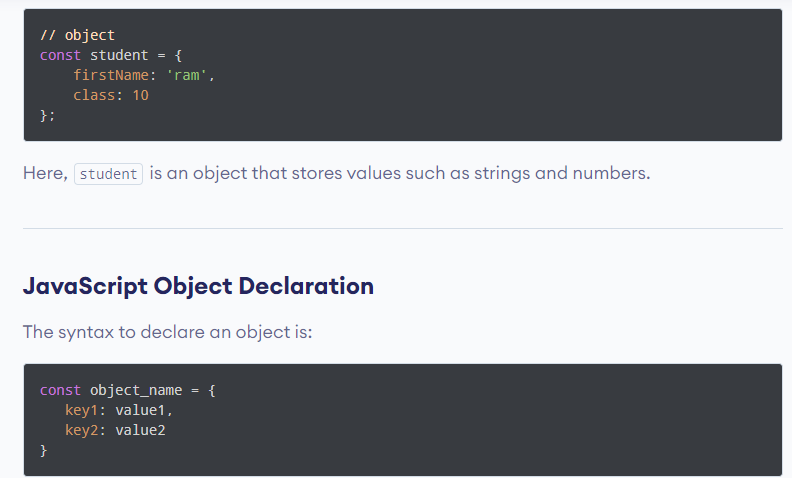
**WEEK :- 4**

**JAVASCRIPT OBJECTS**

What are objects in JS?

In JavaScript, an object is **a standalone entity, with properties and type**. Compare it with a cup, for example. A cup is an object, with properties. A cup has a color, a design, weight, a material it is made of, etc. The same way, JavaScript objects can have properties, which define their characteristics.

Here is an example of a JavaScript object



**value** pair separated by commas and enclosed in curly braces {}.

Here, an object object\_name is defined. Each member of an object is a **key:**

**JAVASCRIPT METHODS**

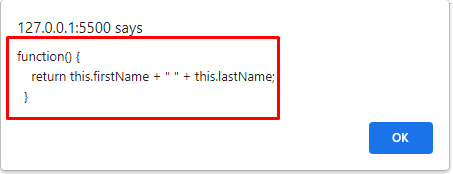
What are JS methods?

JavaScript methods are **actions that can be performed on objects**. A JavaScript method is a property containing a function definition.

What is method in JavaScript example?

**function definition**, for example, suppose you have a function that has firstName, lastName, regno, and a method with the name of fullName that returns the first name and the last name of a person i-e the full name of a person.

A Javascript method is **an action done on an object, and it is a property that holds a**



**JAVASCRIPT CONSTRUCTORS**

What is a JS constructor?

A constructor is **a special function that creates and initializes an object instance of a class**. In JavaScript, a constructor gets called when an object is created using the new keyword. The purpose of a constructor is to create a new object and set values for any existing object properties.

JavaScript Constructor Examples

In JavaScript, multiple objects can be created in a constructor: **//Constructor function User() { this.name = 'Bob'; } var user1 = new User(); var user2 = new User();** In the above example, two objects are created using the same constructor.



**JAVASCRIPT OBJECT PROPERTIES**

What is the object properties in javascript

Properties are **the values associated with a JavaScript object**. A JavaScript object is a collection of unordered properties. Properties can usually be changed, added, and deleted, but some are read only.

JS OBJECT PROPERTIES EXAMPLES

const person = {

fname:" John",

lname:" Doe",

age: 25

add person.nationality = "English";

};

Adding New Properties

You can add new properties to an existing object by simply giving it a value.

Assume that the person object already exists - you can then give it new properties:

const person = {

fname:" John",

lname:" Doe",

age: 25

nationality = "English";

};

## Deleting Properties

The delete keyword deletes a property from an object:

const person = {

firstName: "John",

lastName: "Doe",

age: 50,

eyeColor: "blue"

};

delete person.age;

AFTER DELETING

const person = {

firstName: "John",

lastName: "Doe",

eyeColor: "blue"

};

TYPES OF OBJECT PROPERTIES:

* What is data properties in JavaScript?

The data property **sets or returns the value of the data attribute of an <object> element**. The data attribute specifies the URL of the resource to be used by the object.

* What is object accessor in JavaScript?

There are two keywords which define the accessors functions: **a getter and setter for the fullName property**. When the property is accessed, the return value from the getter is used. When a value is set, the setter is called and passed the value that was set.

* What is prototype in JavaScript with example?

In JavaScript, **every function and object has a property named prototype by default**. For example, function Person () { this.name = 'John', this. age = 23 } const person = new Person(); // checking the prototype value console.

What are arrow functions in JavaScript?

Arrow functions were introduced in ES6. Arrow functions **allow us to write shorter function syntax**: let myFunction = (a, b) => a \* b;

Why arrow functions are used?

We may use arrow function syntax with our method associated with the array, like map (), reduce (), filter() since by using arrow function syntax instead of using normal function syntax one could easily read and understand as well as write the code more effectively.

What is a template string in JavaScript?

Template strings are a powerful feature of modern JavaScript released in ES6. **It lets us insert/interpolate variables and expressions into strings without needing to concatenate like in older versions of JavaScript**. It allows us to create strings that are complex and contain dynamic elements.

What is a prototype method in JavaScript?

Prototypes are **the mechanism by which JavaScript objects inherit features from one another**.

What are spread operator in JavaScript?

The spread operator is **a new addition to the set of operators in JavaScript ES6**. It takes in an iterable (e.g an array) and expands it into individual elements. The spread operator is commonly used to make shallow copies of JS objects. Using this operator makes the code concise and enhances its readability.

What is Map in JavaScript?

Map is **a collection of elements where each element is stored as a Key, value pair**. Map object can hold both objects and primitive values as either key or value. When we iterate over the map object it returns the key, value pair in the same order as inserted.

What is a set in JavaScript?

A JavaScript Set is **a collection of unique values**. Each value can only occur once in a Set. A Set can hold any value of any data type.



**INTRODUCTION TO TYPESCRIPT**

## What is TypeScript?

TypeScript is a syntactic superset of JavaScript which adds **static typing**.

This basically means that TypeScript adds syntax on top of JavaScript, allowing developers to add **types**.

TypeScript being a "Syntactic Superset" means that it shares the same base syntax as JavaScript, but adds something to it.

## Why should I use TypeScript?

JavaScript is a loosely typed language. It can be difficult to understand what types of data are being passed around in JavaScript.

In JavaScript, function parameters and variables don't have any information! So developers need to look at documentation, or guess based on the implementation.

TypeScript allows specifying the types of data being passed around within the code, and has the ability to report errors when the types don't match.

For example, TypeScript will report an error when passing a string into a function that expects a number. JavaScript will not.

**STEPS TO INSTALL TYPESCRIPT COMPILER**

# **TypeScript Installation**

In this section, we will learn how to install TypeScript, pre-requisites before installation of TypeScript and in how many ways we can install TypeScript.

### **Pre-requisite to install TypeScript**

1. Text Editor or IDE
2. Node.js Package Manager (npm)
3. The TypeScript compiler

### **Ways to install TypeScript**

There are two ways to install TypeScript:

1. Install TypeScript using Node.js Package Manager (npm).
2. Install the TypeScript plug-in in your IDE (Integrated Development Environment).

### **Install TypeScript using Node.js Package Manager (npm)**

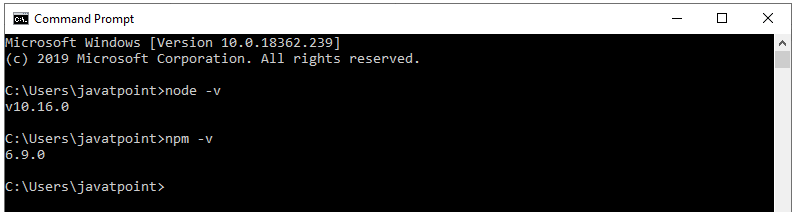
**Step-1** Install Node.js. It is used to setup TypeScript on our local computer.

To install Node.js on Windows, go to the following link: **[https://www.javatpoint.com/install-nodejs](https://www.javatpoint.com/install-nodejs" \t "_blank)**

To install Node.js in Linux/Ubuntu/CentOS, go to the following link: **[https://www.javatpoint.com/install-nodejs-on-linux-ubuntu-centos](https://www.javatpoint.com/install-nodejs-on-linux-ubuntu-centos" \t "_blank)**

To verify the installation was successful, enter the following command in the Terminal Window.

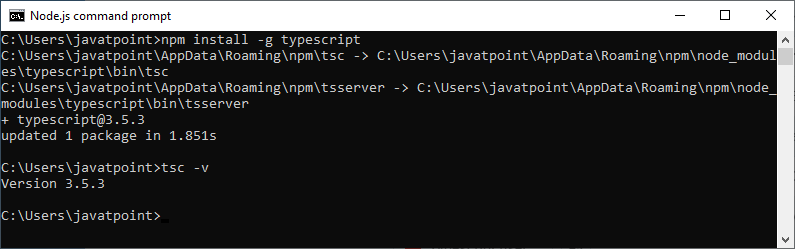
1. $ node -v
2. $ npm -v



**Step-2** Install TypeScript. To install TypeScript, enter the following command in the Terminal Window.

1. $ npm install typescript --save-dev         //As dev dependency
2. $ npm install typescript -g                      //Install as a global module
3. $ npm install typescript@latest -g          //Install latest if you have an older version

**Step-3** To verify the installation was successful, enter the command **$ tsc -v** in the Terminal Window.



### **Install TypeScript plug-in in your IDE**

**Step-1** Install IDE like Eclipse, Visual Studio, WebStorm, Atom, Sublime Text, etc. Here, we install Eclipse. To install Eclipse, go to the following link:

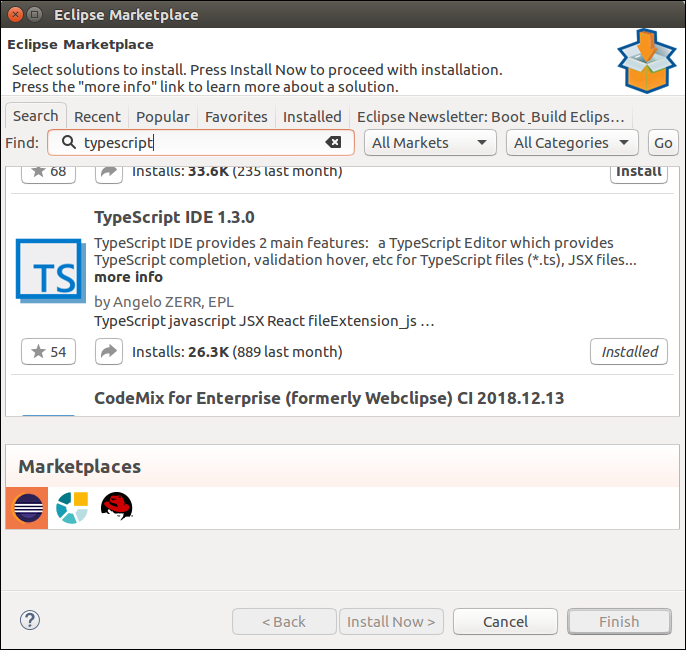
**In Windows:** **[https://www.javatpoint.com/javafx-how-to-install-eclipse](https://www.javatpoint.com/javafx-how-to-install-eclipse" \t "_blank)**

**In Ubantu:** **[https://www.javatpoint.com/how-to-install-eclipse-in-ubuntu](https://www.javatpoint.com/how-to-install-eclipse-in-ubuntu" \t "_blank)**

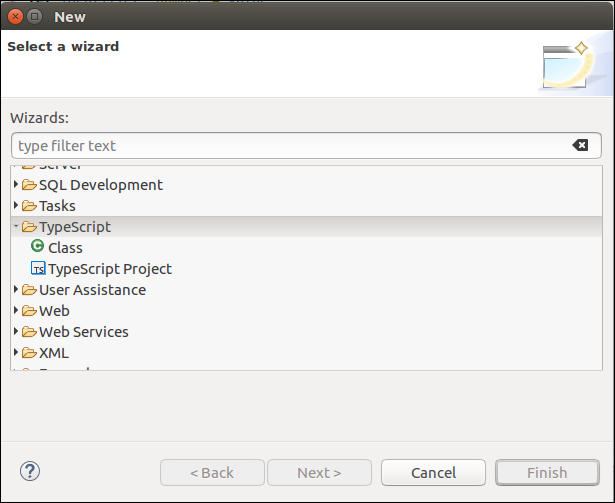
**In CentOS:** **[https://www.javatpoint.com/how-to-install-eclipse-on-centos](https://www.javatpoint.com/how-to-install-eclipse-on-centos" \t "_blank)**

**Step-2** Install TypeScript plug-in.

* Open Eclipse and go to **Help->Eclipse Market Place**.
* Search for **TypeScript** and choose **TypeScript IDE**, Click Install.



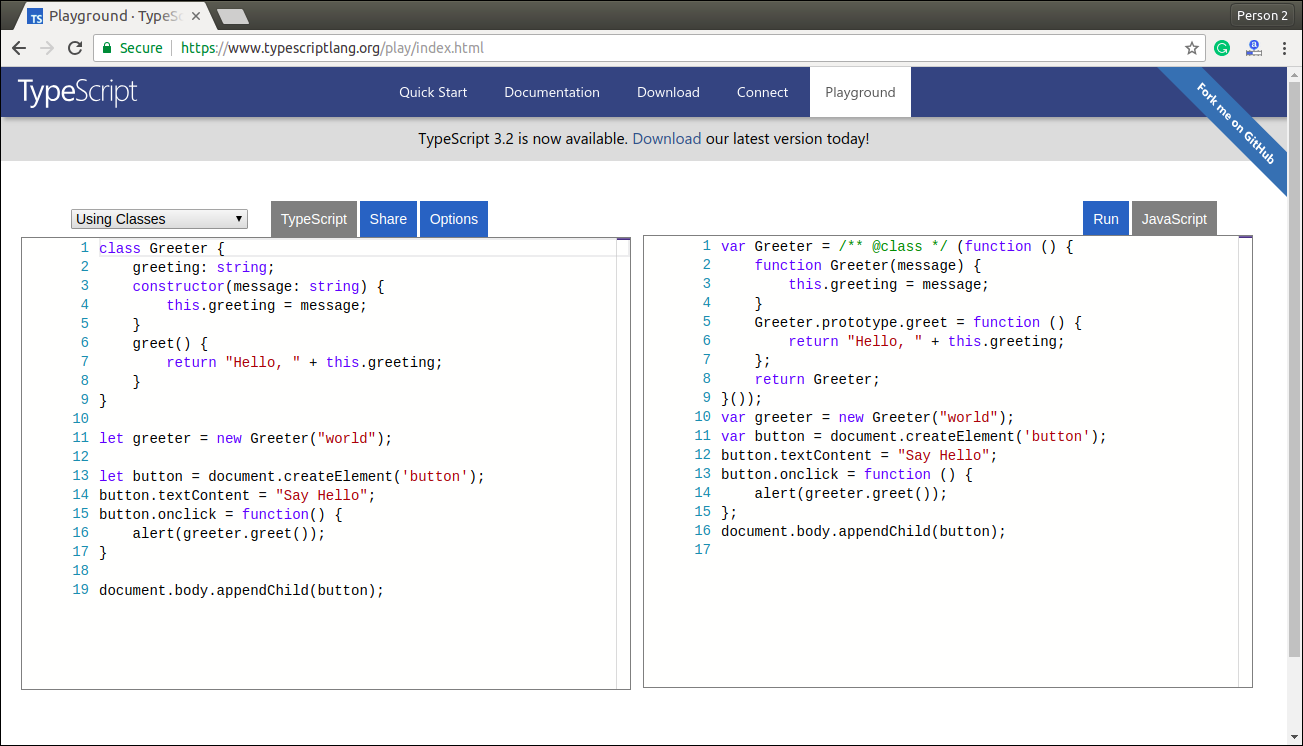
* In the next window, select **Features** which you want to install, and click **Confirm**.
* A new window will open, select **Accept Terms and Condition**, Click **Next**, and follow the on-screen instructions.
* Now **restart** Eclipse. To verify the TypeScript, go to **New->Other->TypeScript**. Once the TypeScript shows in the window, it means that TypeScript is successfully installed on your machine.



## Online Compiler for TypeScript

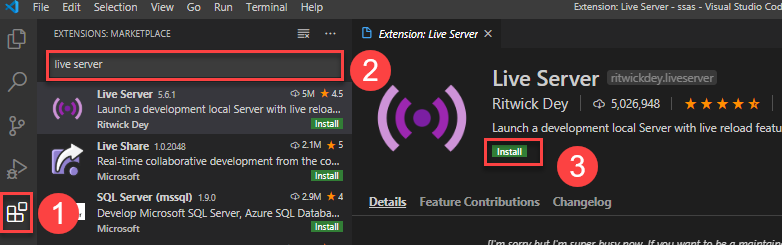
We can also run our script online with the official compiler. To do this, go to the below link. **[https://www.typescriptlang.org/play/index.html](https://www.typescriptlang.org/play/index.html" \t "_blank)**

The following screen appears. Now, you can do any TypeScript program on this.



**STEPS TO INSTALL LIVE SERVER**

To install the **Live Server** extension, you follow these steps:



* Click the **Extensions**tab to find the extensions for VS Code.
* Type the **live server** to search for it.
* Click the **install**button to install the extension.

**BASIC TYPES IN TYPESCRIPT**

TypeScript also provides basic data types to handle numbers, strings, etc. Some common data types in TypeScript are:

* Number
* String
* Boolean
* Enum
* Void
* Null
* Undefined
* Any
* Never
* Array
* Tuple

**CONTROL FLOW STATEMENT IN TYPESCRIPT**

TypeScript control statement is used to control the flow of program based on the specified condition.

## TypeScript control statements:

**1. If Statement  
2. If else statement  
3. if else if statement**

## TypeScript If Statement:

If statement is used to execute a block of statements if specified condition is true.

## Syntax:

|  |
| --- |
| **if**(condition){  *//Block of TypeScript statements.*  } |

## TypeScript If Else Statement:

If else statement is used to execute either of two block of statements depends upon the condition. If condition is true then if block will execute otherwise else block will execute.

## Syntax:

|  |
| --- |
| **if**(condition){  *//Block of TypeScript statements1.*  }**else**{  *//Block of TypeScript statements2.*  } |

## TypeScript If Else If Statement:

If else statement is used to execute one block of statements from many depends upon the condition. If condition1 is true then block of statements1 will be executed, else if condition2 is true block of statements2 is executed and so on. If no condition is true, then else block of statements will be executed.

## Syntax:

|  |
| --- |
| **if**(condition1){  *//Block of TypeScript statements1.*  }**else** **if**(condition2){  *//Block of TypeScript statements2.*  } . . . **else** **if**(conditionn){  *//Block of TypeScript statementsn.*  }**else**{  *//Block of TypeScript statements.*  } |

**FUNCTIONS IN TYPESCRIPT**

Functions are the building blocks of readable, maintainable, and reusable code. A function is a set of statements to perform a specific task. Functions organize the program into logical blocks of code. Once defined, functions may be called to access code. This makes the code reusable. Moreover, functions make it easy to read and maintain the program’s code.

In TypeScript, functions can be of two types: named and anonymous.

## Named Functions

A named function is one where you declare and call a function by its given name.

Example: Named Function

 Copy

function display() {

console.log("Hello TypeScript!");

}

display(); //Output: Hello TypeScript

Functions can also include parameter types and return type.

Example: Function with Parameter and Return Types

 Copy

function Sum(x: number, y: number) : number {

return x + y;

}

Sum(2,3); // returns 5

## Anonymous Function

An anonymous function is one which is defined as an expression. This expression is stored in a variable. So, the function itself does not have a name. These functions are invoked using the variable name that the function is stored in.

Example: Anonymous Function

 Copy

let greeting = function() {

console.log("Hello TypeScript!");

};

greeting(); //Output: Hello TypeScript!

An anonymous function can also include parameter types and return type.

Example: Function with Paramter and Return Types

 Copy

let Sum = function(x: number, y: number) : number

{

return x + y;

}

Sum(2,3); // returns 5