JavaScript

Trainer: Dong Tang Binh

06/2011

Warm up - Introductions

- Your role
- Your background, experiences in the subject
- What do you want from this course

Course Objectives

- At the end of the course, you will have acquired sufficient knowledge to write JavaScript code for Dynamic Web Pages

Agenda

– Introduction and First Looks
– Programming with JavaScript
– JavaSript Objects
– Q&A

Course Audience and Prerequisite

- Programming Language Knowledge
- HTML
- CSS (option)

Assessment Disciplines

- Class Participation: 100%
- Assignment: 100%
- Passing Scores: 65%

Duration and Course Timetable

- Course Duration: 6 hrs

References

- **Learning JavaScript (***By Shelley Powers*)
- **www.google.com**, keyword="JavaScript"
- http://www.w3schools.com/jsref/default.asp
- http://www.javascriptkit.com/jsref/
- http://www.hunlock.com/blogs/Essential_Javascript_--_A_Javascript_Tutorial

Setup Environment

- To complete the course, your PC have to install:
  - A Text Editor such as NotePad, NotePad++, EditPlus, Eclipse, ...
  - A Browser such as IE, FireFox, ..

Course Administration

- In order to complete the course you have to:
  - Sign in the Class Attendance List
  - Participate in the course
  - Provide your feedback in the End of Course Evaluation

Introduction and First Looks

Lesson 1

JavaScript Code Example

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
        "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <title>First Example</title>
    <script type="text/javascript">
        alert(“Hello JavaScript!");
    </script>
</head>
<body>
    <h1>This is a first example</h1>
</body>
</html>
```

What is JavaScript?

- What exactly is JavaScript?
- Where is JavaScript in dynamic web page?
- Are JavaScript and Java the same?
- What can a JavaScript do?
- Advantages and Disadvantages of JavaScript?

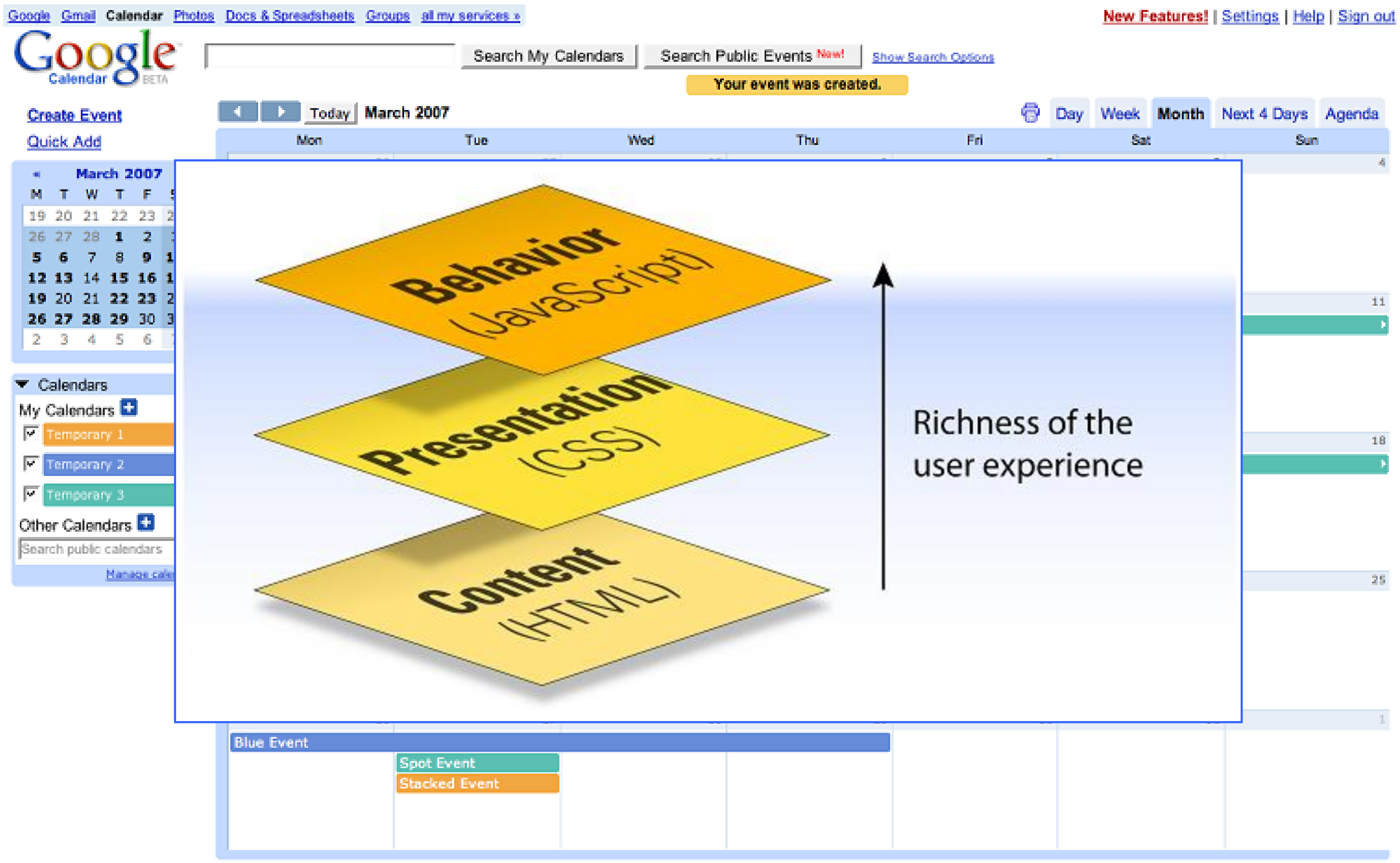
History

- JavaScript was developed by Brendan Eich, then working at Netscape, as a client side scripting language
- Originally the language was called Live Script, but when it was about to be released Java had become immensely popular (and slightly hypey). At the last possible moment Netscape changed the name of its scripting language to “JavaScript”.

JavaScript introduction

- JavaScript was designed to add interactivity to HTML pages
- JavaScript is a scripting language
- A scripting language is a lightweight programming language
- JavaScript is usually embedded directly into HTML pages
- JavaScript is an interpreted language (means that scripts execute without preliminary compilation)
- Everyone can use JavaScript without purchasing a license

Where is JavaScript in a dynamic web page?



Are Java and JavaScript the same?

- NO!
- Java and JavaScript are two completely different languages in both concept and design!
- Java (developed by Sun Microsystems) is a powerful and much more complex programming language - in the same category as C and C++.

What can a JavaScript do?

- **<u>JavaScript gives HTML designers a programming tool</u> -** 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
- **<u>JavaScript can put dynamic text into an HTML page</u> -** A JavaScript statement like this: document.write("<h1>" + name + "</h1>") can write a variable text into an HTML page
- **<u>JavaScript can react to events</u> -** 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

What can a JavaScript do?(cont)

- **JavaScript can read and write HTML elements -** A JavaScript can read and change the content of an HTML element
- **JavaScript can be used to validate data -** A JavaScript can be used to validate form data before it is submitted to a server. This saves the server from extra processing
- **JavaScript can be used to detect the visitor's browser** - A JavaScript can be used to detect the visitor's browser, and - depending on the browser - load another page specifically designed for that browser
- **JavaScript can be used to create cookies** - A JavaScript can be used to store and retrieve information on the visitor's computer

Advantages of JavaScript

- JavaScript is executed on the client side
- JavaScript is a relatively easy language
- JavaScript is relatively fast to the end user
- Extended functionality to web pages

Disadvantages of JavaScript

- Security Issues :
  - JavaScript snippets, once appended onto web pages execute on client servers immediately and therefore can also be used to exploit the user's system.
  - While a certain restriction is set by modern web standards on browsers, malicious code can still be executed complying with the restrictions set.
- JavaScript rendering varies
  - Different layout engines may render JavaScript differently resulting in inconsistency in terms of functionality and interface

Location in HTML

- JavaScript is located in HTML between <script> tags
- Kept in an external JavaScript file and called in HTML code

JavaScript Location Example

/js/script.js

```
alert("This is external JavaScript");
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Tra
        "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <title>JavaScript Location Example</title>
    <script type="text/javascript" src="js/script.js"></script>
    <script type="text/javascript">
        alert("This is inline JavaScript in <head> tag.");
    </script>
</head>
<body>
    <script type="text/javascript">
        alert("This is inline JavaScript in <body> tag.");
    </script>
    <h1>This is a JavaScript Location Example</h1>
</body>
</html>
```

Programming with JavaScript

JavaScript Basic

Variables

- **Introduce with “var”**
  - For global variables (!) and local variables.
  - No “var” for function arguments
- **You do not declare types**
  - Some people say JavaScript is “untyped” language, but really it is “dynamically typed” language
  - JavaScript is *very liberal about converting types*

Variables(cont)

- **There are only two scopes**
  - Global scope
    - Be very careful with this when using Ajax
    - Can cause race conditions.
  - Function (lexical) scope
    - There is *not block scope as in Java*

Operators and Statements

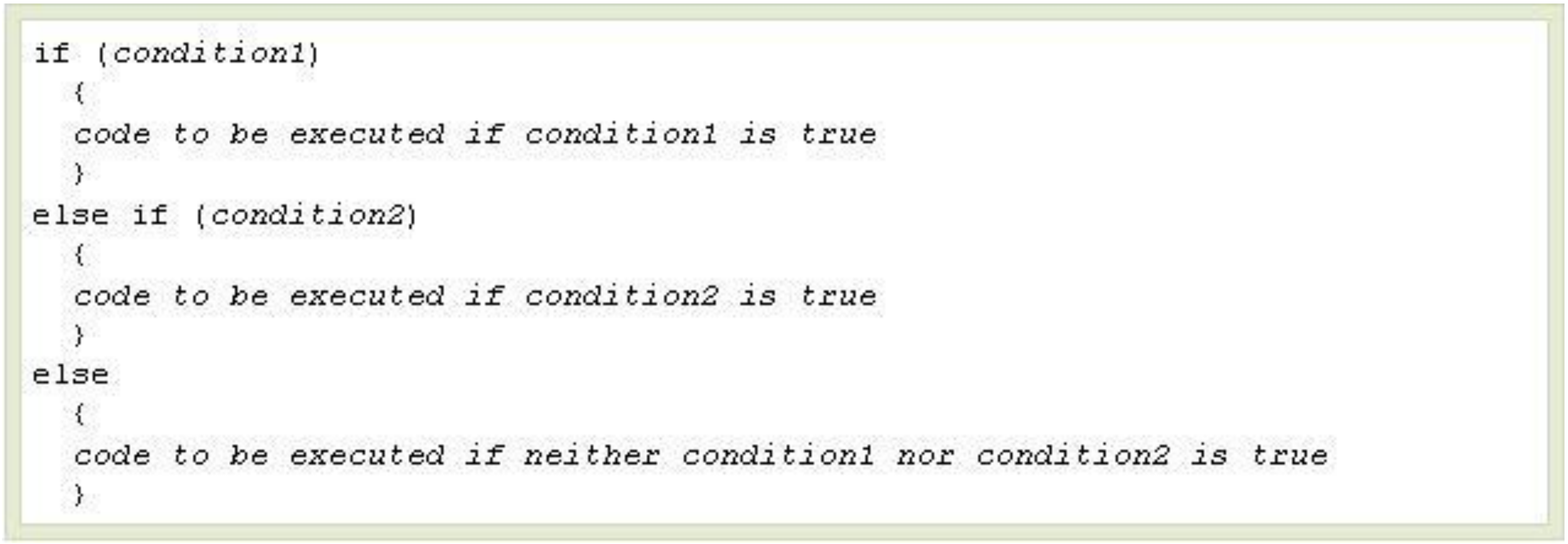
- **Almost same set of operators as Java**
  - + (addition and String concatenation), -, *, /
  - &&, ||, ++, --, etc
  - The == comparison is more akin to Java's "equals“
- **Statements**
  - A JavaScript statement is a command to a browser. The purpose of the command is to tell the browser what to do.
  - Example:
    - document.write("Hello Dolly")**;**
  - **Note:** Using semicolons makes it possible to write multiple statements on one line.

Conditionals

- Conditional statements are used to perform different actions based on different conditions.
- Very often when you write code, you want to perform different actions for different decisions. You can use conditional statements in your code to do this.
- In JavaScript we have the following conditional statements:
  - **if statement** - use this statement to execute some code only if a specified condition is true

Conditionals(cont)

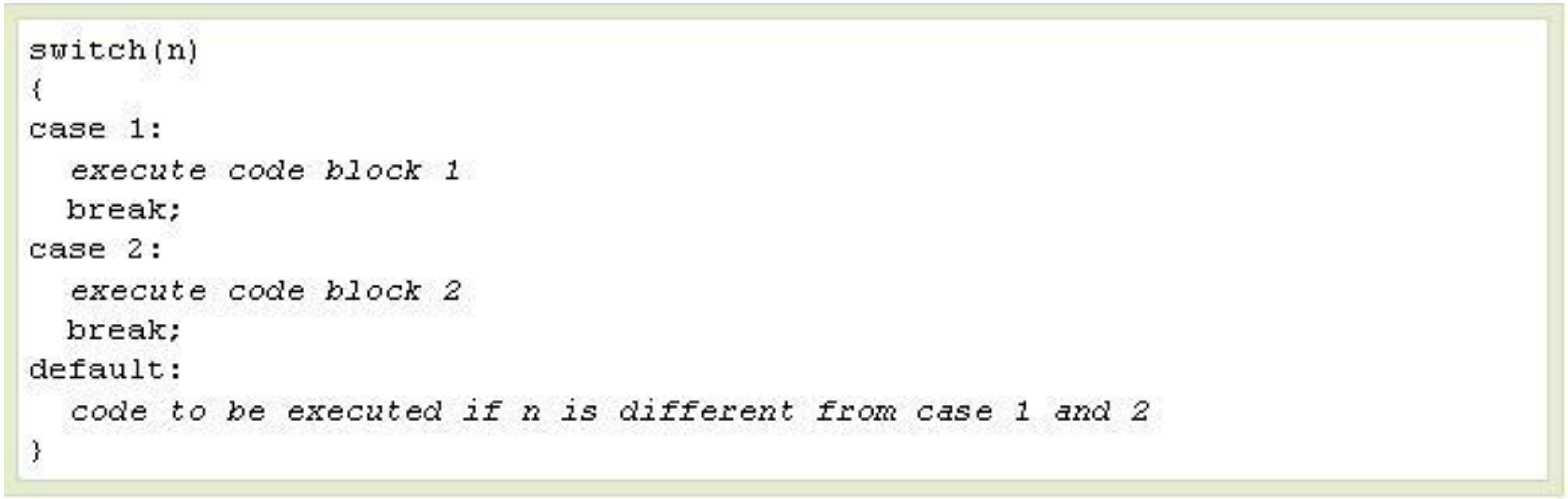
- **if...else statement** - use this statement to execute some code if the condition is true and another code if the condition is false
- **if...else if....else statement** - use this statement to select one of many blocks of code to be executed



```
if (condition1)
  {
  code to be executed if condition1 is true
  }
else if (condition2)
  {
  code to be executed if condition2 is true
  }
else
  {
  code to be executed if neither condition1 nor condition2 is true
  }
```

Conditionals(cont)

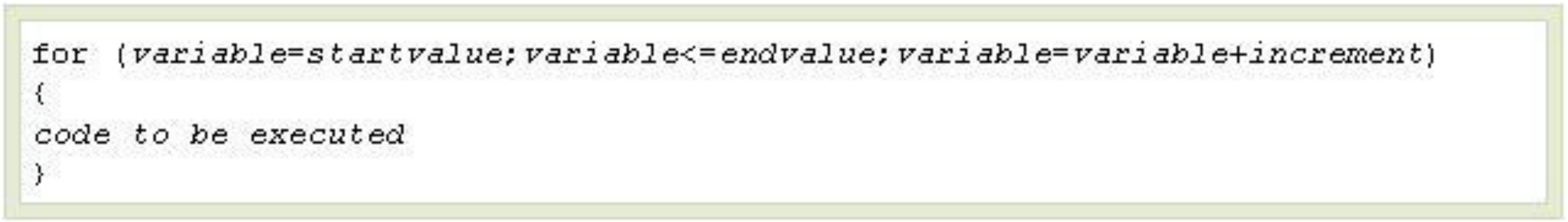
- **Switch statement** - use this statement to select one of many blocks of code to be executed
  - Example:



```
switch(n)
{
case 1:
  execute code block 1
  break;
case 2:
  execute code block 2
  break;
default:
  code to be executed if n is different from case 1 and 2
}
```

Loops

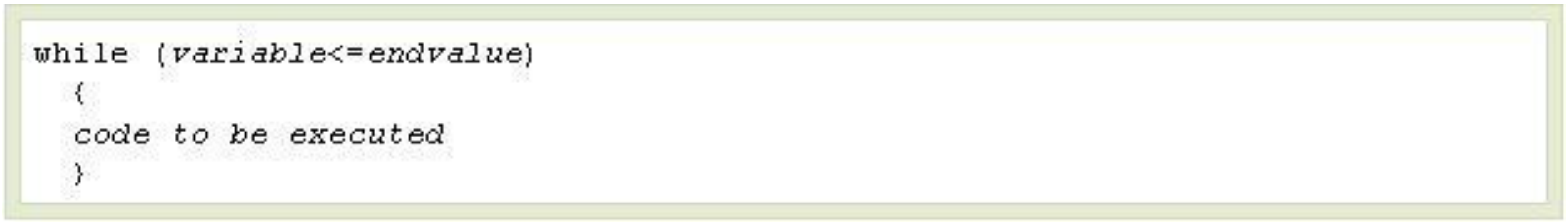
- **The for loop**
  - The for loop is used when you know in advance how many times the script should run
  - Example:



```
for (variable=startvalue;variable<=endvalue;variable=variable+increment)
{
code to be executed
}
```

Loops(cont)

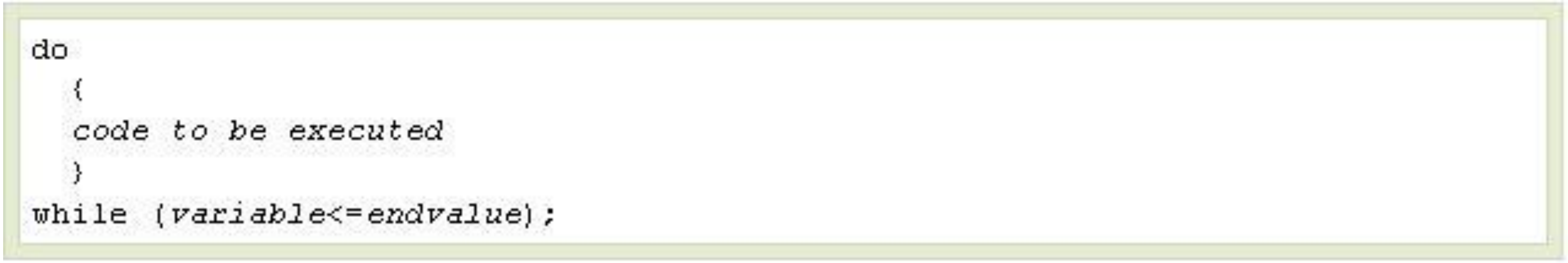
- **The while loop**
  - The while loop loops through a block of code while a specified condition is true.
  - Example:



```
while (variable<=endvalue)
  {
  code to be executed
  }
```

Loops(cont)

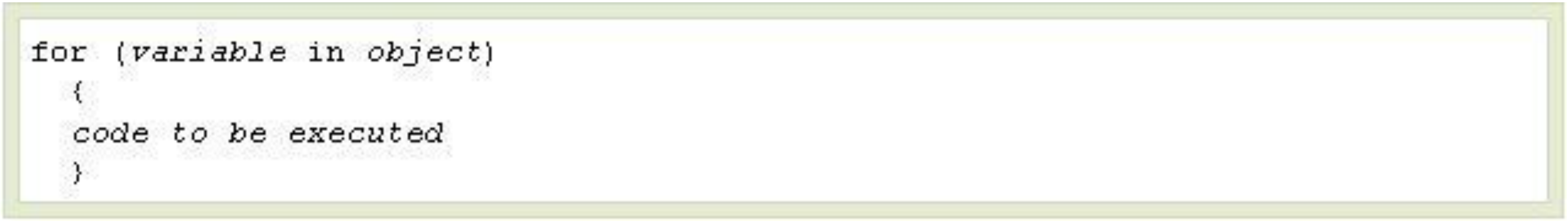
- **The do\ while loop**
  - The do...while loop is a variant of the while loop. This loop will execute the block of code ONCE, and then it will repeat the loop as long as the specified condition is true..
  - Example:



```
do
   {
   code to be executed
   }
while (variable<=endvalue);
```

Loops(cont)

- **The for/in loop**
  - The for...in statement loops through the properties of an object
  - Example



```
for (variable in object)
  {
  code to be executed
  }
```

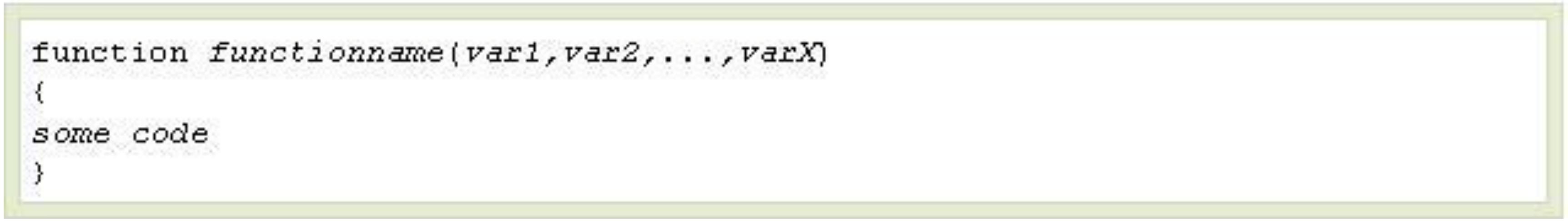
Note: The code in the body of the for...in loop is executed once for each property

Popup Boxes

- JavaScript has three kind of popup boxes: Alert box, Confirm box, and Prompt box.
- Alert box:
  - An alert box is often used if you want to make sure information comes through to the user
  - Syntax: **alert("*sometext*");**
- Confirm box:
  - A confirm box is often used if you want the user to verify or accept something.
  - Syntax: **confirm(“*sometext*”);**
- Prompt box:
  - A prompt box is often used if you want the user to input a value before entering a page.
  - Syntax: **prompt("*sometext*","*defaultvalue*");**

Functions

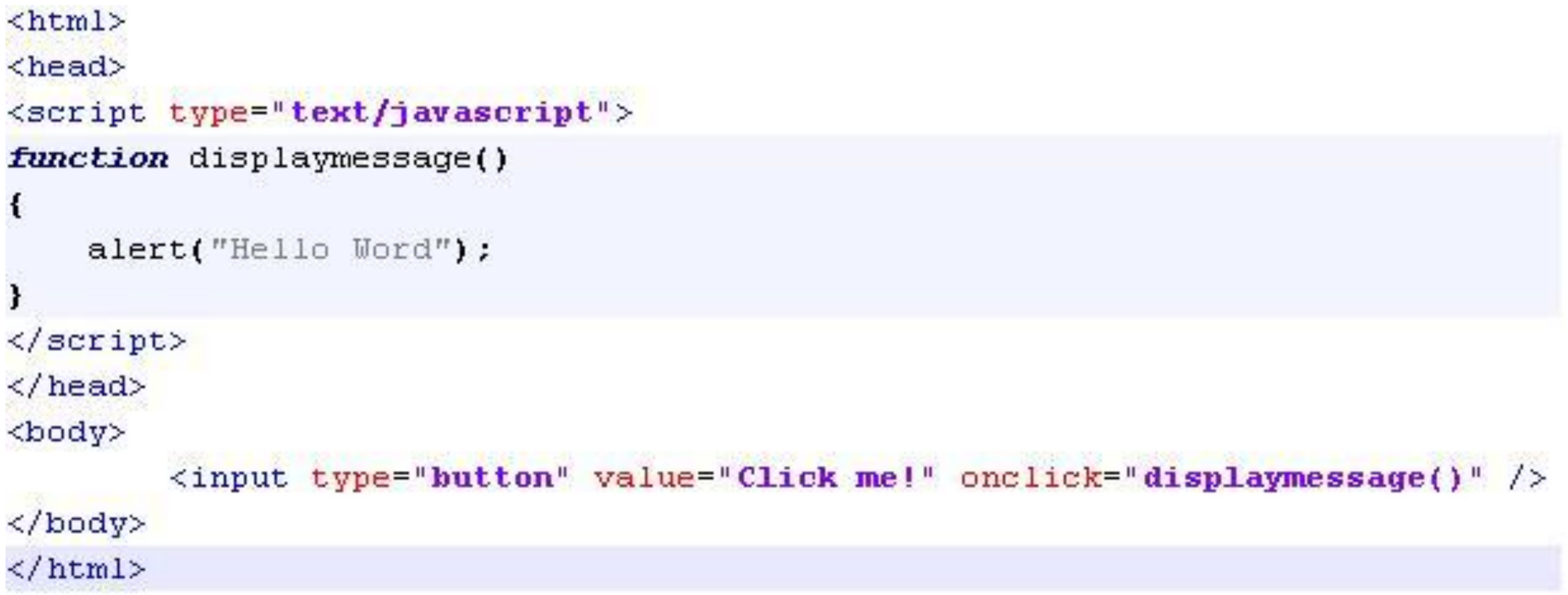
- To keep the browser from executing a script when the page loads, you can put your script into a function.
- A function contains code that will be executed by an event or by a call to the function.
- You may call a function from anywhere within a page (or even from other pages if the function is embedded in an external .js file).
- Syntax:



```
function functionname(var1,var2,...,varX)
{
some code
}
```

Functions (cont)

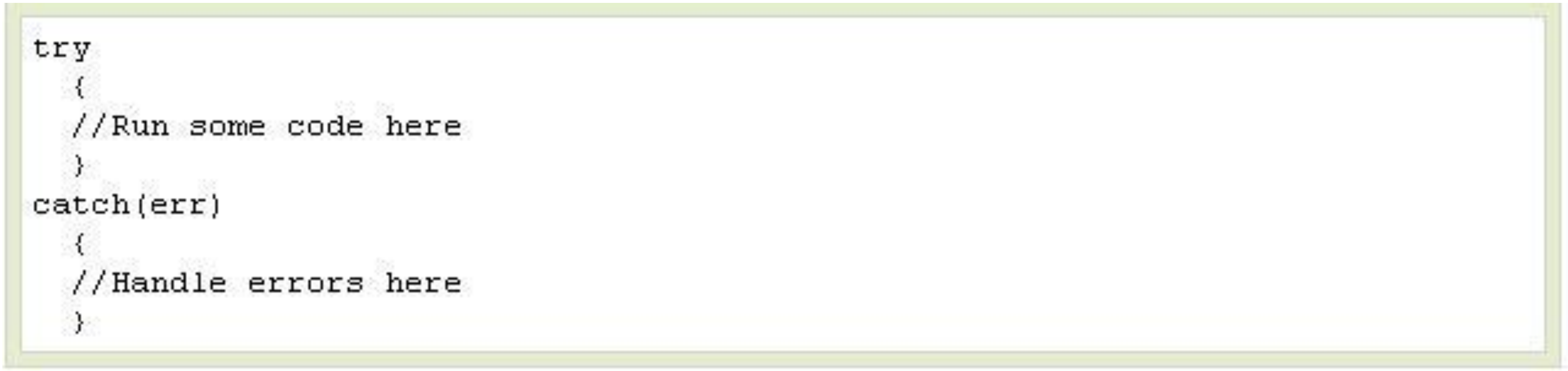
Example:



```
<html>
<head>
<script type="text/javascript">
function displaymessage()
{
    alert("Hello Word");
}
</script>
</head>
<body>
        <input type="button" value="Click me!" onclick="displaymessage()" />
</body>
</html>
```

The try…catch

- The try...catch statement allows you to test a block of code for errors.
- The try block contains the code to be run, and the catch block contains the code to be executed if an error occurs.
- Syntax:



```
try
  {
  //Run some code here
  }
catch(err)
  {
  //Handle errors here
  }
```

Events

- **What are Events?**
  - Events are actions that can be detected by JavaScript.
- **Examples of events:**
  - A mouse click
  - A web page or an image loading
  - Mousing over a hot spot on the web page
  - Selecting an input field in an HTML form
  - Submitting an HTML form
  - A keystroke

Events(cont)

- **onLoad and onUnload**
  - The onLoad and onUnload events are triggered when the user enters or leaves the page.
  - The onLoad event is often used to check the visitor's browser type and browser version, and load the proper version of the web page based on the information.
- **onFocus, onBlur and onChange**
  - The onFocus, onBlur and onChange events are often used in combination with validation of form fields.
- **onSubmit**
  - The onSubmit event is used to validate ALL form fields before submitting it.
- **onMouseOver**
  - The onmouseover event can be used to trigger a function when the user mouses over an HTML element

JavaScript Object

JavaScript is an Object Oriented Programming (OOP) language

- JavaScript is an Object Oriented Programming (OOP) language. An OOP language allows you to define your own objects and make your own variable types.
- **Constructors**
  - Functions named for class names. Then use “new”.
  - Can define properties with “this”
    - You must use “this” for properties used in constructors
  - **Example:**

```
function MyClass(n1) { this.foo = n1; }
var m = new MyClass(10);
```

JavaScript is an Object Oriented Programming (OOP) language (cont)

- **Properties**
  - Properties are the values associated with an object.
- **Methods**
  - Methods are the actions that can be performed on objects.
- **Example**

```
<script type="text/javascript“>
        var str="Hello World!“;
        document.write(str.length);
        document.write(str.toUpperCase());
</script>
```

JavaScript Object Type

- String
- Date
- Array
- Boolean
- Math
- RegExp

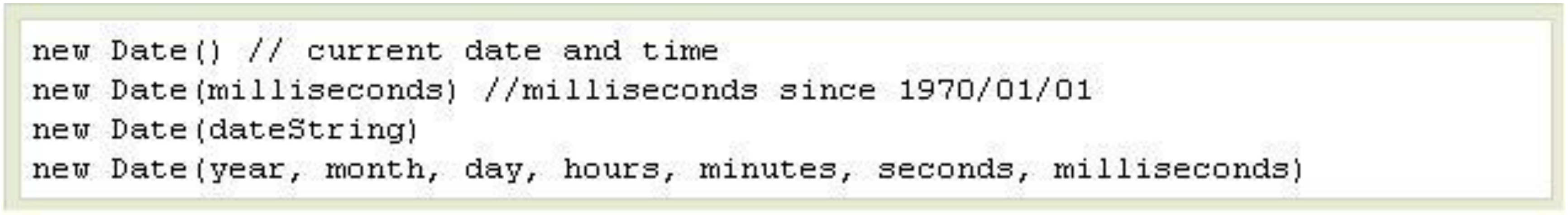
JavaScript String Object

- The String object is used to manipulate a stored piece of text.
- Example:

```
var str=“Hello word!”;
```

JavaScript Date Object

- The Date object is used to work with dates and times.
- Date objects are created with the Date() constructor.
- There are four ways of instantiating a date:



```
new Date() // current date and time
new Date(milliseconds) //milliseconds since 1970/01/01
new Date(dateString)
new Date(year, month, day, hours, minutes, seconds, milliseconds)
```

JavaScript Arrays

- An array is a special variable, which can hold more than one value, at a time.
- Example:

```
var names = ["Joe", "Jane", "John", "Juan"];
```

- Indexed at 0 as in Java

```
for(var i=0; i<names.length; i++) {
        doSomethingWith(names[i]);
}
```

JavaScript Arrays (cont)

- Arrays can be sparse
- Arrays can be resized
- Arrays have methods
- Regular objects can be treated like arrays

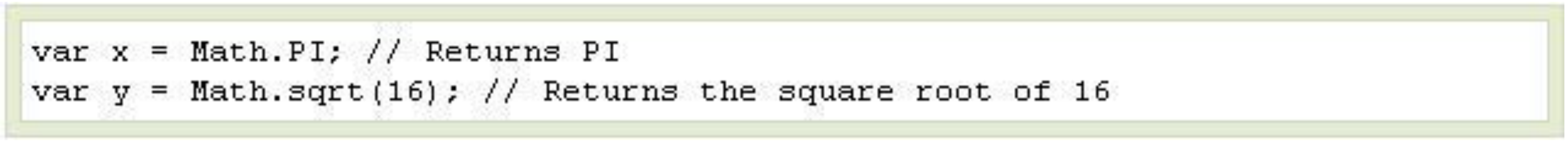
JavaScript Boolean

- The Boolean object represents two values: "true" or "false“

  var myBoolean=new Boolean();

JavaScript Math Object

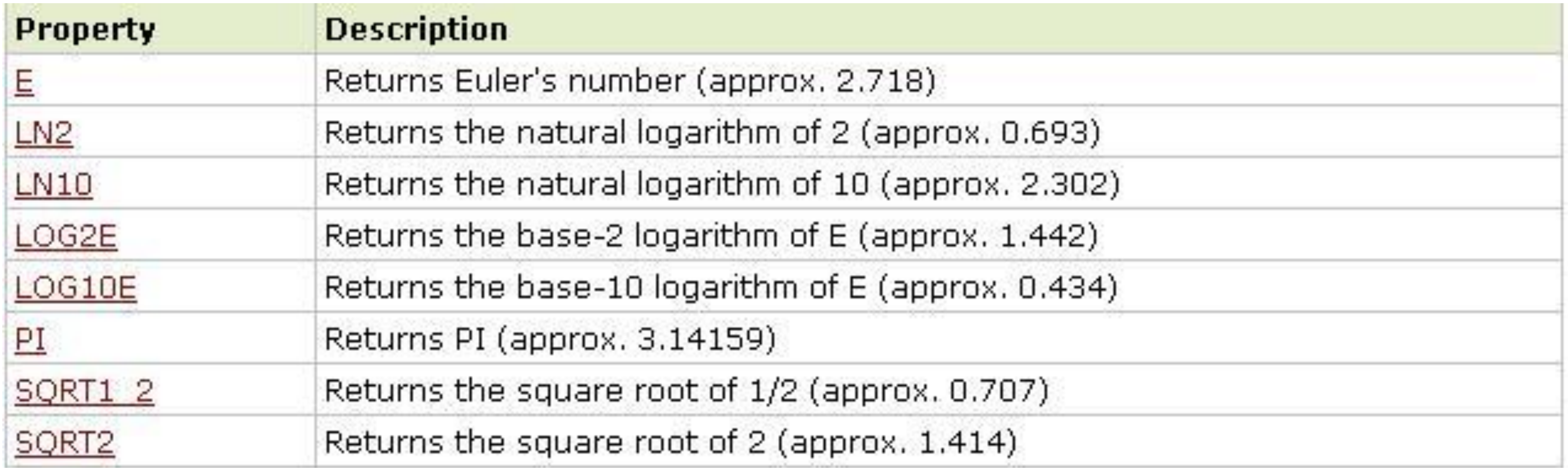
- The Math object allows you to perform mathematical tasks.
- Math is not a constructor. All properties/methods of Math can be called by using Math as an object, without creating it.
- Syntax:



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

JavaScript Math Object (cont)

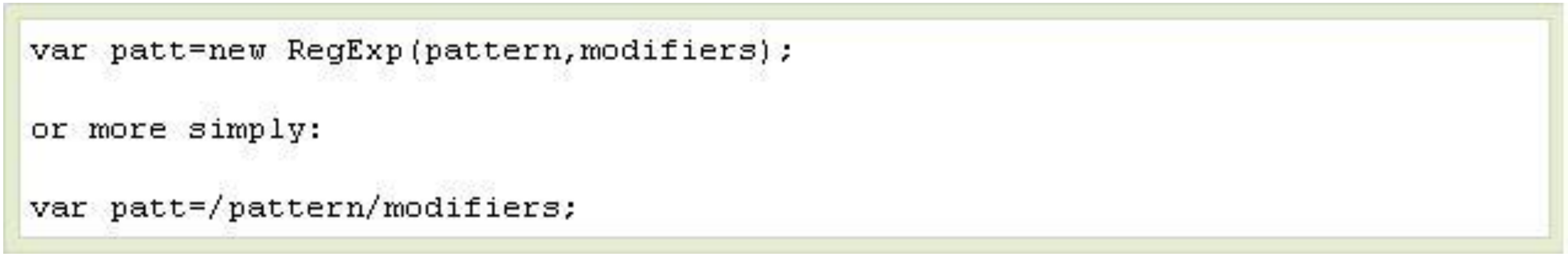
- Constants



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. 3.14159)
SQRT1_2	Returns the square root of 1/2 (approx. 0.707)
SQRT2	Returns the square root of 2 (approx. 1.414)

JavaScript RegExp Object

- A regular expression is an object that describes a pattern of characters.
- Regular expressions are used to perform pattern-matching and "search-and-replace" functions on text.
- Syntax:



```
var patt=new RegExp(pattern,modifiers);

or more simply:

var patt=/pattern/modifiers;
```

JavaScript Advance

Browser Detection

- **Browser Detection**
  - Almost everything in this tutorial works on all JavaScript-enabled browsers. However, there are some things that just don't work on certain browsers - especially on older browsers.
  - Sometimes it can be useful to detect the visitor's browser, and then serve the appropriate information.
  - The Navigator object contains information about the visitor's browser name, version, and more.
  - **Note:** There is no public standard that applies to the navigator object, but all major browsers support it.

Cookies

- A cookie is a variable that is stored on the visitor's computer. Each time the same computer requests a page with a browser, it will send the cookie too. With JavaScript, you can both create and retrieve cookie values.
- Example about cookie:
  - Name cookie
  - Password cookie
  - Date cookie

Form Validation

- JavaScript can be used to validate data in HTML forms before sending off the content to a server.
- Form data that typically are checked by a JavaScript could be:
  - 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?

Timing Events

- With JavaScript, it is possible to execute some code after a specified time-interval. This is called timing events.
- It's very easy to time events in JavaScript. The two key methods that are used are:
  - setTimeout() - executes a code some time in the future
  - clearTimeout() - cancels the setTimeout()

Examples

Q & A

Thank You

CSC

Experience. Results.

Summary

- **JavaScript Summary**
  - This tutorial has taught you how to add JavaScript to your HTML pages, to make your web site more dynamic and interactive.
  - You have learned how to create responses to events, validate forms and how to make different scripts run in response to different scenarios.
  - You have also learned how to create and use objects, and how to use JavaScript's built-in objects.