**Day 1**

**JavaScript Training**

https://[www.google.com](http://www.google.com) -🡪 URL

req (http/https)--------------------------🡪

Client Server

🡨---------------Res(http/https) html /html5

Static or dynamic

CSS/CSS3

JS : JavaScript

JavaScript was object based interpreter scripting language till ES5 JavaScript.

JavaScript is object oriented interpreter scripting language from ES6 onwards.

ECMA Script

Object : object is any real world entity

Properties or state

Person

Behaviour

Place

Bank

Car

Customer

Class to describe the object.

Function to describe the object.

Function are divided into two category

1. User defined function
2. Build in function or global function (aler(), prompt(), eval(), parseInt(), parseFloat() etc)

Closures function : closures is a type of inner function which access outer function scope after outer function execution terminate or close.

**Day 2:**

**Event : event is an interaction between user and DOM (Document Object Model) (Component) or event is delegation model.**

**In JavaScript all event start with prefix on followed by event name**

**Event provided bridge between html and javascript code.**

**onClick**

**ondblclick : button**

**onMouseOver**

**onMouseOut**

**onKeyUp**

**onKeyDown textfield, passwordfield, textarea**

**onChange drop down**

**onSubmit validation**

**onFocus when we enter in textfield**

**onBlur when we exit from textfield**

**onload when dom loaded or refresh**

**onUnload close the application**

**etc**

**Dom**

**User defined object**

**In literal style, function and class style**

**JavaScript object**

**Pre defined object**

**BOM : browser object model**

**DOM : document object model**

**Object --🡪 properties (fields) 🡪 window object in top most object in BOM**

**Behaviour**

**Object -🡪 property**

**Behaviour**

**Object -🡪 property**

**Behaviour**



**alert() or window.alert()**

**propmpt() or window.prompt()**

**eval()**

**document.write(“welcome”);**

**window.document.write(“welcome”);**

**DOM API (Document Object Model Application programming interface). Lot of programming language provided class, function or methods which help to read, write and update html contents dynamically.**

**Java, Asp.net, Python, JavaScript etc.**

**DOM Hierarchy**

**Index.html**

**Html**

**Head body**

**Script div content(textNode)**

**Style p**

**Meta**

**Title**

**Storage object part of html5 features**

**sessionStrorage.setItem(“key”,value);**

**localStrorage.setItem(“key”,value);**

**sessionStrorage.getItem(“key”);**

**localStorage.getItem(“key”);**

**sessionStrorag.remove(“key”);**

**localStorage.removeItem(“key”);**

**synchronous and asynchronous**

**synchronous statement**

**document.write(“first statement”);**

**document.write(“second statement”);**

**document.write(“third statement”);**

**asynchronous statement**

**document.write(“first statement”);**

**document.write(“second statement”); asynchronous**

**document.write(“third statement”); asynchronous**

**synchronous function call**

**fun1();**

**fun2()**

**fun3()**

**asynchronous function call**

**fun1();**

**fun2() asynchronous**

**fun3()**

**client server communication as synchronous**

**1st req**

**2nd req**

**3rd req**

**client Server**

**client server communication as asynchronous**

**1st req**

**2nd req**

**3rd req**

**client Server**

**AJAX : Asynchronous JavaScript and XML**

**XMLHttpRequest**

**ActiveXObject**

**In JavaScript 3 pre defined function part of window object**

**setTimeout() : 1st parameter callback function name or function body and 2nd parameter time.**

**setInterval()**

**clearInterval()**

**promise**

**fetch**

**Observable**

**User defined object**

**Day 3**

**JavaScript Training**

Promise : promise is pre defined object which help to handle asynchronous event of data.

Promise can be resolve or can rejected.

Creating user defined promise.

Giving the service for web application when both application running using different technologies.

Web Service

SOAP web service : in soap we can consume and produce the data only in the form of xml

Rest full web service In Rest full web service we can consume the data in any format base upon application like xml, json, text, html etc.

HDFC XML/JSON HSBC

Java Node JS

Fetch function return type is promise object.

TypeError

SyntaxError

RangeError

ReferenceError

In JavaScript we can create user defined object in three style

1. Literal style ES5
2. Function style ES5
3. Class style ES6

IN JSON (JavaScript Object Notation)

In Json we store data in key-value pairs. Key must be unique and it must be double quote. Value can be number type, string, Boolean, array , complex object.

JSON is a pre defined object.

Day 3 : 2nd session

Inheritance is use to inherits the properties as well as behaviour from old object to new object.

Constructor is special function which help to create the memory.

Function part of class or object creation in function style we need to call explicitly. But constructor get call automatically when we create the object.

Creating constructor we have give function name as constructor.

Generator : it is a type of function which help to pause and resume the function after yield.

function\* functionName() {

set of code

yield retrunValue;

set of code

yield returnValue;

set of code

yield returnValue;

}

Module : module is a like a package. Module is a combination of more than one variable, function, classes which have same name but different purpose.

Using module we can break our coding in different file and using import or require and export we can link both file together.

Method chaining or function chaining : one of the features provided by JavaScript which help reduce the size of the code and increate readability or we can avoid calling function.

Lexical Scope : lexical scope is the definition for area of an expression.

Design pattern : it is best practises or solution of repeating problem statement.

Core Design pattern divided into 3 types

GOF :

1. Creational design pattern

Object creation

Singleton design pattern

Factory design pattern

Abstract factory design pattern

Builder design pattern

Prototype design pattern

1. Structural design pattern

Creating structure of the application using class style or function style

Adapter design pattern

Proxy design pattern.

1. Behavioural design pattern

Communication between more than object.