Good  
afternoon</b>")  
}  
</script>
```

If and If..Else
statement

- ▶ while
while (condition) {
 code to be executed
}

- ▶ do...while
do {
 code to be executed
} while (condition)

- ▶ for
for (initial; condition; increment) {
 code to be executed
}

Loops

JavaScript Flow Control

▶ Switch

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

```
//You will receive a different greeting based on what day it is. Note that Sunday=0,  
//Monday=1, Tuesday=2, etc.
```

```
var d=new Date()  
theDay=d.getDay()  
switch (theDay) {  
  case 5:  
    document.write("Finally Friday")  
    break  
  case 6:  
    document.write("Super Saturday")  
    break  
  case 0:  
    document.write("Sleepy Sunday")  
    break  
  default:  
    document.write("I'm looking forward to this weekend!")  
}  
</script>
```

JavaScript Events

```
<html> <head> <title> JS1 </title> </head>  
<body>
```

```
<a href="http://www.google.com "  
onMouseOver="status= 'Happy Googling'; return true;">  
Google search </a>  
</body>  
</html>
```

This lets the
hyperlink take
place.

This is an
event.

Note single quotes
inside double quotes

Simple Input

```
var input = prompt("What is your name?", "Chris");  
print("Well hello " + (input || "dear"));
```

- ▶ If the user presses 'Cancel' or closes the prompt dialog in some other way without giving a name, the variable `input` will hold the value `null` or `""`
 - Both of these would give `false` when converted to a boolean
 - The expression `input || "dear"` can in this case be read as 'the value of the variable `input`, or else the string `"dear"`'
 - It is an easy way to provide a 'fallback' value.

Error Handling

- ▶ Similar to Java → `try {...} catch (error) {...} finally {...}` statements

```
function lastElement(array) {
  if (array.length > 0)
    return array[array.length - 1]
  else
    throw "Can not take the last element of an empty array."
}
function lastElementPlusTen(array) {
  return lastElement(array) + 10
}
...
try {
  print(lastElementPlusTen([]));
} catch (error) {
  print("Something went wrong: ", error)
} finally {
  print("Finished")
}
```

Manipulating HTML Page Elements

- ▶ Document Object Model (DOM)
 - Contains all HTML elements loaded in a web page
 - Can be used by scripting languages to manipulate HTML on the fly
 - DOM objects:

Anchor (hyperlink)	Select
Body	Style
Button	Submit
Checkbox	Table
Document	TableData
Form	TableHeader
Frame	TableRow
Frameset	Text
Image	Textarea
Link	Window
Location	
Password	
Radio	

Element Manipulation: Anchor (Changing URL and text)

```
<html>
<head>
  <script type="text/javascript">
    function myHref() {
      document.getElementById('myAnchor').innerText="Visit W3Schools"
      document.getElementById('myAnchor').href="http://www.w3schools.com"
    }
  </script>
</head>
<body>
  <a id="myAnchor" href="http://www.microsoft.com">Visit Microsoft</a>
  <form>
    <input type="button" onclick="myHref()" value="Change URL and text">
  </form>
</body>
</html>
```

Element Manipulation: Forms

```
<html>
<head>
<script type="text/javascript">
function showFormElements(theForm) {
    str="Form Elements: "
    for (i=0; i<theForm.length; i++)
        str+="\n "+theForm.elements[i].name
    alert(str)
}
</script>
</head>
```

```
<body>
<form>
First name: <input type="text"
            name="fname" size="20">
<br />
Last name: <input type="text"
              name="lname" size="20">
<br /><br />
<input type="button" name="button1"
value="Display name of form elements"
onClick="showFormElements(this.form)">
</form>
</body>
</html>
```



Element Manipulation: Submitting Forms

```
<html>
<head>
<script type="text/javascript">
function formSubmit() {
    document.forms.myForm.submit()
}
</script>
</head>
<body>
<form name="myForm" action="sample2.asp" method="get">
    Firstname: <input type="text" name="firstname" size="20"><br/>
    Lastname: <input type="text" name="lastname" size="20"><br/><br/>
<input type="button" onclick="formSubmit()" value="Submit">
</form>
</body>
</html>
```

Element Manipulation: Validating Forms

```
<html>
<head>
<script type="text/javascript">
function validate() {
    x=document.myForm
    at=x.email.value.indexOf("@")
    if (at == -1) {
        alert("Not a valid e-mail")
        return false
    }
}
</script>
</head>
```

```
<body>
<form name="myForm"
    action="tryjs_submitpage.htm"
    onsubmit="return validate()">
Enter your E-mail:
<input type="text" name="email"
    size="20">
<input type="submit" value="Submit">
</form>
<p><b>Note:</b> This example
    ONLY tests if the e-mail address
    contains an "@" character. A "real-
    life" code would have to test for
    punctuations, spaces and other
    things as well.</p>
</body>
</html>
```

Javascript Resources

- ▶ <http://www.javascript.com>
 - ▶ <http://www.pageresource.com/jscript/>
 - ▶ <http://www.htmlgoodies.com/beyond/javascript/index.php>
 - ▶ <http://javascript.internet.com/>
 - ▶ <http://javascriptkit.com/>
 - ▶ <http://www.w3schools.com/js/default.asp>
 - ▶ Use Google
- 

Practice 1

- ▶ Create a page with JavaScript to do the following. These can all be on one page.
 - Prompt the user for their name.
 - Use a pop-up box to welcome the user by name.
 - Display the current date on the page in the following format: October 30, 2000. Do not display the time. Do not "hard code" the date; if I load the page tomorrow, I should get a different date than if I load it today.
 - Display the last modified date of the document.
 - Put some useful advice, on any subject, in the status line of the browser.
 - Display a message saying Good Morning if it is in the morning, Good Afternoon if it is in the afternoon, and Good Evening if it is in the Evening.

Practice 2

▶ JavaScript Form Validation Exercise

- Create a form that includes 2 textboxes (one for name, one for email address), a textarea, a dropdown box (<select> tag), and a set of checkboxes. Format the form using a table. The ACTION attribute of the form tag should be to email the form results to your email address. Feel free to use a preexisting form rather than creating one for this exercise.
 - Add JavaScript event handlers to the form to validate the form before it is emailed to you. If there is a problem, display the problem with the page in some way. Possible ways to do this are:
 - a pop-up box
 - displaying the problems in a new window in red text and having the user use the back button to go back to the original page
 - displaying the problems in the current window at the top of the page in red text
 - Validate that an entry has been made in each form element
 - Validate that the email address contains an @ sign.
- ▶ If you have time and want to try something more, add a second drop down list and make this second one dynamic, i.e. make the choices that appear in the second drop down list depend on what choice is made in the first list.