AMRITA VISHWA VIDYAPEETHAM

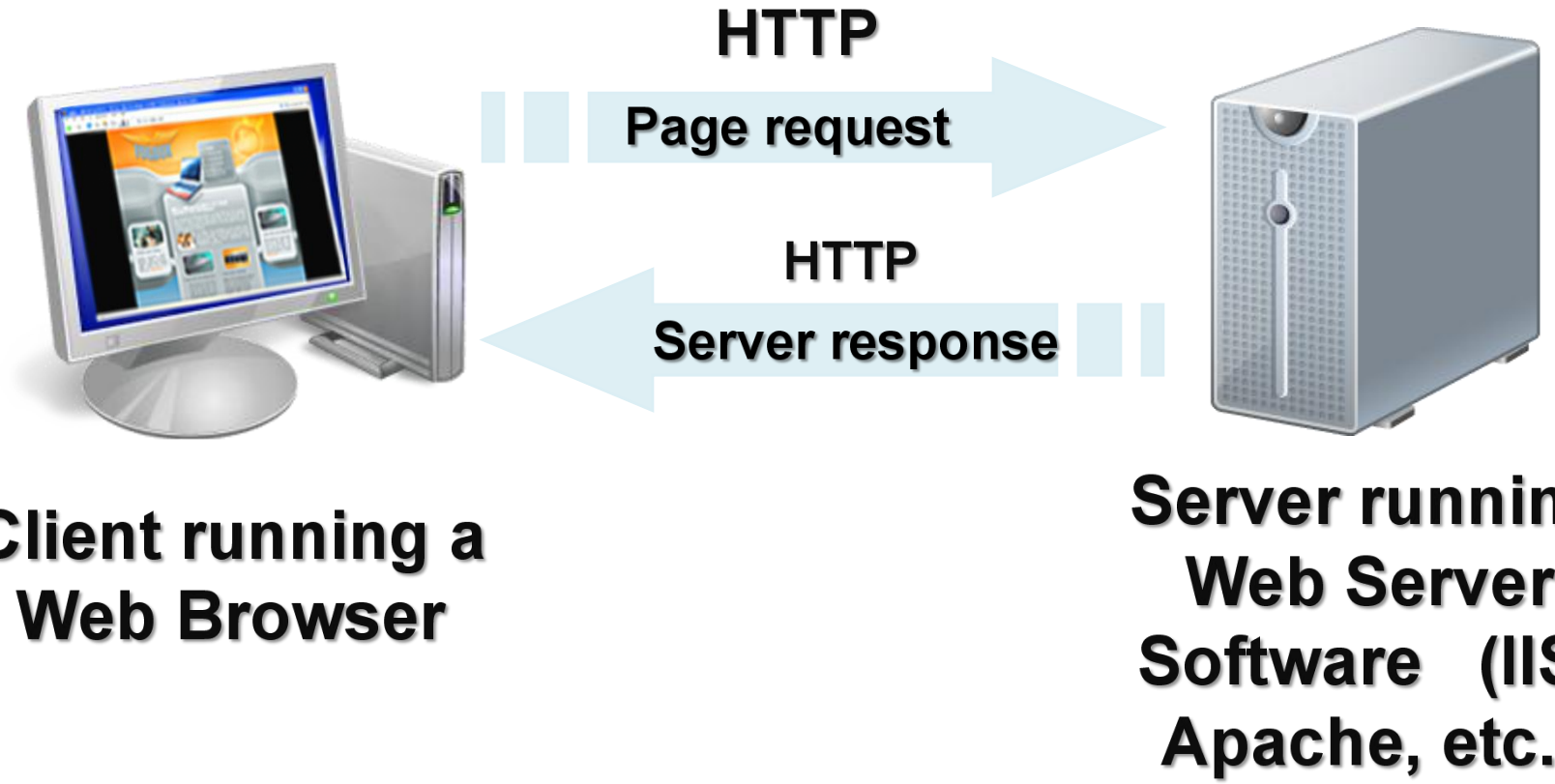
15CSE337
Cloud Computing and Services

Quick Intro to HTML/JavaScript

Dr Ganesh Neelakanta Iyer
Associate Professor, Dept of Computer Science and Engg
Amrita Vishwa Vidyapeetham, Coimbatore

How the Web Works?

- WWW use classical client / server architecture
  - HTTP is text-based request-response protocol



What is a Web Page?

- Web pages are text files containing HTML
- HTML – Hyper Text Markup Language
  - A notation for describing
    - document structure (semantic markup)
    - formatting (presentation markup)
  - Looks (looked?) like:
    - A Microsoft Word document
- The markup tags provide information about the page content structure

Markup Languages

- Markup:
  - Embedded codes in documents
  - Codes are called `tags'
  - Codes
    - Describe the structure documents
    - Include instructions for processing
- Markup language:
  - Computer language for describing syntax of tags
  - May be used with other tools to specify rendering

HTML Elements

- Tags are the elements that create the components of a page
- Tags surrounded by angle brackets < >
- Usually come in pairs
  - Example: Start tag and end tag
- Stuff between is called “element content”
- Tags are not case sensitive
  - New standard is to use lower case
- For example, <B>HI THERE</B> would display the words HI THERE in bold.

Tag Attributes

- Tags can have attributes that provide additional information to an HTML element
  - Attributes always come in name/value pairs like: name="value"
  - Attributes are always specified in the start tag
  - Attribute values should always be enclosed in quotes. Double quotes are most common.
  - Also case-insensitive: however, lowercase is recommended
  - <tagname a1="v1" a2="v2"></tagname>
  - For example, <table border="0"> is a start tag that defines a table that has no borders

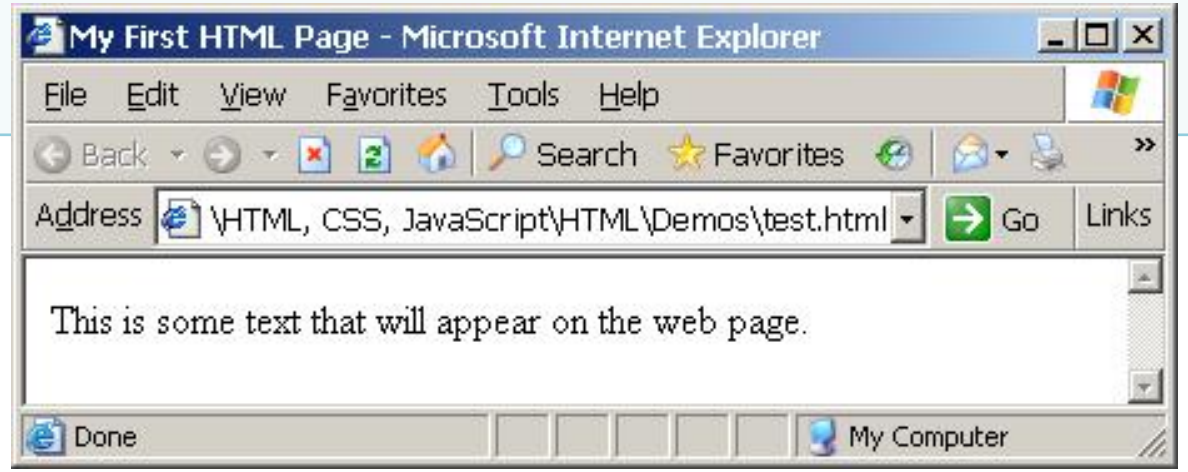
HTML Document Structure

- Entire document enclosed within <html> and </html> tags
- Two subparts:
  - Head
    - Enclosed within <head> and </head>
    - Within the head, more tags can be used to specify title of the page, meta-information, etc.
  - Body
    - Enclosed within <body> and </body>
    - Within the body, content is to be displayed
    - Other tags can be embedded in the body

First HTML Page

test.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>My First HTML Page</title>
  </head>
  <body>
     <p>This is some text...</p>
  </body>
</html>
```



https://codepen.io/pen
https://html-online.com/editor/


JavaScript

Table of Contents

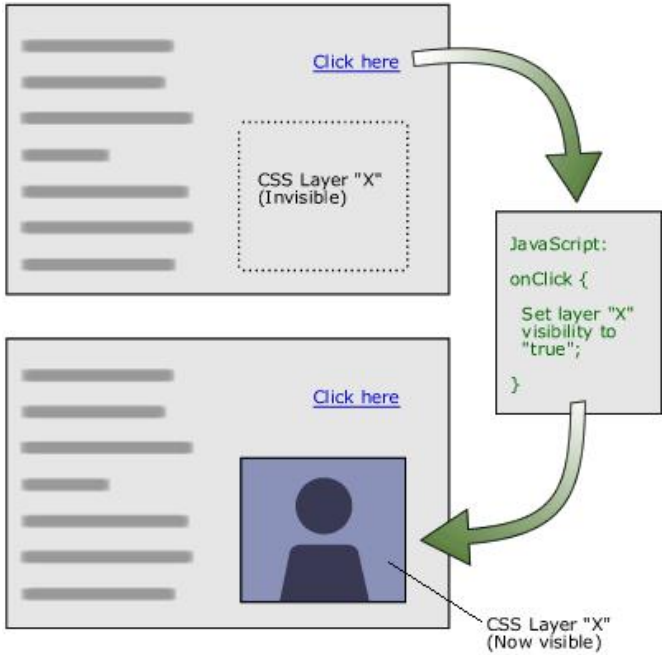
- What is DHTML?
- DHTML Technologies
  - XHTML, CSS, JavaScript, DOM

Table of Contents (2)

Table of Contents (3)

DHTML

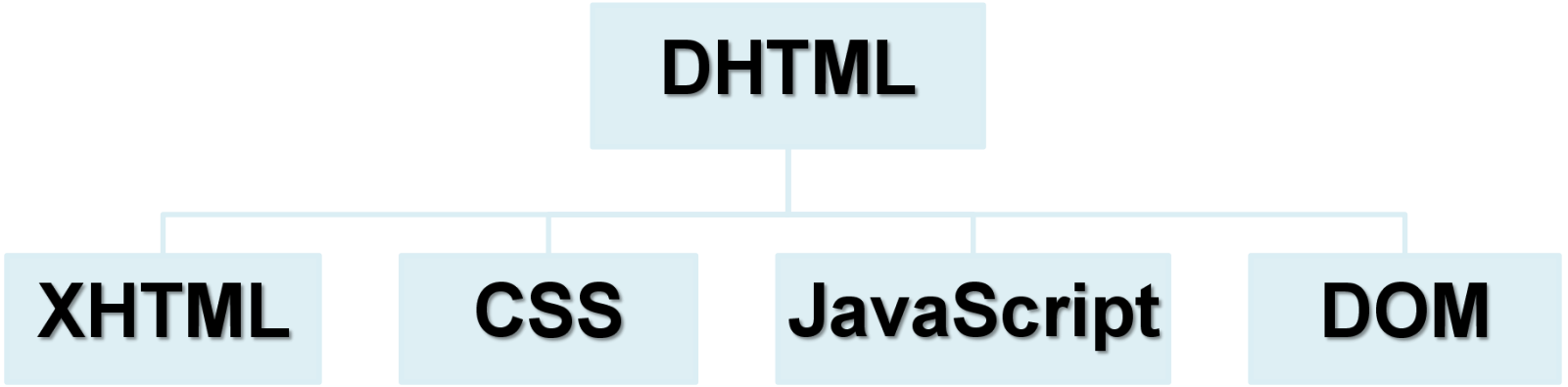
Dynamic Behavior at the Client Side



What is DHTML?

- Dynamic HTML (DHTML)
  - Makes possible a Web page to react and change in response to the user's actions

- DHTML = HTML + CSS + JavaScript



DTHML = HTML + CSS + JavaScript

- HTML defines Web sites content through semantic tags (headings, paragraphs, lists, …)
- CSS defines 'rules' or 'styles' for presenting every aspect of an HTML document
  - Font (family, size, color, weight, etc.)
  - Background (color, image, position, repeat)
  - Position and layout (of any object on the page)
- JavaScript defines dynamic behavior
  - Programming logic for interaction with the user, to handle events, etc.

JavaScript

Dynamic Behavior in a Web Page

JavaScript

- JavaScript is a front-end scripting language developed by Netscape for dynamic content
  - Lightweight, but with limited capabilities
  - Can be used as object-oriented language
- Client-side technology
  - Embedded in your HTML page
  - Interpreted by the Web browser
- Simple and flexible
- Powerful to manipulate the DOM

JavaScript Advantages

- JavaScript allows interactivity such as:
  - Implementing form validation
  - React to user actions, e.g. handle keys
  - Changing an image on moving mouse over it
  - Sections of a page appearing and disappearing
  - Content loading and changing dynamically
  - Performing complex calculations
  - Custom HTML controls, e.g. scrollable table
  - Implementing AJAX functionality

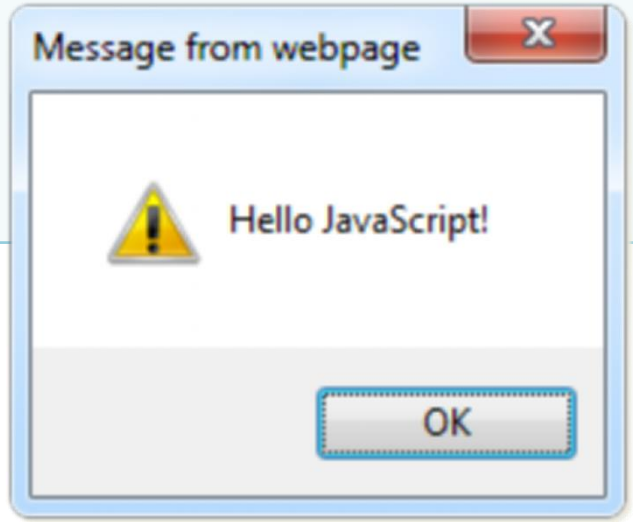
What Can JavaScript Do?

- Can handle events
- Can read and write HTML elements and modify the DOM tree
- Can validate form data
- Can access / modify browser cookies
- Can detect the user's browser and OS
- Can be used as object-oriented language
- Can handle exceptions
- Can perform asynchronous server calls (AJAX)

The First Script

first-script.html

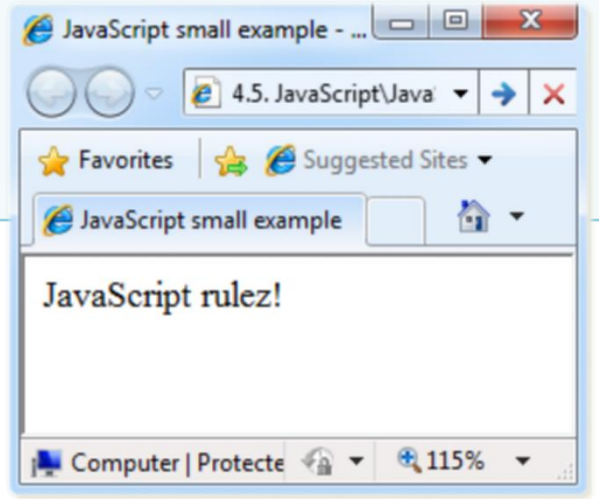
```
<html>

<body>
  <script type="text/javascript">
     alert('Hello JavaScript!');
  </script>
</body>

</html>
```



Another Small Example

small-example.html

```
<html>

<body>
  <script type="text/javascript">
     document.write('JavaScript rulez!');
  </script>
</body>

</html>
```



Using JavaScript Code

- The JavaScript code can be placed in:
  - `<script>` tag in the head
  - `<script>` tag in the body – not recommended
  - External files, linked via `<script>` tag the head
    - Files usually have `.js` extension

```
<script src="scripts.js" type="text/javscript">
<!- code placed here will not be executed! -->
</script>
```

    - Highly recommended
    - The `.js` files get cached by the browser

JavaScript – When is Executed?

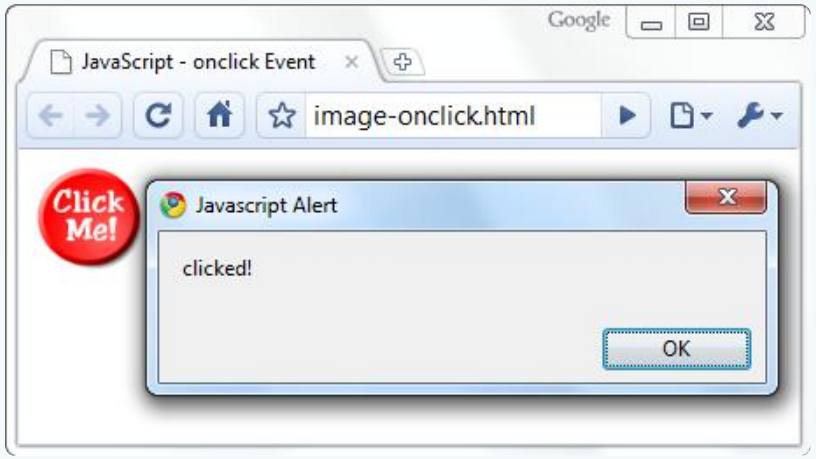
- JavaScript code is executed during the page loading or when the browser fires an event
  - All statements are executed at page loading
  - Some statements just define functions that can be called later
- Function calls or code can be attached as "event handlers" via tag attributes
  - Executed when the event is fired by the browser

```
<img src="logo.gif" onclick="alert('clicked!')" />
```

Calling a JavaScript Function from Event Handler – Example

image-onclick.html

```
<html>
<head>
<script type="text/javascript">
  function test (message) {
    alert(message);
  }
</script>
</head>

<body>
  <img src="logo.gif"
    onclick="test('clicked!')" />
</body>
</html>
```

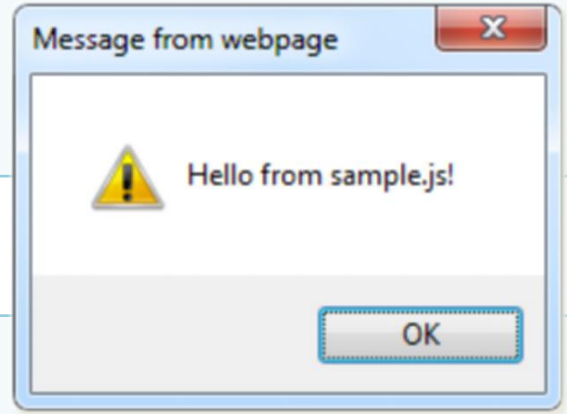


Using External Script Files

- Using external script files:

external-JavaScript.html

```
<html>
<head>
  <script src="sample.js" type="text/javascript">
  </script>
</head>
<body>
  <button onclick="sample()" value="Call JavaScript
    function from sample.js" />
</body>
</html>
```



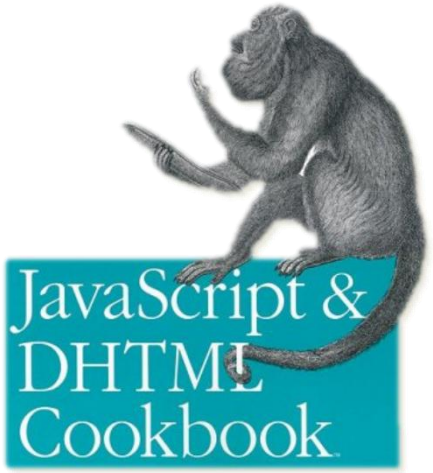
- External JavaScript file:

```
function sample() {
  alert('Hello from sample.js!')
}
```

sample.js

The JavaScript Syntax

```
if (pop < 10)
{
    map.graphics.add(features[i].setSymbol(onePopSymbol));
}
else if (pop >= 10 && pop < 95)
{
    map.graphics.add(features[i].setSymbol(twoPopSymbol));
}
else if (pop >= 95 && pop < 365)
{
    map.graphics.add(features[i].setSymbol(threePopSymbol));
}
else if (pop >= 365 && pop < 1100)
{
    map.graphics.add(features[i].setSymbol(fourPopSymbol));
}
else
{
    map.graphics.add(features[i].setSymbol(fivePopSymbol));
}
```



JAVA SCRIPT

JavaScript Syntax

- The JavaScript syntax is similar to C# and Java
  - Operators (+, *, =, !=, &&, ++, …)
  - Variables (typeless)
  - Conditional statements (`if`, `else`)
  - Loops (`for`, `while`)
  - Arrays (`my_array[]`) and associative arrays (`my_array['abc']`)
  - Functions (can return value)
  - Function variables (like the C# delegates)

Data Types

- JavaScript data types:
  - Numbers (integer, floating-point)
  - Boolean (true / false)
- String type – string of characters

```
var myName = "You can use both single or double
quotes for strings";
```

- Arrays

```
var my_array = [1, 5.3, "aaa"];
```

- Associative arrays (hash tables)

```
var my_hash = {a:2, b:3, c:"text"};
```

Everything is Object

- Every variable can be considered as object
  - For example strings and arrays have member functions:

objects.html

```
var test = "some string";
alert(test[7]); // shows letter 'r'
alert(test.charAt(5)); // shows letter 's'
alert("test".charAt(1)); //shows letter 'e'
alert("test".substring(1,3)); //shows 'es'
```

```
var arr = [1,3,4];
alert (arr.length); // shows 3
arr.push(7); // appends 7 to end of array
alert (arr[3]); // shows 7
```

String Operations

- The + operator joins strings

```
string1 = "fat ";
string2 = "cats";
alert(string1 + string2);  // fat cats
```

- What is "9" + 9?

```
alert("9" + 9);  // 99
```

- Converting string to number:

```
alert(parseInt("9") + 9);  // 18
```

Arrays Operations and Properties

- Declaring new empty array:

```
var arr = new Array();
```

- Declaring an array holding few elements:

```
var arr = [1, 2, 3, 4, 5];
```

- Appending an element / getting the last element:

```
arr.push(3);
var element = arr.pop();
```

- Reading the number of elements (array length):

```
arr.length;
```

- Finding element's index in the array:

```
arr.indexOf(1);
```

Standard Popup Boxes

- Alert box with text and [OK] button
  - Just a message shown in a dialog box:

```
alert("Some text here");
```

- Confirmation box
  - Contains text, [OK] button and [Cancel] button:

```
confirm("Are you sure?");
```

- Prompt box
  - Contains text, input field with default value:

```
prompt ("enter amount", 10);
```

Sum of Numbers – Example

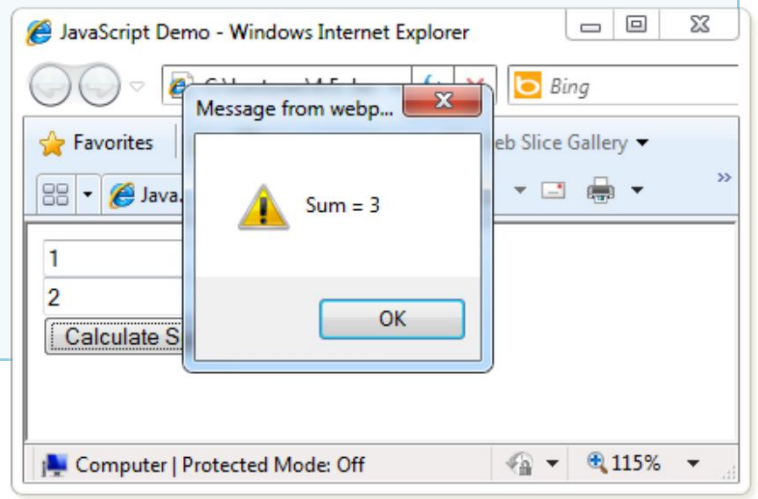
sum-of-numbers.html

```
<html>

<head>
  <title>JavaScript Demo</title>
  <script type="text/javascript">
    function calcSum() {
      value1 =
        parseInt(document.mainForm.textBox1.value);
      value2 =
        parseInt(document.mainForm.textBox2.value);
      sum = value1 + value2;
      document.mainForm.textBoxSum.value = sum;
    }
  </script>
</head>
```

Sum of Numbers – Example (2)

sum-of-numbers.html (cont.)

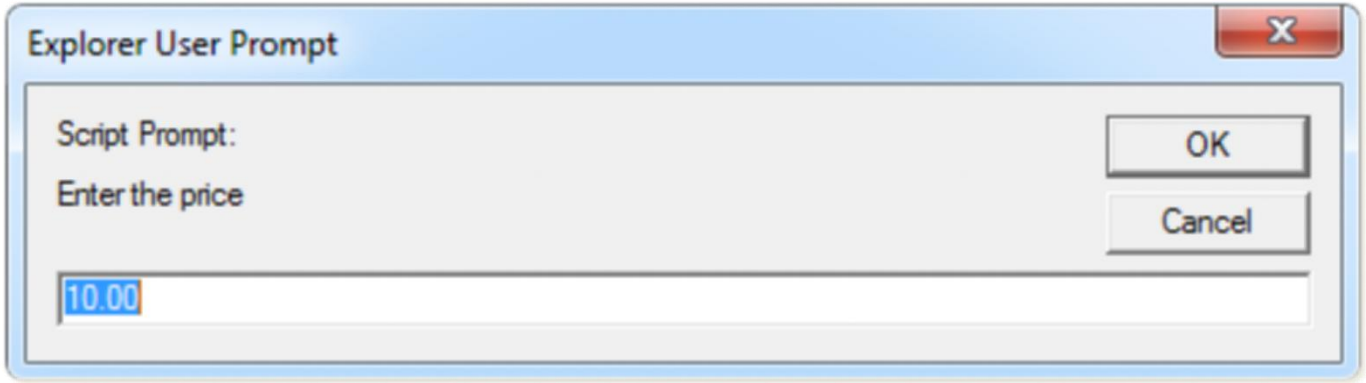
```
<body>
  <form name="mainForm">
    <input type="text" name="textBox1" /> <br/>
    <input type="text" name="textBox2" /> <br/>
    <input type="button" value="Process"
      onclick="javascript: calcSum()" />
    <input type="text" name="textBoxSum"
      readonly="readonly"/>
  </form>
</body>

</html>
```

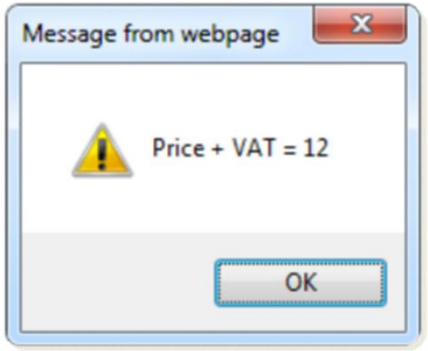


JavaScript Prompt – Example

prompt.html

```
price = prompt("Enter the price", "10.00");
alert('Price + VAT = ' + price * 1.2);
```





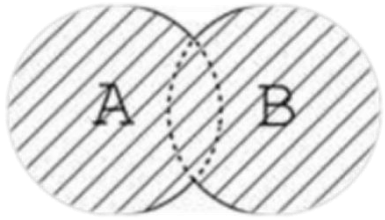
Conditional Statement (`if`)

```
unitPrice = 1.30;
if (quantity > 100) {
  unitPrice = 1.20;
}
```

Symbol	Meaning
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
==	Equal
!=	Not equal

Conditional Statement (`if`) (2)

- The condition may be of Boolean or integer type:



conditional-statements.html

```
var a = 0;
var b = true;
if (typeof(a)=="undefined" || typeof(b)=="undefined") {
  document.write("Variable a or b is undefined.");
}
else if (!a && b) {
  document.write("a==0; b==true;");
} else {
  document.write("a==" + a + "; b==" + b + ";");
}
```

Switch Statement

- The `switch` statement works like in C#:

switch-statements.html

```
switch (variable) {
  case 1:
    // do something
    break;
  case 'a':
    // do something else
    break;
  case 3.14:
    // another code
    break;
  default:
    // something completely different
}
```

Loop

- Like in C#
  - `for` loop
  - `while` loop
  - `do … while` loop


```
var counter;
for (counter=0; counter<4; counter++) {
  alert(counter);
}
while (counter < 5) {
  alert(++counter);
}
```

loops.html

Functions

- Code structure – splitting code into parts
- Data comes in, processed, result returned

```
function average(a, b, c)
{
    var total;
    total = a+b+c;
    return total/3;
}
```

Parameters come in here.

Declaring variables is optional. Type is never declared.

Value returned here.

Function Arguments and Return Value

- Functions are not required to return a value
- When calling function it is not obligatory to specify all of its arguments
  - The function has access to all the arguments passed via `arguments` array

```
function sum() {
  var sum = 0;
  for (var i = 0; i < arguments.length; i ++)
    sum += parseInt(arguments[i]);
  return sum;
}
alert(sum(1, 2, 4));
```

functions-demo.html

References

- https://developer.mozilla.org/en-US/docs/Web/JavaScript/A_re-introduction_to_JavaScript

Thank you!

Dr Ganesh Neelakanta Iyer

ni_amrita@cb.amrita.edu
ganesh.vigneswara@gmail.com

Office Hours
– Tuesday 4-445 PM @ My office

त्यागेनैके अमृतत्वमानशुः
MATA AMRITANANDAMAYI MATH
श्रद्धावान् लभते ज्ञानम्