Add your logo here

JavaScript: Part A

Learning with Examples

1

A Quick Glance

- Introduction: What is JavaScript?
- Advantages: Why learn JavaScript?
- Programming: How and Where to write JavaScript code?
- Applications of JavaScript Programming.
- Discussion of different programming terminologies and syntax.
- Programming examples for each concept.

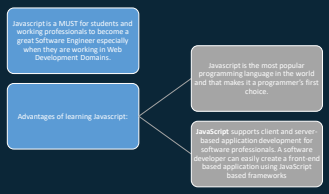
2

Introduction: What is Javascript?

- **JavaScript** is a lightweight, interpreted **programming** language.
- It is designed for creating network-centric applications.
- It is complimentary to and integrated with Java.
- **JavaScript** is very easy to implement because it is integrated with HTML.
- It is open and cross-platform.

3

Advantages: Why Learn Javascript?



4

Advantages: Why Learn Javascript? Cont…

Javascript is everywhere, it comes installed on every modern web browser and so to learn Javascript you really do not need any special environment setup. For example Chrome, Mozilla Firefox , Safari and every browser you know as of today, supports Javascript.

JavaScript usage has now extended to mobile app development, desktop app development, and game development. This opens many opportunities for you as a Javascript Programmer.

Great thing about Javascript is that you will find a lot of frameworks and Libraries ,already developed, which can be used directly in software development reducing the time to market.

5

How and Where to Write Program:

Writing and Playing with JavaScript Programing is simple.

Open Online JavaScript Editor.

If it is not working in your current browser change the browser and test.

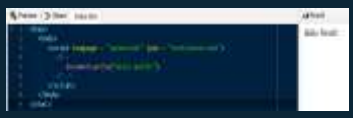
Once there, register yourself.

6

7

Hello World using Javascript

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



Reopen the Online JavaScript Editor and create a new program.
Write the code and add three document.write () to print
a) Your name
b) Your City name
c) Your email id.

8

Executing Code Using Online JavaScript Editor

9

Javascript frameworks

- There are many useful **Javascript frameworks** and libraries available:
  - Angular
  - React
  - jQuery
  - Vue.js
  - Ext.js
  - Ember.js
  - Meteor
  - Mithril
  - Node.js
  - Polymer
  - Aurelia
  - Backbone.js

10

Applications of Javascript Programming

Javascript is one of the most widely used **programming languages** (Front-end as well as Back-end).

It has its presence in almost every area of software development.

- Client-side validation.
- Manipulating HTML Pages.
- User Notifications.
- Back-end Data Loading.
- Server Applications.

11

Javascript Syntax

- Javascript can be implemented using Javascript statements that are placed within the <script>…</script> HTML tags in a web page.

```
<script…>
    JavaScript Code
</script>
```

12

Javascript Syntax Cont…

- The script tag takes two important attributes:
  - Language: specifies what scripting language is being used For example, HTML, XHTML
  - Type: specifies what is now recommended to indicate the scripting language in use and its value should be set to "text/javascript".

```
<script language = "javascript" type = "text/javascript">
   JavaScript code
</script>
```

13

Whitespace and Line breaks

Javascript ignores spaces, tabs, and newlines that appear in JavaScritp programs.

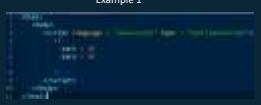
Spaces, tabs, and newlines can be freely used in programs and you are free to format and indent programs in a neat and consistent way that makes the code easy to read and understand.

14

Semicolons are optional

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

Example 1



Example 2

15

Case Sensitivity

JavaScript is a case-sensitive language.

It means that the language keywords, variables, function names, and any other identifiers must always be typed with a consistent capitalisation of letter.

For example: the identifiers Time and TIME will convey different meanings in JavaScript.

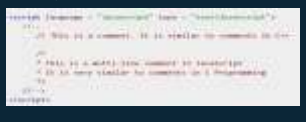
Note: care should be taken while writing variable and function names in JavaScript.

16

Comments in JavaScript

- JavaScript supports both C and C++ style comments.
- Any text between a // and the end of a line is treated as a comment and is ignored by JavaScript.
- Any text between the characters /* and */ is treated as a comment. This may span multiple lines.
- JavaScript also recognises the HTML comment opening sequence <!--. JavaScript treats this as a single-line comment, just as it does the // comment.
- The HTML comment closing sequence --> is not recognised by JavaScript so it should be written as //-->.

Example



17

JavaScript - Placement in HTML File

There is a flexibility given to include JavaScript code anywhere in an HTML document.

However the most preferred ways to include JavaScript in an HTML file are as follows –

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

18

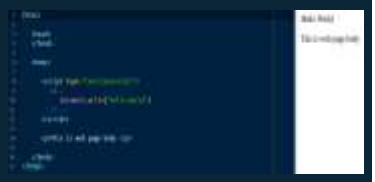
JavaScript in <head>...</head> section

- If you want to have a script run on some event, such as when a user clicks somewhere, then you will place that script in the head as follows –

19

JavaScript in <body>...</body> section

- If you need a script to run as the page loads so that the script generates content in the page, then the script goes in the <body> portion of the document.
- In this case, you would not have any function defined using JavaScript.



20

JavaScript in <body> and <head> Sections

- You can put your JavaScript code in <head> and <body> section altogether as follows –

21

JavaScript Output

- JavaScript can "display" data in different ways:

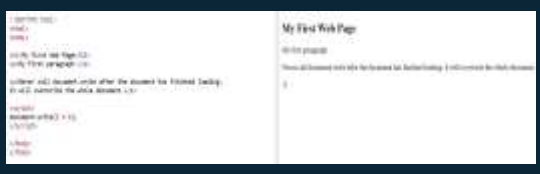
- Writing into an HTML element, using `innerHTML`.
- Writing into the HTML output using `document.write()`.
- Writing into an alert box, using `window.alert()`.
- Writing into the browser console, using `console.log()`.

22

Using innerHTML

- To access an HTML element, JavaScript can use the document.getElementById(id) method.
- The **id** attribute define the HTML elements.
- The innerHTML property defines the HTML content:

23



Using document.write()

For testing purposes, it is convenient to use **document.write()**

24

Example: Using document.write()

Open Online JavaScript Editor

25

Using window.alert()

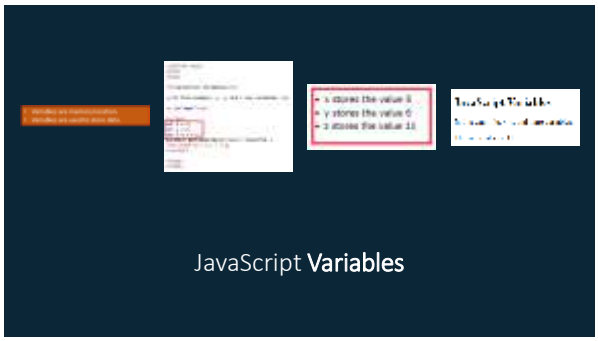
Open Online JavaScript Editor

26

Using console.log()

Open Online JavaScript Editor

27



28

JavaScript Identifiers

All JavaScript **variables** must be **identified** with **unique names**.

These unique names are called **identifiers**.

Identifiers can be short names (like x and y) or more descriptive names (age, sum, totalVolume).

The general rules for constructing names for variables (unique identifiers) are:

- Names can contain letters, digits, underscores, and dollar signs.
- Names must begin with a letter
- Names can also begin with $ and _ (but we will not use it in this tutorial)
- Names are case sensitive (y and Y are different variables)
- Reserved words (like JavaScript keywords) cannot be used as names

29

The Assignment Operator

var x = 100
- it is assignment of 100 at memory location of variable x.

var y == 100
- value of y is equal to 100.

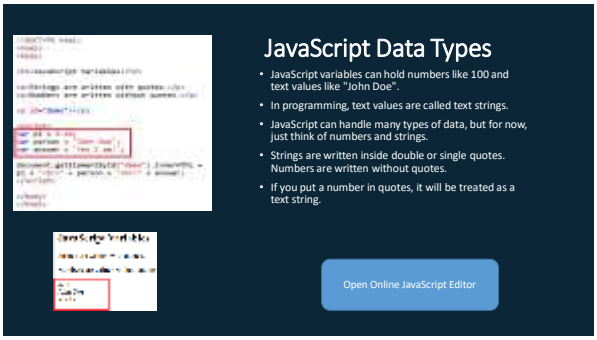
In JavaScript, the equal sign (=) is an "assignment" operator, not an "equal to" operator.

x = x + 5

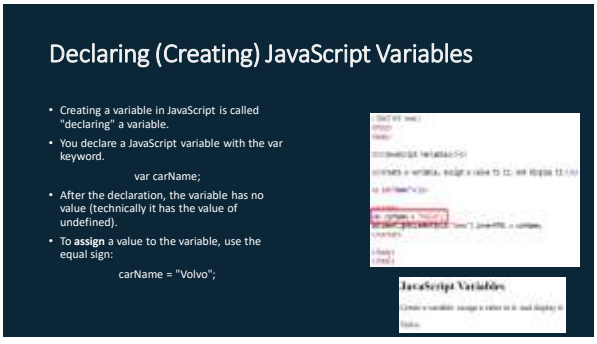
In JavaScript, however, it makes perfect sense: it assigns the value of **x + 5** to **x**.

The "equal to" operator is written like == in JavaScript.

30


JavaScript Data Types
• JavaScript variables can hold numbers like 100 and text values like "John Doe".
• In programming, text values are called text strings.
• JavaScript can handle many types of data, but for now, just think of numbers and strings.
• Strings are written inside double or single quotes. Numbers are written without quotes.
• If you put a number in quotes, it will be treated as a text string.
Open Online JavaScript Editor

31


Declaring (Creating) JavaScript Variables
• Creating a variable in JavaScript is called "declaring" a variable.
• You declare a JavaScript variable with the var keyword.
var carName;
• After the declaration, the variable has no value (technically it has the value of undefined).
• To **assign** a value to the variable, use the equal sign:
carName = "Volvo";

32


Well done
Completed

33