WEB :: JavaScript

TalentSprint

Licensed To Skill

July 29, 2016

Learning Objectives

The content in this presentation is aimed at teaching learners to:

- Describe Java Script as a scripting language
- List the advantage of Java Script
- Explain fundamental elements of Java Script
- Define functions
- Deploy event driven code

JavaScript

What is Java Script?

- Is a scripting language produced by Netscape for use within HTML Web pages.
- Loosely based language and it is built into all the major modern browsers.
- Lightweight, interpreted programming language
- Open and cross-platform

JavaScript

Why Java Script?

To overcome HTML Problems:

- HTML cannot validate the data at client-side
- HTML is static, it cannot react to the events

JavaScript

Advantages of Java Script

- Provides HTML designers a programming tool.
- Can reacts to the events.
- Validates the data at client-side.
- Can read and change the content of an HTML element.

JavaScript

Syntax

```html
<html>
    <head>
        <script type = "text/javascript">
            document.write("<p>" + Date() + "</p>");
        </script>
    </head>
    <body>
        <h1> My First Web Page </h1>
    </body>
</html>
```

JavaScript

```
<script type = "text/javascript">
    document.write("<p>" + Date() + "</p>");
</script>
```

<script> Tag Attributes: language, type

JavaScript
First Java Script

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

Note

JavaScript ignores spaces, tabs, and newlines that appear in JavaScript programs.

JavaScript

Semicolons

- Statements in JavaScript are generally followed by a semicolon character, just as they are in C, C++, and Java.
- However, Javascript, allows you to omit this semicolon if your statements are each placed on a separate line.

code without semicolons:

```
<script language = "javascript" type = "text/javascript">
    var1 = 10
    var2 = 20
</script>
```

JavaScript

Semicolons

Semicolons are required when formatted in a single line.

Example

```
<script language = "javascript" type = "text/javascript">
    var1 = 10; var2 = 20;
</script>
```

It is a good programming practice to use semicolons.

JavaScript

Case Sensitivity

- JavaScript is a case-sensitive language.

- Language keywords, variables, function names, and any other identifiers must always be typed with a consistent capitalization of letters.

- For example

  Identifier: *Time and TIME will have different meanings in JavaScript.*

Note

Care should be taken while writing your variable and function names in JavaScript.

JavaScript

Comments

- Any text before a **//** and the end of a line is treated as a comment and is ignored by JavaScript. This is for single line comment.
- Any text between the characters **/*** and ***/** is treated as a comment. This is for multiple lines comments.

JavaScript

JavaScript code can be placed anywhere in an HTML document.

Script in `<head>`...`</head>` section.
Script in `<body>`...`</body>` section.
Script in `<body>`...`</body>` and `<head>`...`</head>` sections.
Script in and external file and then include in `<head>`...`</head>` section.

Note

The preferred way is to include it in the `<head>` ... `</head>` section.

JavaScript

Java Script Data Types

Three primitive data types:

Numbers   123, 120.50 etc.
             Example: var num = 2534;

Strings   "This text string" etc.
             Exapmle: var strString = "This is a string";

Boolean   true or false
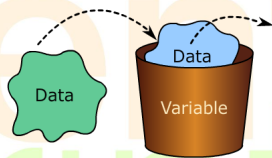   Example: var isMarried = true;

JavaScript

Java Script Data Types

Note

JavaScript is a loosely typed language, the declaration for a string variable, a number and a boolean is same. But it differentiates a string variable, a number and a boolean from the literal value assigned to it and the context of its use.

JavaScript

Java Script Variables

Variables can be thought of as named containers. You can place data into these containers and then refer to the data simply by naming the container.

JavaScript

Java Script Variables

Before you use a variable in a JavaScript program, you must declare it. Variables are declared with the var keyword.

```
<script type = "text/javascript">
    var money;
    var name;
</script>
```

JavaScript

Java Script Variable Scope

The scope of a variable is the region of your program in which it is defined. JavaScript variable will have only two scopes.

Global Variables A global variable has global scope which means it can be accessed any where in a JavaScript code.

JavaScript

Java Script Variable Scope

Local Variables   A local variable will be
                  visible only within a function
                  where it is defined.

> **Note**
> Function parameters are always local to
> that function.

JavaScript

JavaScript Variable Names

Rules for naming JavaScript variables:

- You should not use any of the JavaScript reserved keyword as variable name.
  Example: Break or Boolean variable names are not valid as variable names.
- Variable names are case-sensitive. Example, Name and name are two different variables.

JavaScript

JavaScript Variable Names

- Variable names should not start with numeral (0-9). They must begin with a letter or the underscore character. Example: 123test is an invalid variable name but _123test is a valid one.

JavaScript

Java Script Reserved Words

break	case	catch	continue	do
default	delete	else	finally	for
function	if	in	new	return
switch	this	throw	try	typeof
var	void	while		

Reserve words cannot be used as JavaScript variables, functions, methods, loop labels, or any object names as they have some special meaning.

JavaScript

What are Operators?

- An operator is a symbol that is used to perform an operation.
  + - * / % -- && >

- Operators generally work on variables or constants.

  Variables ← Operators → Constants

JavaScript

JavaScript Operators

Arithmetic Operators +, -, *, /, %, ++, --
Comparison Operators ==, !=, >, <, >=, <=
Logical Operators &&, ||, !
Assignment Operators =, +=, -+, *=, /=
Miscellaneous Opertors Conditional
            Operator (?)

JavaScript

Why Conditional Statements?

Very often, in our programs we will get a situation where in we have to execute some statements depending on some condition.

Example

Customer wants to withdraw Rs:5000
Balance availabel Rs:4000
ATM must display: Insufficient Funds

For above situations we use conditional statements like if, if-else.

JavaScript

Conditional Statements - Flow Control Structures

if - Statement: Syntax

if (expression) {
    Statement(s); // *executed if expression is true*
}

JavaScript

Conditional Statements - Flow Control Structures

if else - Statement: Syntax

```
if (expression) {
    Statement(s); // executed if expression is true
} else {
    Statement(s); // executed if expression is false
}
```

JavaScript

Switch Statement - Syntax

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

JavaScript

Conditional Statements - Looping Control Structures

while - Syntax

```
while (expression) {
    Statement(s); // executed if expression is true
}
```

JavaScript

Conditional Statements - Looping Control Structures

do..while - Syntax

```
do {
    Statement(s); // to be executed;
} while (expression);
```

JavaScript

Conditional Statements - Looping Control Structures

for - Syntax

for (initialization; test condition; iteration statement) {
    Statement(s); // *to be executed if test condition is true*
}

Advanced 'for' - Syntax

for (variablename in object) {
    statement or block to execute
}

JavaScript

How to define a function?

The most common way to define a function in JavaScript is by using the function keyword, followed by a unique function name, a list of parameters (that might be empty), and a statement block surrounded by curly braces.

JavaScript

Function - Syntax

```
<script type = "text/javascript">
    function functionname(parameter-list) {
        // statements
    }
</script>
```
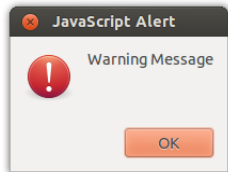
Syntax for calling a function

```
<script type = "text/javascript">
    sayHello();
</script>
```

JavaScript

Dialog Boxes

Alert Dialog Box Used to give a warning
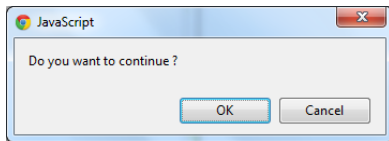message to the users

alert (" Warning Message");

JavaScript

Confirmation Dialog Box

- Used to take user's consent on any option.
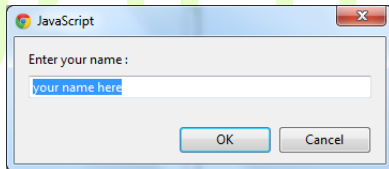- Displays a dialog box with two buttons: OK and Cancel.

confirm("Do you want to continue ?");

JavaScript

Prompt Dialog Box Prompts the user for a single input

```
var input = prompt("Enter your name : ", "your name here");
```

JavaScript

Page Printing

The JavaScript print function **window.print()** will print the current web page when executed.

```
<input type = "button" value = "Print" onclick = "window.print()"/>
```

JavaScript

Page Redirection

Redirects your site visitors to a new page.

```
window.location = "http://www.newlocation.com";
```

JavaScript