

# Client side scripting basics

**OBJECT<sup>TM</sup>**  
TECHNOLOGIES

# What will be covered

OBJECT

- Need of scripting language
- Introduction to Javascript
- JS features
- JS Syntax
- Application of Javascript
- Limitations of Javascript
- FirstJS
- Language Basics
  - Data types
  - Variables
  - Use of let and var
  - Use of const
  - typeof operator
  - Operators
  - Conditional Statement
  - Loops
  - Popup boxes
  - Functions in javascript

# Need of scripting language

## OBJECT

- Client-side scripting generally refers to the class of computer programs on the web that are executed on *client-side*, by the user's web browser.
- Using a client-side script for interactivity in a Web application usually provides better performance than using a server-side script to implement the same functionality.
- Client-side script does not incur the delay of a round trip between the client and the server, making the Web application more responsive.
- In addition, client-side scripting offers the benefit of off-loading part of the computational demands of an application from the server to the client.

# Introduction to Java Script

## OBJECT

- JavaScript is a lightweight interpreted programming language with object-oriented capabilities. The general purpose core of the language has been embedded in Netscape Navigator, IE and other web browsers.
- One of the most common misconceptions about JavaScript is that it is a simplified version of Java. The fact that both Java and JavaScript can provide executable content in web browsers, the two languages are entirely unrelated.
- The name JavaScript is owned by Netscape. The original version of the language, now almost obsolete, is JavaScript 1.0. The current version of the language is JavaScript 1.8.
- *Client-side JavaScript* extends the core language by supplying objects to control a browser (Navigator or another web browser) and its Document Object Model (DOM).

# JS Features

## OBJECT

- **Lightweight scripting language** - Since it is not a general-purpose language so it has a limited set of libraries. Also as it is only meant for client-side execution and that too for web applications, hence the lightweight nature of JavaScript is a great feature.
- **Dynamic typing** - In JavaScript we do not have to provide the data type while declaring a variable. Any type of value can be stored in the variable.
- **Object based Lanaguage** - Starting from ES6, the concept of class and OOPs has been more refined. Also, in JavaScript, two important principles with OOP in JavaScript are Object Creation patterns (Encapsulation) and Code Reuse patterns (Inheritance). Extensible
- **Easy syntax** - Most of the javascript control statements syntax is same as syntax of control statements in C language.

- **Interpreted Language** - JavaScript is an interpreted language which means the script written inside javascript is processed line by line. These Scripts are interpreted by JavaScript interpreter which is a built-in component of the Web browser.
- **Can be easily embedded in web page** - Javascript code can be written in the existing HTML page using script tag which makes HTML page dynamic.
- **Can catch events** - Javascript is designed to react to user actions like button click, checking the checkbox, keeping mouse pointer over certain element, loading of HTML page. Different functions can be called on occurrence of such events.

- **Interpreted Language** - JavaScript is an interpreted language which means the script written inside javascript is processed line by line. These Scripts are interpreted by JavaScript interpreter which is a built-in component of the Web browser.
- **Can be easily embedded in web page** - Javascript code can be written in the existing HTML page using script tag which makes HTML page dynamic.
- **Can catch events** - Javascript is designed to react to user actions like button click, checking the checkbox, keeping mouse pointer over certain element, loading of HTML page. Different functions can be called on occurrence of such events.

- JavaScript is a case-sensitive language.
- It follows the Hungarian notation
- JavaScript ignores spaces, tabs, and new lines that appear between tokens in programs.
- A token is a variable name, keyword, function name, number, or any other entity.
- JavaScript supports both C and C++ style comments.

# Applications of javascript

OBJECT

- JavaScript works on web users' computers — even when they are offline.
- JavaScript allows you to create highly responsive interfaces that improve the user experience and provide dynamic functionality, without having to wait for the server to react and show another page.
- JavaScript can load content into the document if and when the user needs it, without reloading the entire page — this is commonly referred to as Ajax.
- JavaScript can detect client's browser and react accordingly
- Java script can be used extensively for validation of data entered by user

# Limitations of javascript

OBJECT

- Client-side JavaScript does not allow the reading or writing of files.
- It cannot be used for networking applications because there is no such support available.
- It doesn't have any multithreading or multiprocessor capabilities.

# First JS Program

OBJECT

```
|<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
|<head>
|<meta charset="utf-8" />
<title></title>
|<script>
|    document.write("Hello From Javascript");
</script>
</head>
<body>
|    <script>
|        document.write("Hello From body section");
</script>
</body>
</html>
```

- JS statements should be written in script tag
- Script tag can be embedded anywhere in the html page and any no of times.
- Java script can be written in the external file

# First JS Program

OBJECT

```
<html>
<head>
<title>My First JavaScript Program</title>
<script src="myscripts.js"></script>
</head>
</html>
```

myscript.js

```
document.write("Hello from external javascript file");
```

- External JS files can be linked with html page using src attribute of script tag
- Scripts get executed in the order they are written

### Data Types

- **JavaScript does not require you to specify the type of the data contained in a variable**
- JavaScript supports the following types of values
- **number** – This consists of numbers including both integer and floating-point.
- **string** – This consists of text specified within single or double quotes.
- **null** – This consists of “null” value. The “null” value indicates that a variable is un-initialized
- **boolean** – This consists of Boolean values “true” and “false”
- **object** - This consists of key-value pairs of collection of data
- **array** - This consists of multiple values identified by diff indexes

### Variables

- Variable names can consists of alphabet , digit, Underscore character, Dollar sign character.
- Variable name should not begin with digit but can begin with alphabet, underscore character or dollar character
- For example
  - Sum1, total ,qty2\_1,\_count, Next\_Count, \$\$\$, \$mymoney
- The scope of the variable is the region of your program in which it is defined.
- A global variable has global scope – it is defined everywhere in your JavaScript code.

# Use of let and var

## OBJECT

- Variables are containers for storing data (values) and can be declared using var keyword.

```
var num;
```

- After the declaration, the variable has no value (technically it has the value of undefined)
- Value can be assigned at the time of declaration also.

```
var num = 20;
```

- Many variables can be declared in one statement.  

```
var num = 20, msg = "Welcome"
```
- If variable is redeclared, it loses its value  

```
var num = 20; //num will hold the value 20
```
- var num; //now num is defined

# Use of let and var

## OBJECT

- The let keyword was introduced in ES6
  - Variables defined with let cannot be Redeclared and must be Declared before use.

```
let x = 20;
```

let x = "hello"; //generates error, redeclaration in the same block is not allowed

- Before ES6 (2015), JavaScript had only Global Scope and Function Scope. ES6 introduced two important new JavaScript keywords: let and const. These two keywords provide Block Scope in JavaScript. Variables declared inside a {} block cannot be accessed from outside the block:

```
{  
    let x = 2;  
}  
  
//x can not be accessed here
```

- Variables declared inside a {} block using var can be accessed from outside the block.

# Use of const

## OBJECT

- JavaScript const variables must be assigned a value when they are declared:

```
const PI = 3.14159265359; //correct
```

```
const PI;
```

```
PI = 3.14159265359; //incorrect
```

- A const variable cannot be reassigned.

```
const PI = 3.141592653589793;
```

```
PI = 3.14; // This will give an error
```

- It does not define a constant value. It defines a constant reference to a value. So it is possible to change the array elements of const array. But can not re-assign

```
const numbers = [3,4,5,7];
```

```
numbers[2] = 20; //allowed
```

```
numbers = [5,6,78] //not allowed
```

# typeof operator

## OBJECT

- **typeof** is a JavaScript keyword that will return the type of a variable when you call it. You can use this to validate function parameters or check if variables are defined.

- **typeof** is important because JavaScript is a dynamically typed language.

- The **typeof** operator returns a string that represents the current type of a variable. You use it by typing `typeof(variable)` or `typeof` variable.

```
var x = 12345;  
console.log(typeof x) // number  
  
x = 'string';  
console.log(typeof x) // string  
  
x = { key: 'value' };  
console.log(typeof x) // object
```

# typeof operator

## OBJECT

```
var x = [1,2,3,4];
console.log(typeof x) // object
```

- This can be useful for checking the type of a variable in a function and continuing as appropriate.

```
function(x){
  if (typeof(x) === 'undefined'){
    console.log('variable x is not defined');
    return;
  }
  // continue with function here...
}
```

# Language Basics

## OBJECT

### Operators

Following operators are supported by Java Script

Category	Operators
Arithmetric Operators	+,-, *, /, %, ++,--
Assignment Operators	=, +=, -=, *=, /=, %=
Logical operators	&&,   , !
Comparison operators	<, >, <=, >=, ==, !=
String operators	+
Special operators	typeof
Conditional operator	: ? :

# Language Basics

OBJECT

## Flow Control

- If statement
- If/else statement
- Switch case
- Use of conditional operator

## Loops

- The loops in JavaScript are essentially similar to the ones we are familiar with in C.
- All three common loop types can be used with JavaScript:
  - ◆ while loops;
  - ◆ for loops;
  - ◆ do-while loops.
- **break** and **continue** can be used in JavaScript loops.



# JS Popup boxes

## OBJECT

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. When an alert box pops up, the user will have to click "OK" to proceed.

- **Confirm Box**

A confirm box is often used if you want the user to verify or accept something.

When a confirm box pops up, the user will have to click either "OK" or "Cancel" to proceed. If the user clicks "OK", the box returns true. If the user clicks "Cancel", the box returns false.

- **Prompt Box**

A prompt box is often used if you want the user to input a value before entering a page. When a prompt box pops up, the user will have to click either "OK" or "Cancel" to proceed after entering an input value. If the user clicks "OK" the box returns the input value. If the user clicks "Cancel" the box returns null.

# JS Popup boxes

## OBJECT

### Examples:

```
alert("I am an alert box!");
```

```
var r = confirm("Press a button");
if (r == true) {
    x = "You pressed OK!";
} else {
    x = "You pressed Cancel!";
}
```

```
var person = prompt("Please enter your name", "Harry Potter");
if (person != null) {
    document.getElementById("demo").innerHTML =
        "Hello " + person + "! How are you today?";
}
```

# Functions in javascript

## OBJECT

- JavaScript functions are used to perform operations.
- A JavaScript function is defined with the **function keyword**, followed by a name, followed by parentheses () .
- Function names can contain letters, digits, underscores, and dollar signs (same rules as variables).
- The parentheses may include parameter names separated by commas: (parameter1, parameter2, ...)
- The code to be executed, by the function, is placed inside curly brackets:

```
function functionName([arg1, arg2, ...argN]){
    //code to be executed
}
```
- The code inside the function will execute when "something" invokes (calls) the function:

When an event occurs (when a user clicks a button)  
When it is invoked (called) from JavaScript code  
Automatically (self invoked)

## Introduction to Javascript

So far we have seen about displaying static HTML but in today's scenario, more interactive and dynamic web sites are very common. Javascript is the client side scripting language which helps in making web page dynamic. In this chapter we will learn about very basics of Javascript language

### Client Side Scripting

Client-side scripting generally refers to the class of computer programs on the web that are executed client-side, by the user's web browser. Using a client-side script for interactivity in a Web application usually provides better performance than using a server-side script to implement the same functionality.

Client-side script does not incur the delay of a round trip between the client and the server, making the Web application more responsive. In addition, client-side scripting offers the benefit of off-loading part of the computational demands of an application from the server to the client.

In general client side scripting gives the benefit of utilizing computational power of client and avoids unnecessary request-response cycles between client and server which unnecessary occupy network bandwidth.

### Client side scripting languages :

Javascript, Jscript, VB script were different client side scripting languages used in early days of dynamic HTML.



### VB Script

Visual Basic Script is a lightweight active scripting language modeled on Visual basic. It is developed and maintained by Microsoft for the purpose of developing dynamic web applications.

### JavaScript

JavaScript is a language based on ECMAScript. A standard for scripting languages like JavaScript, JScript is ECMAScript. JavaScript is considered as one of the most popular implementations of ECMAScript.

### JScript

JScript released in 1996 by Microsoft, as Microsoft's dialect of ECMAScript. JScript and JavaScript are different names for the same language. They have different names to avoid trademark issues.

### TypeScript

It is a typed superset to javascript. It compiles into the regular javascript compiler. It is purely object-oriented and statically typed like Java. It is developed and maintained by Microsoft. It is a relatively new entrant into the market and is regularly getting updated

## TypeScript

It is a typed superset to javascript. It compiles into the regular javascript compiler. It is purely object-oriented and statically typed like Java. It is developed and maintained by Microsoft. It is a relatively new entrant into the market and is regularly getting updated.

Now-a-days the language that has dominated the client side scripting is **javascript**.  
**ECMAScript** is a JavaScript standard intended to ensure the interoperability of web pages across different browsers. The full form of ECMA is European Computer Manufacturer's Association. ECMAScript is a Standard for scripting languages such as JavaScript, JScript, etc. It is a trademark scripting language specification.

## Basics of Javascript :

JavaScript is a lightweight interpreted programming language with object-oriented capabilities.

The general purpose core of the language has been embedded in Netscape Navigator, IE and other web browsers.

One of the most common misconceptions about JavaScript is that it is a simplified version of Java. The fact that both Java and JavaScript can provide executable content in web browsers, the two languages are entirely unrelated. The name JavaScript is owned by Netscape.

The original version of the language, now almost obsolete, is JavaScript 1.0. The current version of the language is JavaScript 1.8.

Client-side JavaScript extends the core language by supplying objects to control a browser (Navigator or another web browser) and its Document Object Model (DOM).

## JS Features

**Lightweight scripting language** - Since it is not a general-purpose language so it has a limited set of libraries. Also as it is only meant for client-side execution and that too for web applications, hence the lightweight nature of JavaScript is a great feature.

**Dynamic typing** - In JavaScript we do not have to provide the data type while declaring a variable. Any type of value can be stored in the variable.

**Object based Lanaguage** - Starting from ES6, the concept of class and OOPs has been more refined. Also, in JavaScript, two important principles with OOP in JavaScript are Object Creation patterns (Encapsulation) and Code Reuse patterns (Inheritance). Extensible

**Easy syntax** - Most of the javascript control statements syntax is same as syntax of control statements in C language.

**Interpreted Language** - JavaScript is an interpreted language which means the script written inside Javascript is processed line by line. These Scripts are interpreted by JavaScript interpreter which is a built-in component of the Web browser.

**Easy syntax** - Most of the javascript control statements syntax is same as syntax of control statements in C language.

**Interpreted Language** - JavaScript is an interpreted language which means the script written inside javascript is processed line by line. These Scripts are interpreted by JavaScript interpreter which is a built-in component of the Web browser.

**Can be easily embedded in web page** - Javascript code can be written in the existing HTML page using script tag which makes HTML page dynamic.

**Can catch events** - Javascript is designed to react to user actions like button click, checking the checkbox, keeping mouse pointer over certain element, loading of HTML page. Different functions can be called on occurrence of such events.

**JavaScript is a case-sensitive language.** It follows the camel casing notation. JavaScript ignores spaces, tabs, and new lines that appear between tokens in programs. A token is a variable name, keyword, function name, number, or any other entity.

JavaScript supports both C and C++ style comments.

## Language Basics

**Data Types** : JavaScript does not require you to specify the type of the data contained in a variable JavaScript supports the following types of values :

- number – This consists of numbers including both integer and floating-point.
- string – This consists of text specified within single or double quotes.
- null – This consists of "null" value. The "null" value indicates that a variable is un-initialized
- boolean – This consists of Boolean values "true" and "false"
- object - This consists of key-value pairs of collection of data
- array - This consists of multiple values identified by diff indexes

**Variables** : Variable names can consists of alphabet , digit, Underscore character, Dollar sign character. Variable name should not begin with digit but can begin with alphabet, underscore character or dollar character For example Variable names are case sensitive. Some examples : Sum1 , total ,qty2\_1 , \_count, Next\_Count, \$\$\$, \$myMoney. The scope of the variable is the region of your program in which it is defined. A global variable has global scope – it is defined everywhere in your JavaScript code.

Use of var, let and const

## Operators

### Category

### Operators

Arithmetic Operators

`+,-,*,/,%,++,--`

Assignment Operators

`=,+=,-=,*=,/=%=`

Logical operators

`&&,||,!`

Comparison operators

`<,>,<=,>=,==,!==,!=, ===, !===`

String operators

`+`

Special operators

`typeof`

Conditional operator

`?:`



## Flow Control

If statement

If/else statement

Switch case

Use of conditional operator

Loops : The loops in JavaScript are essentially similar to the ones we are familiar with in C. All three common loop types can be used with JavaScript:

1. while loops
2. for loops
3. do-while loops.

**Loops :** The loops in JavaScript are essentially similar to the ones we are familiar with in C. All three common loop types can be used with JavaScript:

1. while loops
2. for loops
3. do-while loops.

break and continue can be used in JavaScript loops.

**JavaScript functions :** These are used to perform operations. A JavaScript function is defined with the function keyword, followed by a name, followed by parentheses (). Function names can contain letters, digits, underscores, and dollar signs (same rules as variables). The parentheses may include parameter names separated by commas: (parameter1, parameter2, ...). The code to be executed, by the function, is placed inside curly brackets:

```
function functionName([arg1, arg2, ...argN]){
    //code to be executed
}
```

The code inside the function will execute when "something" invokes (calls) the function: When an event occurs (when a user clicks a button) When it is invoked (called) from JavaScript code Automatically (self invoked)



### Assignments

1. Display message "Hello from Java Script" on the web page through java script.
2. Create a separate java script file and link with the web page. External java script should display a message "Hello from External Script" on the web page.
3. Display the pattern

```
1  
1 2  
1 2 3  
1 2 3 4  
1 2 3 4 5
```

4. Display whether the number is odd or even

5. Display whether number is prime or not. Write a separate function for this.

6. Write a function for calculating factorial of the given number. Call this function on the button click and display the result

## Assignments

1. Display message "Hello from Java Script" on the web page through java script.
2. Create a separate java script file and link with the web page. External java script should display a message "Hello from External Script" on the web page.

### 3. Display the pattern

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

4. Display whether the number is odd or even

5. Display whether number is prime or not. Write a separate function for this.

6. Write a function for calculating factorial of the given number. Call this function on the button click and display the result using

- a. document.write
- b. alert
- c. in a particular element

## Important Questions for: Introduction to Javascript

### Interview Questions

Do you know javascript? what is JS? what are the datatypes in javascript? what is null and undefined datatype in JS?

1. Explain var and let
2. What is document.load() ? Difference between document.load() and document.ready() ?
3. Significance of '\$' in JQuery other than identification as jquery function
4. JS datatypes
5. DOM
6. Different events in JS
7. AJAX
8. What is PHP ?