**HTML/HTML5,CSS/CSS3, JavaScript Using ES5 and ES6**

**TypeScript**

**https://**[**www.google.com**](http://www.google.com)

http/ https : protocol : hyper text transfer protocol. Secure.

Set of rules which help to communicate more than one machine.

req(https/http)-🡪

**Client Server**

**🡨-----Res(http/https) HTML/HTML5**

**CSS/CSS3**

**JS**

**To do any validation or to read/write/update DOM (Document Object Model) all html tags are known as dom elements using JavaScript become more complex.**

**jQuery : jQuery is external library or external JavaScript file. jQuery contains lot of pre-defined function which internally connected to each other.**

**Using jQuery we can read, write and update DOM element dynamically.**

**jQuery is not a standard. Standard design pattern( best practise).**

**On View side we want to Framework.**

**Framework contains lot of pre-defined API which internally connected to each other to perform specific task.**

**Framework is known as template or protocol. But not final product.**

**If we develop any application using framework 60 to 70 task taken care by framework. Hardly we have to write 20 to 30% core to make final product.**

**Framework internally follow all design pattern. Implementation of all design pattern taken care by framework.**

**Angular is a front side or web application framework.**

**Angular 1.x or Angular JS:**

**Html, css, JavaScript using ES5 and ES6**

**Angular Framework 2 to 11**

**Html, css, JavaScript, TypeScript and Node JS.**

**Node JS 12.x to 16.x**

**VS Code**

**Node JS : Node JS is a run time environment for JavaScript library or framework.**

**Like JRE for Java program.**

**Node JS container lot of pre-defined modules. With the help of those modules we can create Server Side Scripting language.**

**HTML/HTML5, CSS/CSS3 and JavaScript -🡪 front end technologies.**

**Java, Python, Asp.net , php, Node JS etc ---🡪 backend technologies**

**Before node js JavaScript is known as Client side scripting language.**

**Node JS contains lot of pre-defined module may be local module or external module.**

**But After node JS Using Java Script we can create Server side programming language, We can create Rest Full Web Service, we can file handling programing, we can connect database may my sql or mongo db etc.**

**MEAN Stack**

**MERN Stack**

**Mongo DB /MySQL Express Module Angular Node JS**

**Mongo DB /MySQL Express Module ReactJS Node JS**

**npm ( Node Package Manager).**

**Using npm we can download the external module or dependencies.**

**Syntax**

**npm install –g moduleName (install globally)**

**Or**

**npm install moduleName (install locally)**

**In Node JS we can’t use document and window objects.**

**TypeScript : TypeScript is a type of scripting language which support all ES6 features.**

**ES : ECMA : European Computer Manufacture Association**

**ECMA is a concept.**

**JavaScript as well as TypeScript are the implementation of ES5 and ES6.**

**Programming is a concept**

**Using C, Java, Python**

**OOPs is a concept**

**C++, Java, Python are the implementation of OOPS language**

**JavaScript is a partial implementation of ES6.**

**TypeScript, it support all ES6 features.**

**Browser doesn’t support typescript program directly we have to convert TypeScript to JavaScript using transpiler. Converting one format to another format.**

**Compiler C**

**Interpreter JavaScript**

**Ts 🡪 JS Then we have to include js file in html page.**

**tsc (Transpiler)**

**TypeScript Features**

**To declare the variable in JavaScript (ES5) we are using var keyword.**

**From ES6 and TypeScript to declare the variable we are using var, let and const keyword.**

**Using var keyword we can to re-declaration same variable with same value or different values.**

**But using let keyword we can’t to re-declaration.**

**In C or Java**

**int a=10;**

**int a=40; error**

**using var we can declare global scope of that variable.**

**Using let we can declare local scope or block scope.**

**const keyword is use to declare the constant variable we can’t change the value.**

**DataType :**

**Typescript provide data type concept.**

**Syntax**

**let/var variableName:dataTypeName;**

**number, string, boolean,object, any etc.**

**let num:number =100;**

**num = “Ravi”; Error**

**array : array is use to store the same type of values.**

**literal style**

**let arrayName:datatype[];**

**generic style**

**let arrayName:Array<dataType>;**

**TypeScript functions**

**JavaScript function Using ES5**

**function add(a,b){**

**}**

**add(10,20);**

**add(“Ajay”, “Kumar”);**

**add(1);**

**add();**

**In ES5 JavaScript only name of the function must be match not mandatory number of parameter as well as type of parameter.**

**In ES5 JavaScript can return any type of value as well as not mandatory to return the value.**

**function info() {**

**//return “Welcome”;**

**//return 100;**

**return true;**

**}**

**Need function parameter which takes 0, 1 or many.**

**Rest operator or parameter is use to receive 0 or 1 or many arguments.**

**Syntax**

**…variableName:datatype[]**

**1,Ramesh,15000**

**2,Ravi,18000,C**

**3,Raju,25000,HTML5,CSS3,JavaScript**

**callback : callback function is use to pass the function body or function name or function itself another function as a parameter.**

**User-defined function**

**Pre-defined function**

**forEach() is a pre-defined function part of array which takes callback function as parameter.**

**OOPs :**

**object : any real world entity . concept**

**properties or state have variables/ fields**

**person**

**behaviour do/does functions / methods**

**bank**

**animal**

**car**

**upto ES5 there no class keyword.**

**function Employee() {**

**}**

**var emp = new Employee();**

**From ES6 and TypeScript we can use class keyword.**

**class : class is know as blue print of object or template of object.**

**constructor : It is a type of special function which help to create the objects.**

1. **To write constructor in class we have to use constructor keyword.**
2. **Constructor get call automatically when we create the object.**

**Constructor short cut initialization**

**Module : Using module we can achieve separation of concern.**

**Module is a collection of variable, function, classes, interfaces or enum etc.**

**Module are divided into different files base upon their functionality.**

**Using export and import we can connect one module to another module.**

**a.ts**

**fun1() {}**

**b.ts**

**fun2() {}**

**main.ts**

**calling fun1 and fun2 functions.**