INTRODUCTION TO JAVASCRIPT

Outline – Part A

- Overview of JavaScript
  - Versions, embedding, comments
- JavaScript Basics
  - Variables and Data Types
  - Operators
  - Expressions
- JavaScript Control Structures
  - Conditional Statements
  - Looping Statements

Outline – Part B

- JavaScript Functions and Events
  - Events Handlers
- Using Object in JavaScript
  - Object-Oriented Programming
  - JavaScript Object Model
  - Using Built-In objects (Predefined Object)
  - Custom Object Types
- Error in JavaScript
- Exception Handling in JavaScript

Outline – Part C

- Working with Browser Objects
  - Document Object Model (DOM)
- Creating Cookies in JavaScript
  - Constructing a standard cookie
  - Interaction with the cookie
- Introducing DHTML
  - Styles and Layers
  - Dynamic Positioning

Outline – Part D

- JavaScript Application Development
  - JavaScript Example
  - Discuss the execution requirements
  - How to break down the syntax
- Cool JavaScript Sites
- JavaScript and DHTML Reference
- Hints for JavaScript coding
- Summary

Introduction

- The growth of the WWW has resulted in a demand for dynamic and interactive web sites.
- There are many different kinds of scripting languages – JavaScript, …
- This lecture aims at offering in-depth knowledge of JavaScript, discussing the complexity of scripting and studying various common examples.

JavaScript Capabilities

- Improve the user interface of a website
- Make your site easier to navigate
- Easily create pop-up alert, windows
- Replace images on a page without reload the page
- Form validation
- Many others …

The Future of JavaScript

- ECMA - An International industry association dedicated to standardize information and communication systems.
- JavaScript was invented by Brendan Eich in 1995, and became an ECMA standard in 1997. ECMA-262 is the official name of the standard.

JavaScript Versions

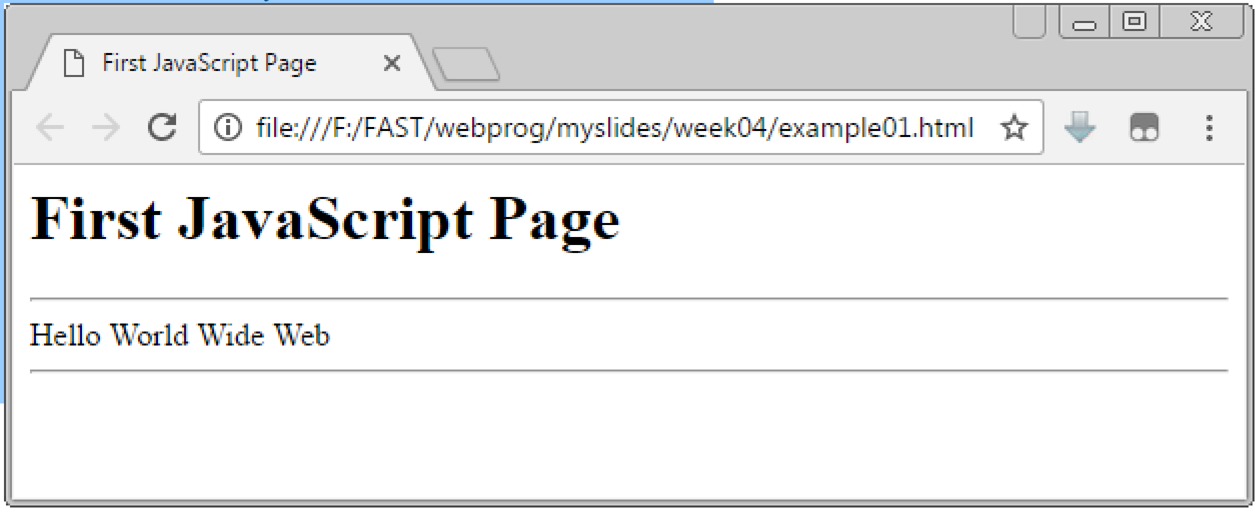
Year	Name	Description
1997	ECMAScript 1	First Edition.
1998	ECMAScript 2	Editorial changes only.
1999	ECMAScript 3	Added Regular Expressions. Added try/catch.
	ECMAScript 4	Was never released.
2009	ECMAScript 5	Added "strict mode". Added JSON support.
2011	ECMAScript 5.1	Editorial changes.
2015	ECMAScript 6	Added classes and modules.
2016	ECMAScript 7	Added exponential operator (**). Added Array.prototype.includes.

JavaScript / ECMAScript / JScript

- **JavaScript** developed by Netscape. The first browser to run JavaScript was Netscape 2 (1996). Now Mozilla foundation continued to develop JavaScript for the Firefox browser. JavaScript version 1.0 to 1.8.
- **ECMAScript** was developed by Ecma International after the organization adopted JavaScript. The first edition of ECMAScript was released in 1997. ECMAScript version numbers run from 1 to 7.
- **JScript** was developed by Microsoft as a compatible JavaScript language for Internet Explorer in 1996. JScript version numbers runs from 1.0 to 9.0.

A Simple Script

```
<html>
<head>  <title>First JavaScript Page</title> </head>
<body>
<h1>First JavaScript Page</h1>
<script type="text/javascript">
<!--
document.write("<hr>");
document.write("Hello World Wide Web");
document.write("<hr>");
-->
</script>
</body>
</html>
```



First JavaScript Page

Hello World Wide Web

Embedding JavaScript

```html
<html>
<head>
<title>First JavaScript Program</title>
</head>
<body>
<script language="JavaScript" src="your_source_file.js"></script>
</body>
</html>
```

- A <script> tag can be placed either within the <head> or <body> tag of an HTML document.

JavaScript Source File

```
<script language="JavaScript"
src="your_source_file.js"></script>
```

- SRC – specifies the location of an external script
- TYPE – specifies the scripting language of the script
- LANGUAGE – specifies the scripting language of the script
- TYPE and LANGUAGE have a similar function, we use LANGUAGE to specify the language used in the script

Need for a source file

- If JavaScript <u>code is short</u>, you should include the code in the HTML document.
- To add <u>clarity </u>to an HTML document.
- To <u>share</u> JavaScript code across multiple HTML documents.
- To help you <u>hide</u> your JavaScript code.
  - Viewer can only see the location of the source file but not the contents.

Hide JavaScript from incompatible browsers

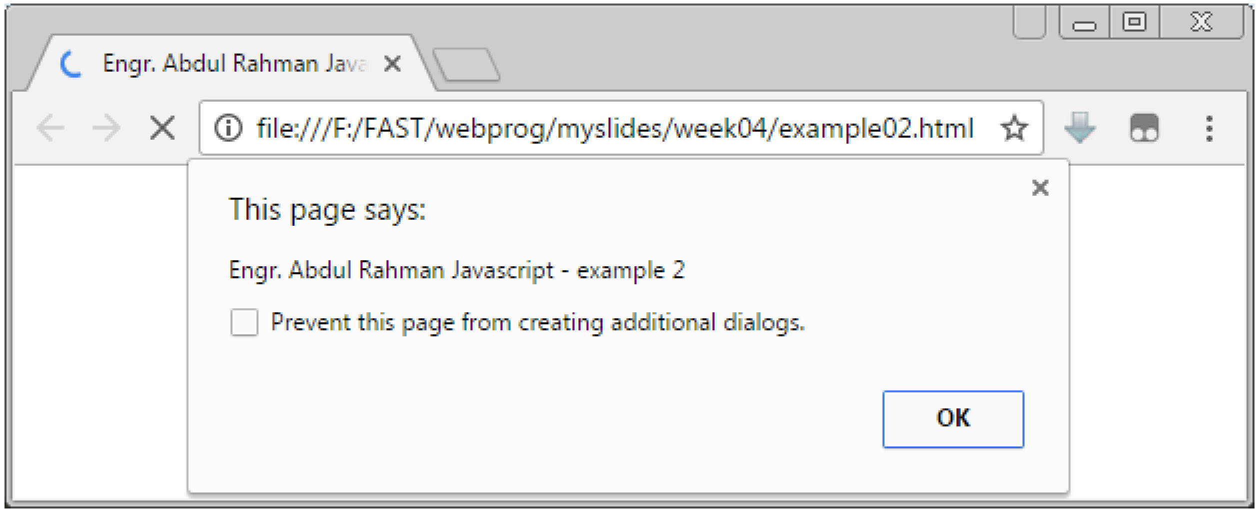
```
<script language="JavaScript">
<!- begin hiding JavaScript
// single-line comment, /* … */ multiple-line comment
End hiding JavaScript -->
</script>
<noscript>
Your browser does not support JavaScript.
</noscript>
```

- Browsers without JavaScript: NN1, IE2, lynx.
- **Modernizr** if you want to use some of the new cool HTML5 features, don't test if the browser is such and such version: test if the browser supports the feature you would like to use.

Using the alert() Method

```
<head>
<script language="JavaScript">
        alert("Engr. Abdul Rahman Javascript - example 2");
</script>
</head>
```

- Use to display text to user.
- Click "OK" to close.



This page says:

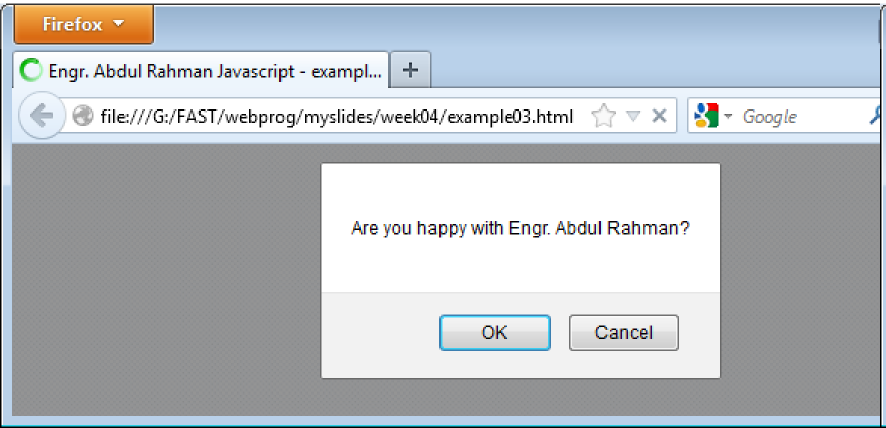
Engr. Abdul Rahman Javascript - example 2

☐ Prevent this page from creating additional dialogs.

Using the confirm() Method

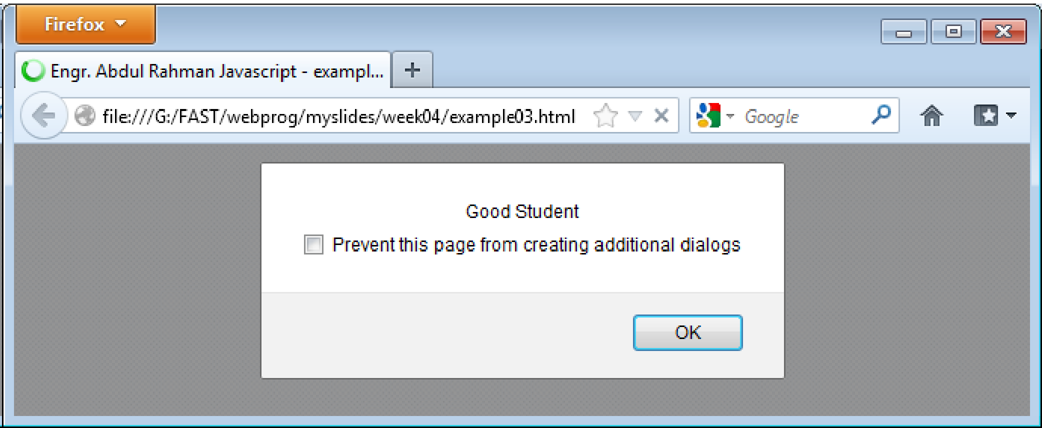
```
<script language="JavaScript">
          var Response  confirm("Are you happy with Engr. Abdul Rahman?");
          if (Response == true) {
                    alert("Good Student");
          } else {
                    alert("Bad Student");
          }
</script>
```

- This box is used to give the user a choice either OK or Cancel.
- It is very similar to the “alert()” method.
- You can also put your message in the method.


Firefox
Engr. Abdul Rahman Javascript - exampl...
file:///G:/FAST/webprog/myslides/week04/example03.html

Are you happy with Engr. Abdul Rahman?

OK Cancel


Firefox
Engr. Abdul Rahman Javascript - exampl...
file:///G:/FAST/webprog/myslides/week04/example03.html

Good Student

☐ Prevent this page from creating additional dialogs

OK

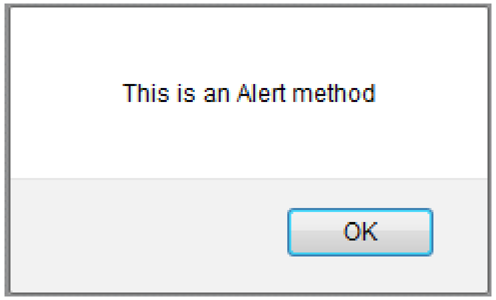
Using the prompt() Method

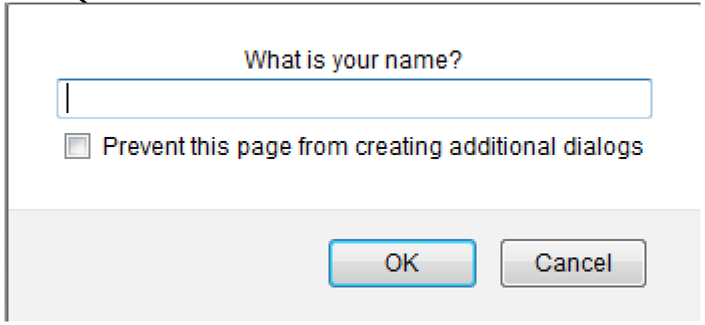
```
<html>
   <head> <script type="text/javascript">
         <!--
            function getValue(){
               var retVal = prompt("Enter your name : ", "your name here");
               document.write("You have entered : " + retVal);
            }
         //-->
      </script> </head>
   <body>
      <p>Click the following button to see the result: </p>
      <form>
         <input type="button" value="Click Me" onclick="getValue();" />
      </form>
   </body>
</html>
```

- allows the user to type in his own response to the specific question.
- You can give a default value to avoid displaying “undefined”.

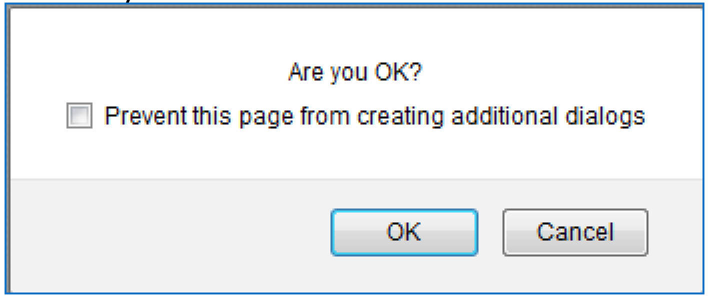
Three methods

```
<script language="JavaScript">
alert("This is an Alert method");
confirm("Are you OK?");
prompt("What is your name?");
prompt("How old are you?","20");
</script>
```

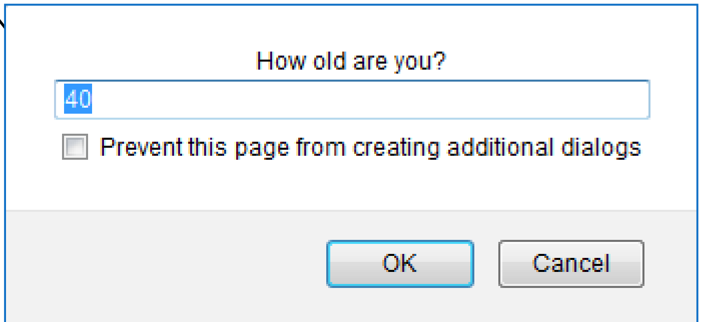

This is an Alert method


What is your name?

Prevent this page from creating additional dialogs


Are you OK?

Prevent this page from creating additional dialogs


How old are you?

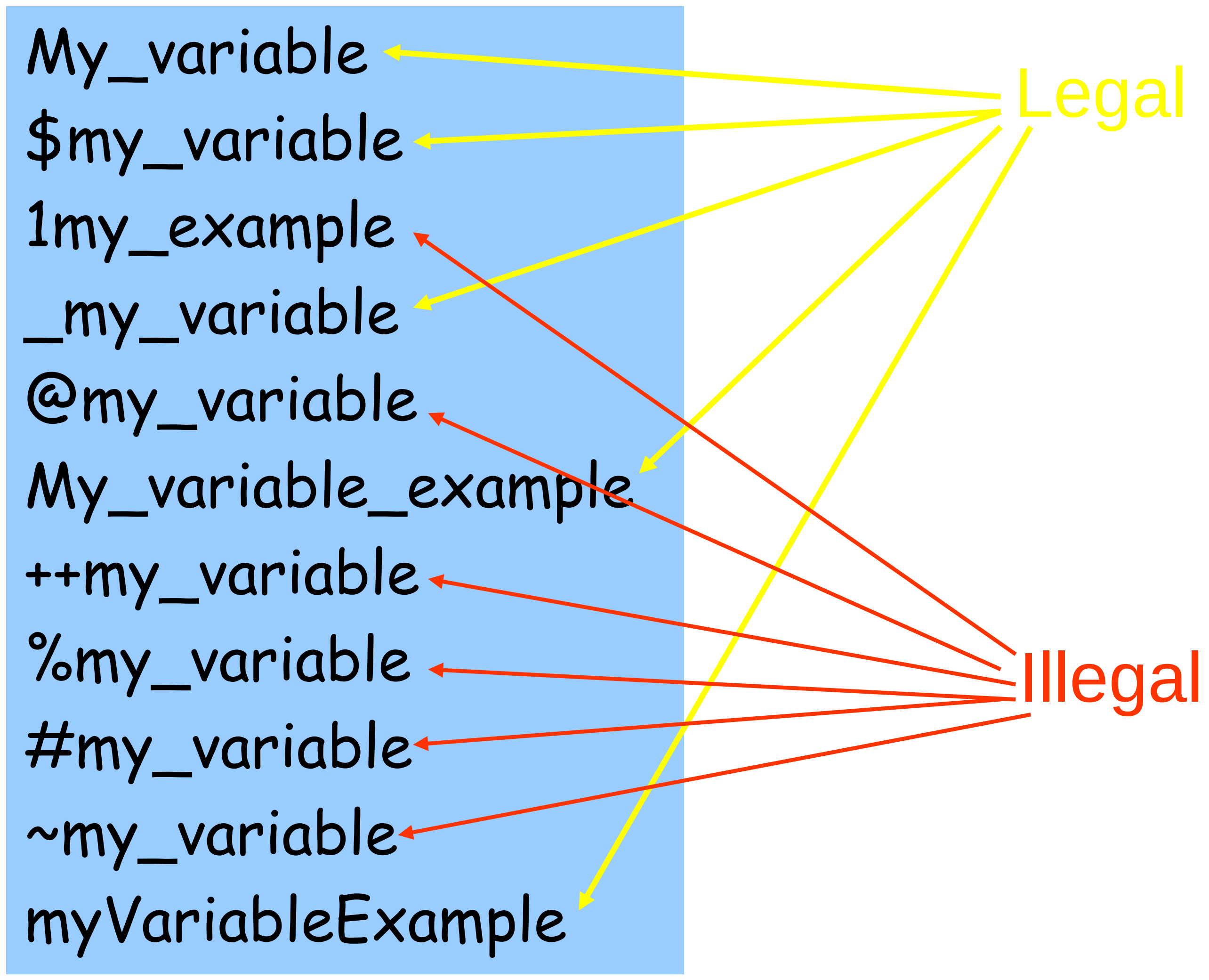
40

Prevent this page from creating additional dialogs

Variables

- JavaScript allows you to declare and use variables to store values.
- How to assign a name to a variable?
  - Include uppercase and lowercase letters
  - Digits from 0 through 9
  - The underscore _ and the dollar sign $
  - No space and punctuation characters
  - First character must be alphabetic letter or underscore
  - Case-sensitive
  - No reserved words or keywords

Which one is legal?



Variable on-the-fly

```
<head>
<script language="JavaScript">
        var id;
        id = prompt("What is your student id number?");
        alert(id);
        name = prompt("What is your name?","No name");
        alert(name);
</script>
</head>
```

Variable declaration

- We should use "var" because it is more easy to keep track of the variables.

Data Types

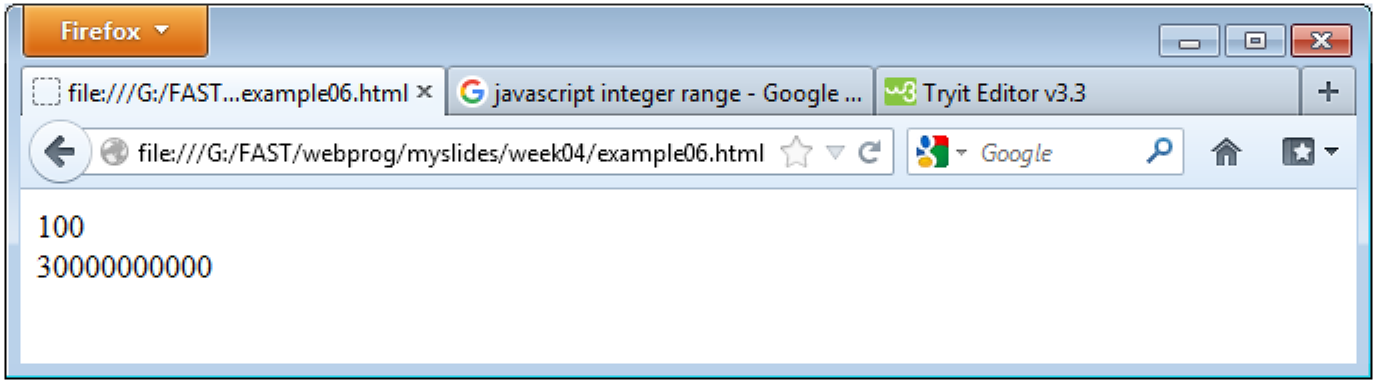
- JavaScript allows the same variable to contain different types of data values.
- Primitive data types
  - Number: integer & floating-point numbers
  - Boolean: logical values “true” or “false”
  - String: a sequence of alphanumeric characters
- Composite data types (or Complex data types)
  - Object: a named collection of data
  - Array: a sequence of values
- Special data types
  - Null: an initial value is assigned
  - Undefined: the variable has been created by not yet assigned a value

Numeric Data Types

- It is an important part of any programming language for doing arithmetic calculations.
- JavaScript supports:
  - Integers: A positive or negative number with no decimal places.
    - Ranged from – (2^{53} -1) to (2^{53} -1)
  - Floating-point numbers: usually written in exponential notation.
    - 3.1415…, 2.0e11

Integer and Floating-point number example

```
<script language="JavaScript">
        var integerVar = 100;
        var floatingPointVar = 3.0e10;
        // floating-point number 30000000000
        document.write(integerVar+"<BR>");
        document.write(floatingPointVar);
</script>
```



Firefox
file:///G:/FAST...example06.html
javascript integer range - Google ...
Tryit Editor v3.3
file:///G:/FAST/webprog/myslides/week04/example06.html
Google

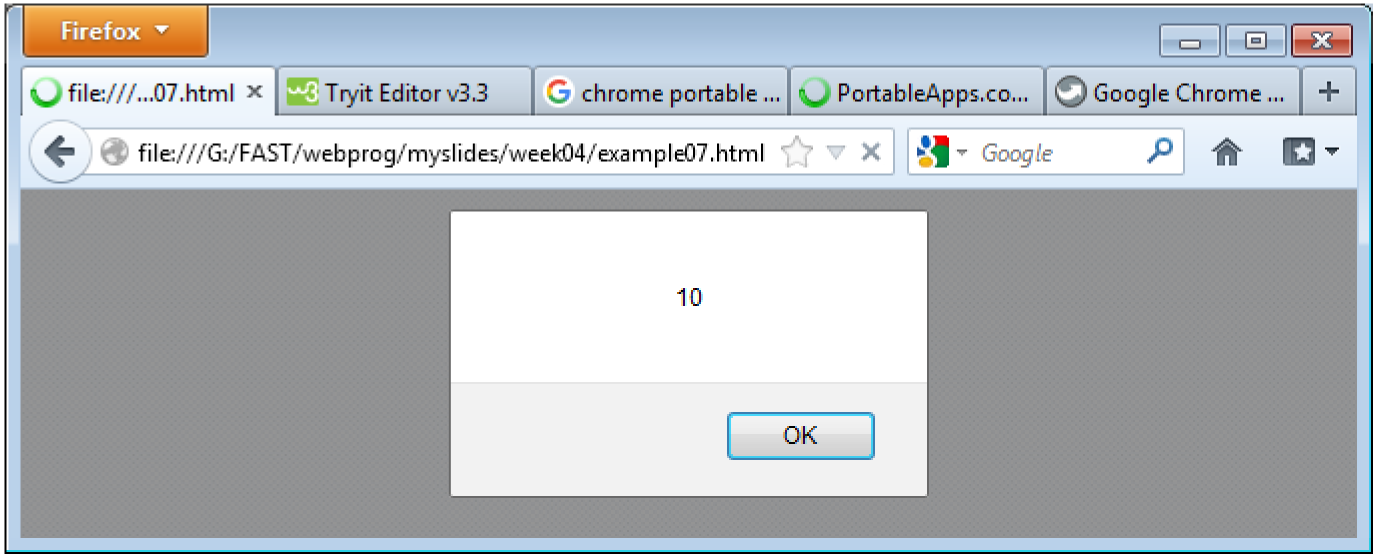
```
100
30000000000
```

Boolean Values

- A Boolean value is a logical value of either true or false. (yes/no, on/off)
- Often used in decision making and data comparison.
- In JavaScript, you can use the words “true” and “false” directly to indicate Boolean values.
- Named by the 19th century mathematician “George Boole”.

Boolean value example

```
<head>
<script language="JavaScript">
        var result;
        result = (true*10 + false*7);
        alert(result);
</script>
</head>
```



```
10
```

(1*10 + 0*7) = 10

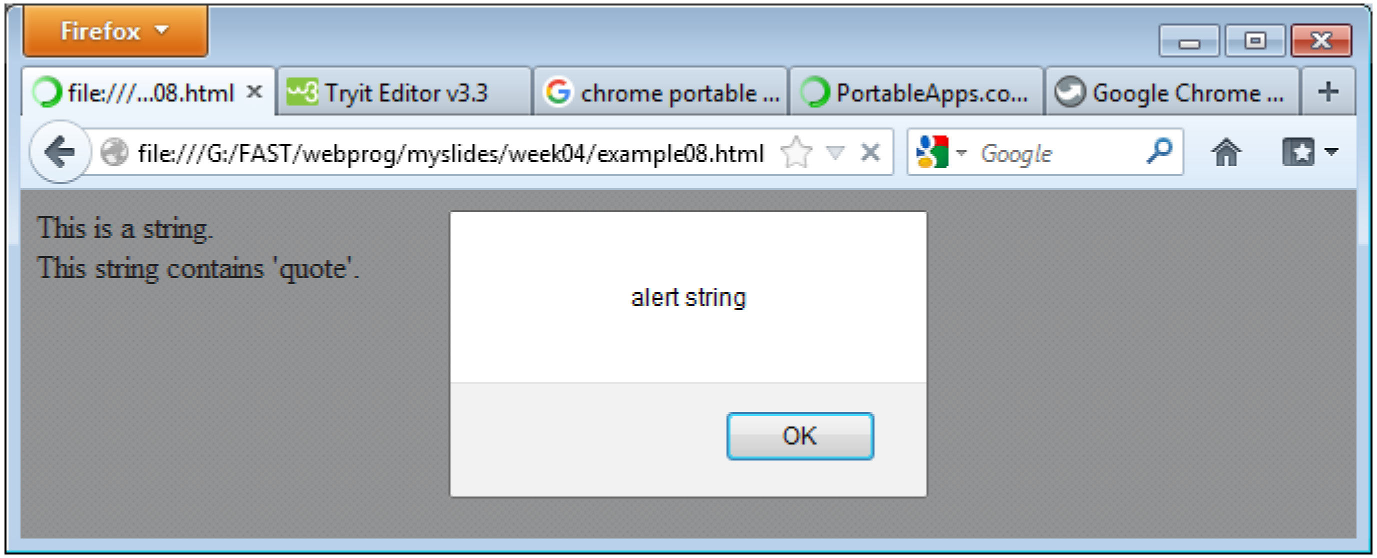
Strings

- A string variable can store a sequence of alphanumeric characters, spaces and special characters.
- String can also be enclosed in single quotation marks (‘) or in double quotation marks (“).
- What is the data type of “100”?
  - String but not number type

Strings example

```
<head> <script language="JavaScript">
            document.write("This is a string."+"<BR>");
            document.write("This string contains 'quote'.");
            var myString = "alert string";
            alert(myString); </script>  </head>
```

- Unlike Java and C, JavaScript does not have a single character (char) data type.



```
This is a string.
This string contains 'quote'.
```

alert string

typeof operator

```
<head> <script language="JavaScript">
       var x = "hello", y;
       alert("Variable x value is " + typeof(x));
       alert("Variable y value is " + typeof(y));
       alert("Variable x value is " + typeof(z));
</script> </head>
```

- It is an unary operator.
  - Return either: Number, string, Boolean, object, function, undefined, null

What is an Object?

- An object is a thing, anything, just as things in the real world.
  - E.g. (cars, dogs, money, books, …)
- In the web browser, objects are the browser window itself, forms, buttons, text boxes, …
- Methods are things that objects can do.
  - Cars can move, dogs can bark.
  - Window object can alert the user "alert()".
- All objects have properties.
  - Cars have wheels, dogs have fur.
  - Browser has a name and version number.

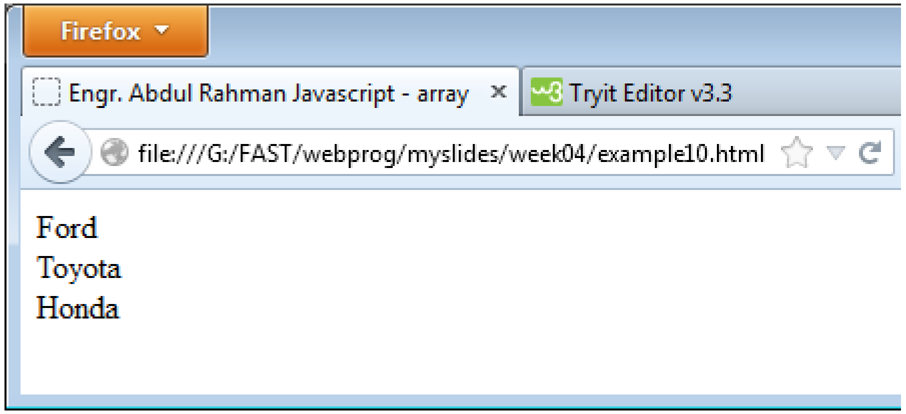
Array

- An Array contains a set of data represented by a single variable name.
- Arrays in JavaScript are represented by the Array Object, we need to "**new Array()**" to construct this object.
- The first element of the array is "Array[0]" until the last one Array[i-1].
- E.g. myArray = new Array(5)
  - We have myArray[0,1,2,3,4].

Array Example

```
<script language="JavaScript">
        Car = new Array(3);
        Car[0] = "Ford";
        Car[1] = "Toyota";
        Car[2] = "Honda";
        document.write(Car[0] + "<br>");
        document.write(Car[1] + "<br>");
        document.write(Car[2] + "<br>");
</script>
```

- You can also declare arrays with variable length.
  - arrayName = new Array();
  - arrayName.length = 0, allows automatic extension of the length.
  - Car[9] = "Ford"; Car[99] = "Honda";


Firefox
Engr. Abdul Rahman Javascript - array
Tryit Editor v3.3
file:///G:/FAST/webprog/myslides/week04/example10.html

```
Ford
Toyota
Honda
```

Dynamic Arrays

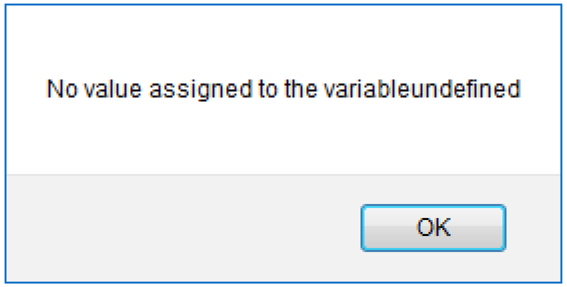
```
<head><script language="JavaScript">
         ar1 = new Array();
         ar1.length = 0
         ar1[0]="zero"; ar1[1]="one"; ar1[2]="two";
         document.write(ar1[0] + "<br>");
         document.write(ar1[1] + "<br>");
         document.write(ar1[2] + "<br>");
         var ar2 = [];
         ar2.length = 4;
         document.write("ar2 length = " + ar2.length + "<br>");
</script> </head>
```

Null & Undefined

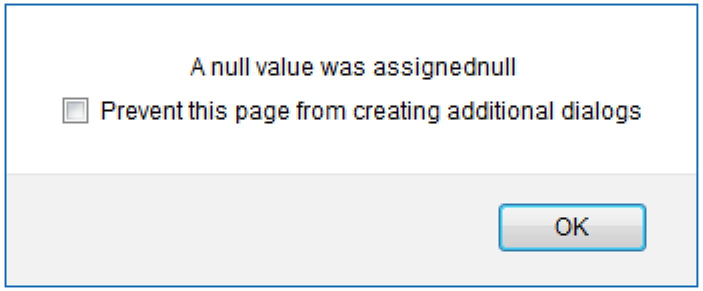
- An "undefined" value is returned when you attempt to use a <u>variable that has not been defined</u> or you have declared but you forgot to provide with a value.
- Null refers to "<u>nothing</u>"
- You can declare and define a variable as "null" if you want absolutely nothing in it, but you just don't want it to be "undefined".

Null & Undefined example

```
<html>
<head>
<title> Null and Undefined example </title>
<script language="JavaScript">
        var test1, test2 = null;
        alert("No value assigned to the variable" + test1);
        alert("A null value was assigned" + test2);
</script>
</head>
<body> … </body>
</html>
```



No value assigned to the variableundefined



A null value was assignednull

Prevent this page from creating additional dialogs

JavaScript Special Characters

Character	Meaning
\b	Backspace
\f	Form feed
\t	Horizontal tab
\n	New line
\r	Carriage return
\\	Backslash
\’	Single quote
\”	Double quote

Expressions

- It is a set of literals, variables, operators that merge and evaluate to a single value.
  - Left_operand *operator* right_operand
- By using different operators, you can create the following expressions.
  - Arithmetic, logical
  - String and conditional expressions.

Operators

- Arithmetic operators
- Logical operators
- Comparison operators
- String operators
- Bit-wise operators
- Assignment operators
- Conditional operators

Arithmetic operators

- left_operand "operator" right_operand

Operator	Name	Description	Example
+	Addition	Adds the operands	3 + 5
-	Subtraction	Subtracts the right operand from the left operand	5 - 3
*	Multiplication	Multiplies the operands	3 * 5
/	Division	Divides the left operand by the right operand	30 / 5
%	Modulus	Calculates the remainder	20 % 5

Unary Arithmetic Operators

- Binary operators take two operands.
- Unary type operators take only one operand.
- Which one add value first, and then assign value to the variable?

Name	Example
Post Incrementing operator	Counter++
Post Decrementing operator	Counter--
Pre Incrementing operator	++counter
Pre Decrementing operator	--counter

Logical operators

- Used to perform Boolean operations on Boolean operands

Operator	Name	Description	Example
&&	Logical and	Evaluate to “true” when both operands are true	3>2 && 5<2
||	Logical or	Evaluate to “true when either operand is true	3>1 || 2>5
!	Logical not	Evaluate to “true” when the operand is false	5 != 3

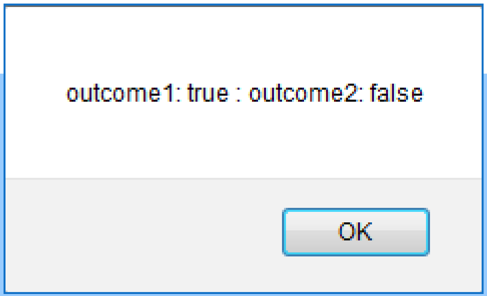
Comparison operators

- Used to compare two numerical values

Operator	Name	Description	Example
==	Equal	Perform type conversion before checking the equality	“5” == 5
===	Strictly equal	No type conversion before testing	“5” === 5
!=	Not equal	“true” when both operands are not equal	4 != 2
!==	Strictly not equal	No type conversion before testing nonequality	5 !== “5”
>	Greater than	“true” if left operand is greater than right operand	2 > 5
<	Less than	“true” if left operand is less than right operand	3 < 5
>=	Greater than or equal	“true” if left operand is greater than or equal to the right operand	5 >= 2
<=	Less than or equal	“true” if left operand is less than or equal to the right operand	5 <= 2

Strict Equality Operators

```
<script language="JavaScript">
        var currentWord="75";
        var currentValue=75;
        var outcome1=(currentWord == currentValue);
        var outcome2=(currentWord === currentValue);
        alert("outcome1: " + outcome1 + " : outcome2: " + outcome2);
</script>
```



```
outcome1: true : outcome2: false
```

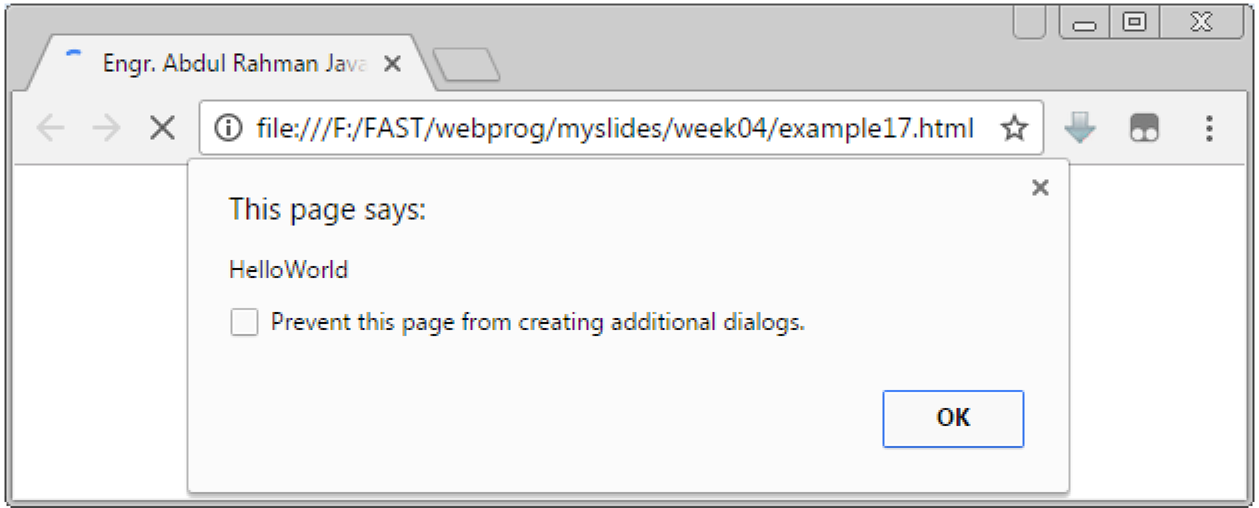
- Surprised that outcome1 is True!
- JavaScript tries every possibility to resolve numeric and string differences.

String operator

- (+) operator for joining two strings.

Operator	Name	Description	Return value
+	String concatenation	Joins two strings	“HelloWorld”

```
<script language="JavaScript">
        var myString = "";
        myString = "Hello" + "World";
        alert(myString);
</script>
```


Engr. Abdul Rahman Java

file:///F:/FAST/webprog/myslides/week04/example17.html

This page says:

HelloWorld

Prevent this page from creating additional dialogs.

OK

Bit Manipulation operators

- Perform operations on the bit representation of a value, such as shift left or right.

Operator	Name	Description
&	Bitwise AND	Examines each bit position
|	Bitwise OR	If either bit of the operands is 1, the result will be 1
^	Bitwise XOR	Set the result bit, only if either bit is 1, but not both
<<	Bitwise left shift	Shifts the bits of an expression to the left
>>	Bitwise signed right shift	Shifts the bits to the right, and maintains the sign
>>>	Bitwise zero-fill right shift	Shifts the bits of an expression to right

Assignment operators

- Used to assign values to variables

Operator	Description	Example
=	Assigns the value of the right operand to the left operand	A = 2
+=	Add the operands and assigns the result to the left operand	A += 5
-=	Subtracts the operands and assigns the result to the left operand	A -= 5
*=	Multiplies the operands and assigns the result to the left operand	A *= 5
/=	Divides the left operands by the right operand and assigns the result to the left operand	A /= 5
%=	Assigns the remainder to the left operand	A %= 2

The most common problem

```
<script language="JavaScript">
        if (alpha = beta) { … }
        if (alpha == beta) { … }
</script>
```

- Don't mix the comparison operator and the assignment operator.
- double equal sign (==) and the equal operator (=)

Order of Precedence

Precedence	Operator
1	Parentheses, function calls
2	~, -, ++, --, new, void, delete
3	*, /, %
4	+, -
5	<<, >>, >>>
6	<, <=, >, >=
7	==, !=, ===, !==
8	&
9	^
10	|
11	&&
12	||
13	?:
14	=, +=, -=, *=, …
15	The comma (,) operator

Precedence Example

Value = (19 % 4) / 1 - 1 - !false

Value = 3 / 1 - 1 - !false

Value = 3 / 1 - 1 - true

Value = 3 - 1 - true

Value = 3 - 2

Value = 1

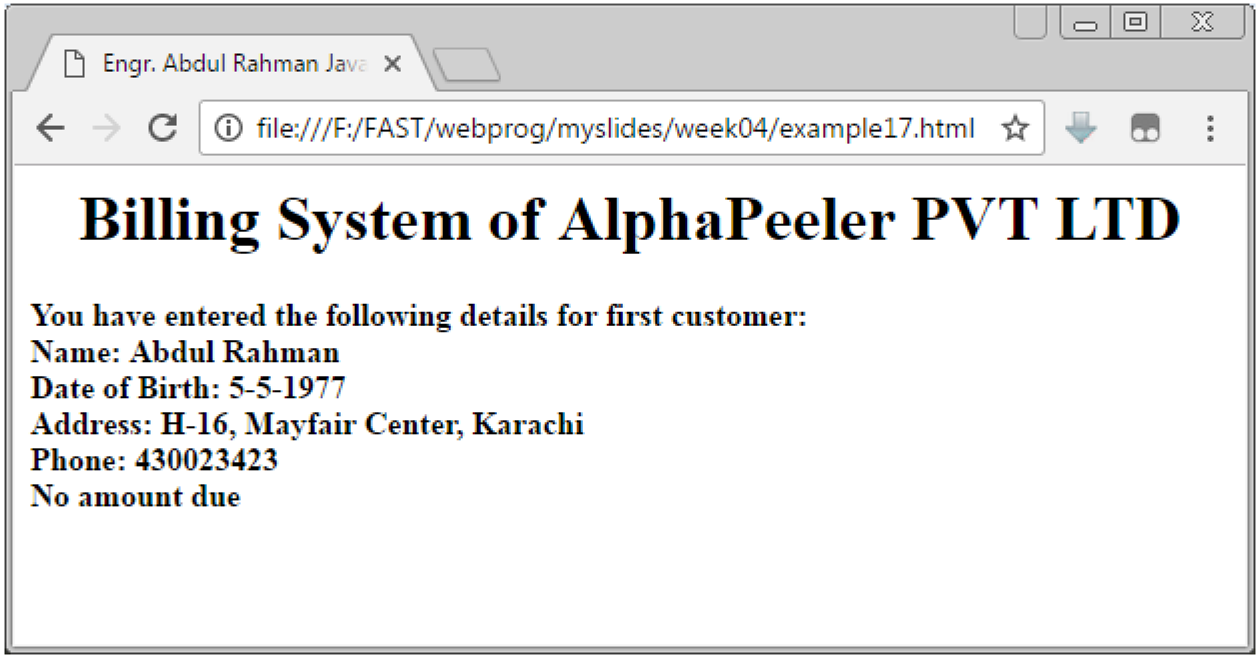
Scope of a Variable

- When you use a variable in a JavaScript program that uses functions.
- A global scope variable is one that is declared outside a function and is accessible in any part of your program.
- A local variable is declared inside a function and stops existing when the function ends.

Example of variable, data types

```
<html> <head><title>Engr. Abdul Rahman Javascript - Variable data types  </title>
</head> <body> <h1 align="center"> Billing System of AlphaPeeler PVT LTD </h1>
<script language="JavaScript">
firstCustomer = new Array();
billDetails = new Array(firstCustomer);
var custName, custDob, custAddress, custCity, custPhone;
var custAmount, custAmountPaid, custBalAmount;
custName=prompt("Enter the first customer's name:", "");
custDob=prompt("Enter the first customer's date of birth:", "");
custAddress=prompt("Enter the first customer's address:", "");
custPhone=prompt("Enter the first customer's phone number:", "");
custAmount=prompt("Enter the total bill amount of the first customer:", "");
custAmountPaid=prompt("Enter the amount paid by the first customer:", "");
custBalAmount = custAmount - custAmountPaid;
firstCustomer[0]=custName;
firstCustomer[1]=custDob;
firstCustomer[2]=custAddress;
firstCustomer[3]=custPhone;
firstCustomer[4]=custBalAmount;
document.write("<B>" + "Details for first customer:" + "<BR>");
document.write("Name: " + billDetails[0][0] + "<BR>");
document.write("Date of Birth: " + billDetails[0][1] + "<BR>");
document.write("Address: " + billDetails[0][2] + "<BR>");
document.write("Phone: " + billDetails[0][3] + "<BR>");
      (custBalAmount == 0) ? document.write("Amount Outstanding: " +
custBalAmount):document.write("No amount due")
</script> </body> </html>
```

Example of variable, data types



```
Billing System of AlphaPeeler PVT LTD

You have entered the following details for first customer:
Name: Abdul Rahman
Date of Birth: 5-5-1977
Address: H-16, Mayfair Center, Karachi
Phone: 430023423
No amount due
```

Conditional Statement

- “if” statement
- “if … else” statement
- “else if” statement
- “if/if … else” statement
- “switch” statement

“if” statement

```
if (condition) { statements; }
```

- It is the main conditional statement in JavaScript.
- The keyword “if” always appears in lowercase.
- The condition yields a logical true or false value.
- The condition is true, statements are executed.

“if” statement example

```
<script language="JavaScript">
var chr = "";
…
if (chr == 'A' || chr == 'O') {
        document.write("Vowel variable");
}
</script>
```

“if … else” statement

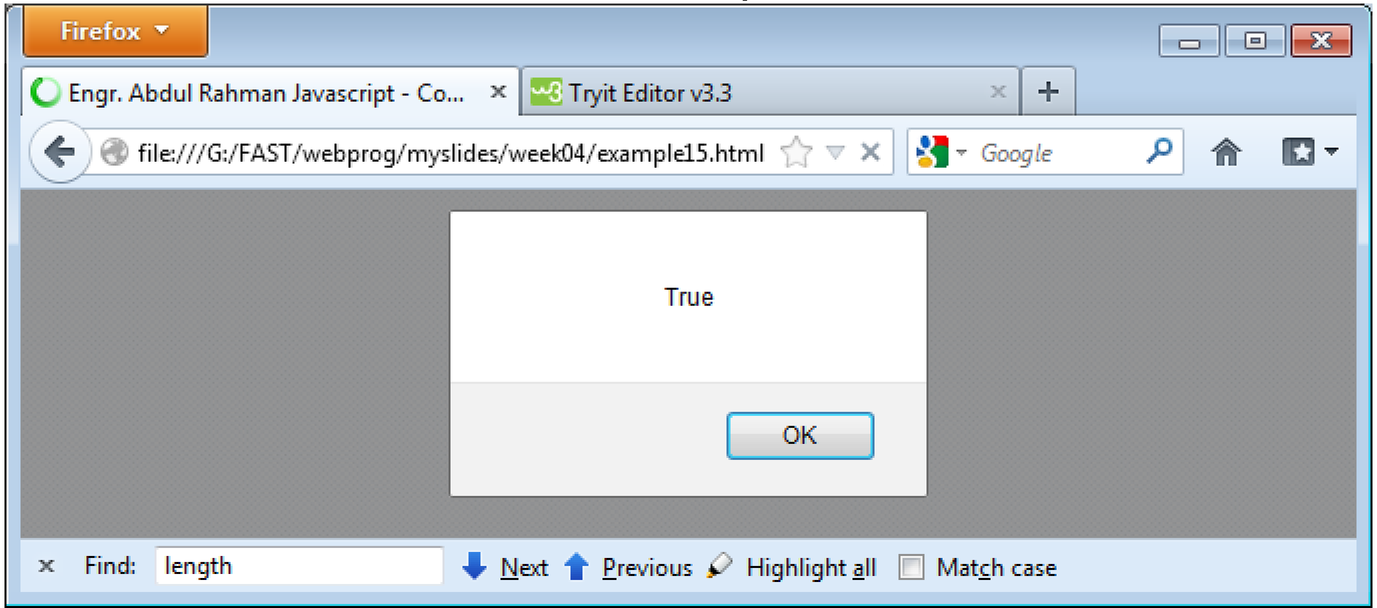
```
if (condition) { statements; }
else { statements; }
```

- You can include an “else” clause in an if statement when you want to execute some statements if the condition is false.

Ternary Shortcut (concise)

```
<script language="JavaScript">
If (3 > 2) {
        alert("True");
} else {
        alert("False");
}
(3 > 2) ? alert("True") : alert("False");
</script>
```

- Substitute for a simple "if/else" statement.


Firefox
Engr. Abdul Rahman Javascript - Co...
Tryit Editor v3.3
file:///G:/FAST/webprog/myslides/week04/example15.html
Google
Find: length
Next
Previous
Highlight all
Match case

True

OK

“else if” statement

```
if (condition) { statement; }
else if (condition) { statement; }
else { statement; }
```

- Allows you to test for multiple expression for one true value and executes a particular block of code.

“if/if…else” statement example

```
<script language="JavaScript">
var chr;
chr = prompt("Please enter a character : ","");
if (chr >= 'A'){
        if (chr <= 'Z')
                alert("Uppercase");
        else if (chr >= 'a'){
                alert("Lowercase");
        }
}
</script>
```

“switch” statement

```
switch (expression) {
        case label1:
                statements; break;
        default:
                statements;
}
```

- Allows you to merge several evaluation tests of the same variable into a single block of statements.

“switch” statement example

```
<script language="JavaScript">
var chr;
chr = prompt("Pls enter a character in lowercase:","");
switch(chr){
        case 'a' :
                alert("Vowel a"); break;
        case 'e' :
                alert("Vowel e"); break;
        default :
                alert("Not a vowel");
}
</script>
```

Looping Statement

- “for” Loops
- “for/in” Loops
- “while” Loops
- “do … while” Loops
- “break” statement
- “continue” statement

“for” statement

```
for (initial_expression; test_exp; change_exp)
{  statements;  }
```

- One of the most used and familiar loops is the for loop.
- It iterates through a sequence of statements for a number of times controlled by a condition.
- The change_exp determines how much has been added or subtracted from the counter variable.

“for” statement example

```
<script language="JavaScript">
var counter;
for (counter = 1; counter <= 10; counter++)
{
        document.write(counter*counter + " ");
}
</script>
```

- Display the square of numbers
- Output: 1 4 9 16 25 36 49 64 81 100

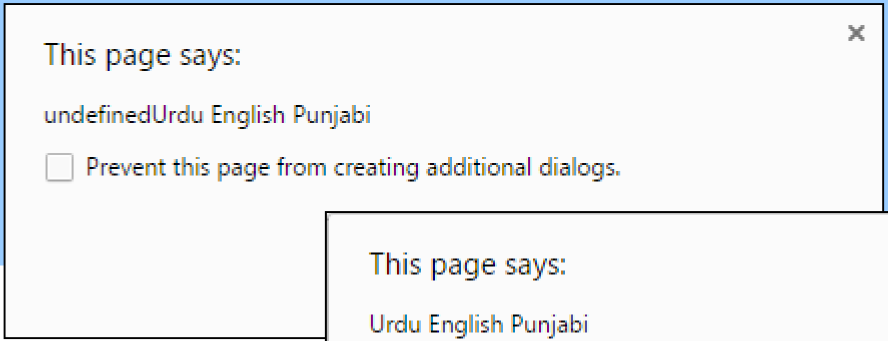
“for/in” statement

```
for (counter_variable in object)
{  statements;  }
```

- When the for/in statement is used, the counter and termination are determined by the length of the object.
- The statement begins with 0 as the initial value of the counter variable, terminates with all the properties of the objects have been exhausted.
  - E.g. array → no more elements found

“for/in” statement example

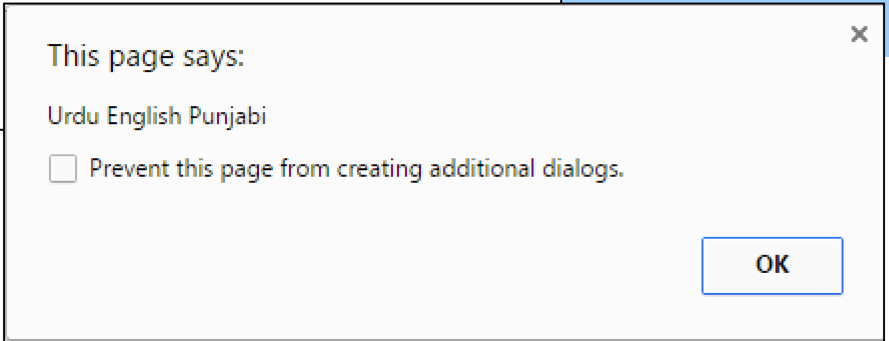
```
<script language=“JavaScript”>
var book;  (What is the difference if “var book=“”;)
var booklist = new Array(“Chinese”, “English”, “Jap”);
for (var counter in booklist) {
        book += booklist[counter] + “ “;
}
alert(book);
</script>
```



This page says:

undefinedUrdu English Punjabi

Prevent this page from creating additional dialogs.



This page says:

Urdu English Punjabi

Prevent this page from creating additional dialogs.

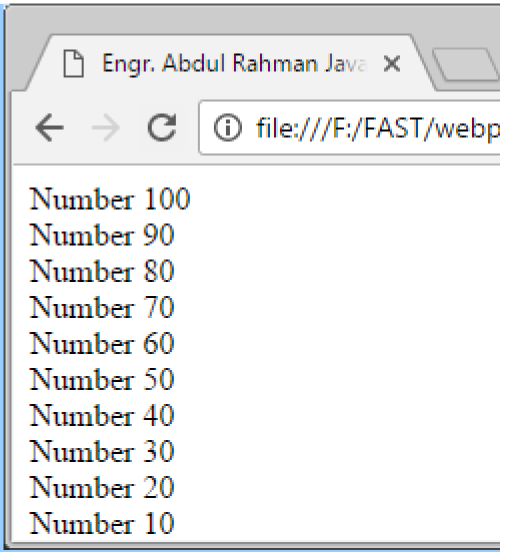
“while” statement

```
initial value declaration;
while (condition) {
        statements;
        increment/decrement statement;
}
```

- The while loop begins with a termination condition and keeps looping until the termination condition is met.
- The counter variable is managed by the context of the statements inside the curly braces.

“While” statement example

```
<html>
<head>
<title>While loop example</title>
<script language="JavaScript">
var counter = 100;
var numberlist = "";
while (counter > 0) {
      numberlist += "Number " + counter + "<br>";
      counter -= 10;
}
document.write(numberlist);
</script> <body> … </body>
</html>
```



```
Number 100
Number 90
Number 80
Number 70
Number 60
Number 50
Number 40
Number 30
Number 20
Number 10
```

“do … while” statement

```
do {
        statements;
        counter increment/decrement;
} while (termination condition)
```

- The do/while loop always executes statements in the loop in the first iteration of the loop.
- The termination condition is placed at the bottom of the loop.