* TypeScript is an **open-source**, object-oriented programing language, which is developed and maintained by **Microsoft.**
* TypeScript was introduced by **Anders Hejlsberg**, a core member of the development team of C# language.
* TypeScript is a strongly typed **superset of JavaScript** which compiles to plain JavaScript.
* It contains all elements of the JavaScript. It is a language designed for large-scale JavaScript application development
* TypeScript can be executed on any **browser**, any **Host**, and any **Operating System**.
* TypeScript is the ES6 version of JavaScript with some additional features.

TypeScript Introduction

* TypeScript is not directly run on the browser. It needs a compiler to compile and generate in JavaScript file, which can run directly on the browser.
* The TypeScript source file is in ".ts" extension.
* TypeScript uses TSC (TypeScript Compiler) compiler, which convert Typescript code (.ts file) to JavaScript (.js file).



Why use TypeScript?

We use TypeScript because of the following benefits.

* TypeScript supports object-oriented programming features such as classes, interfaces, inheritance, generics, etc.
* TypeScript is fast, simple, and most importantly, easy to learn.
* TypeScript provides the error-checking feature at compilation time. It will compile the code, and if any error found, then it highlighted the mistakes before the script is run.
* TypeScript supports all JavaScript libraries because it is the superset of JavaScript.
* TypeScript support reusability because of the inheritance.
* TypeScript make app development quick and easy as possible, and the tooling support of TypeScript gives us autocompletion, type checking, and source documentation.
* TypeScript supports the latest JavaScript features, including ECMAScript 2015.
* TypeScript gives all the benefits of ES6 plus more productivity.

Text Editors with TypeScript Support

The TypeScript was initially supported only in Microsoft's Visual Studio platform. But today, there are a lot of text editors and IDEs available which either natively or through plugins have support for the TypeScript programming. Some of them are given below.

1. Visual Studio Code
2. Official Free Plugin for Sublime Text.
3. The latest version of WebStorm
4. It also supports in Vim, Atom, Emacs, and others.

# TypeScript Version

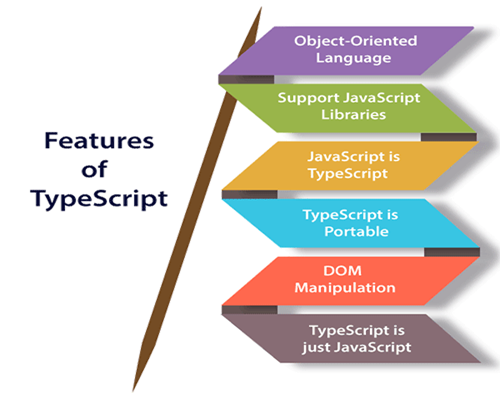
The complete release history for TypeScript is given below. You can also see the full documentation for recent releases on GitHub.

|  |  |  |  |
| --- | --- | --- | --- |
| **SN** | **Version** | **Release Date** | **Significant Changes** |
| **1.** | 0.8 | 01-10-2012 |  |
| **2.** | 0.9 | 18-06-2013 |  |
| **3.** | 1.1 | 06-10-2014 | Performance Improvements |
| **4.** | 1.3 | 12-11-2014 | Protected modifier, tuple types |
| **5.** | 1.4 | 20-01-2015 | union types, let and const declarations, type guards, type aliases, template strings |
| **6.** | 1.5 | 20-07-2015 | ES6 modules, decorators, for..of support, namespace keyword |
| **7.** | 1.6 | 16-09-2015 | JSX support, abstract classes and methods, local type declarations, intersection types, user-defined type guard functions |
| **8.** | 1.7 | 30-11-2015 | async and await support |
| **9.** | 1.8 | 22-02-2016 | constraints generic, control flow analysis errors, string literal types, allowJs |
| **10.** | 2.0 | 22-09-2016 | control flow based type analysis, null-and undefined-aware types, never type, discriminated union types, readonly keyword, type of this for functions |
| **11.** | 2.1 | 08-11-2016 | mapped types, keyof and lookup types, object spread and rest |
| **12.** | 2.2 | 22-02-2017 | object type, mix-in classes |
| **13.** | 2.3 | 27-04-2017 | async iteration, strict option, generic parameter defaults |
| **14.** | 2.4 | 27-06-2017 | dynamic import expressions, improved inference for generics, string enums, strict contravariance, for callback parameters |
| **15.** | 2.5 | 31-08-2017 | optional catch clause variables |
| **16.** | 2.6 | 31-10-2017 | strict function types |
| **17.** | 2.7 | 31-01-2018 | fixed length tuples, constant-named properties |
| **18.** | 2.8 | 27-03-2108 | improved keyof with intersection types, conditional types |
| **19.** | 2.9 | 14-05-2018 | support for symbols and numeric literals in keyof and mapped object types |
| **20.** | 3.0 | 30-07-2018 | project references, extracting and spreading parameter lists with tuples |
| **21.** | 3.1 | 27-09-2018 | mappable tuple and array types |
| **22.** | 3.2 | 30-09-2018 | stricter checking for bind, call, apply |
| **23.** | 3.3 | 31-01-2019 | Improved behavior on methods of union types, incremental builds for composite projects. |
| **24.** | 3.4 | 29-03-2019 | Faster incremental builds with the --incremental flag, type inference from generic functions, readonly modifier for arrays and tuples, const assertions, type-checking for globalThis. |
| **25.** | 3.5 | 29-05-2019 | Speed improvements, Improved excess property checks in union types, faster incremental builds, Omit helper type, smarter union type checking |

## TypeScript Vs. JavaScript

|  |  |
| --- | --- |
| **JavaScript** | **TypeScript** |
| It doesn't support strongly typed or static typing. | It supports strongly typed or static typing feature. |
| Netscape developed it in 1995. | Anders Hejlsberg developed it in 2012. |
| JavaScript source file is in ".js" extension. | TypeScript source file is in ".ts" extension. |
| It is directly run on the browser. | It is not directly run on the browser. |
| It is just a scripting language. | It supports object-oriented programming concept like classes, interfaces, inheritance, generics, etc. |
| It is interpreted language that's why it highlighted the errors at runtime. | It compiles the code and highlighted errors during  the development time. |
| JavaScript doesn't support modules. | TypeScript gives support for modules. |
| In this, number, string are the objects. | In this, number, string are the interface. |
| JavaScript doesn't support generics. | TypeScript supports generics. |

# Features of TypeScript



**Object-Oriented language:** TypeScript provides a complete feature of an object-oriented programming language such as classes, interfaces, inheritance, modules, etc. In TypeScript, we can write code for both client-side as well as server-side development.

**TypeScript supports JavaScript libraries:** TypeScript supports each JavaScript elements. It allows the developers to use existing JavaScript code with the TypeScript. Here, we can use all of the JavaScript frameworks, tools, and other libraries easily.

**JavaScript is TypeScript:** It means the code written in JavaScript with valid .js extension can be converted to TypeScript by changing the extension from .js to .ts and compiled with other TypeScript files.

**TypeScript is portable:** TypeScript is portable because it can be executed on any browsers, devices, or any operating systems. It can be run in any environment where JavaScript runs on. It is not specific to any virtual-machine for execution.

**DOM Manipulation:** TypeScript can be used to manipulate the DOM for adding or removing elements similar to JavaScript.

**TypeScript is just a JS:** TypeScript code is not executed on any browsers directly. The program written in TypeScript always starts with JavaScript and ends with JavaScript. Hence, we only need to know JavaScript to use it in TypeScript. The code written in TypeScript is compiled and converted into its JavaScript equivalent for the execution. This process is known as **Trans-piled**. With the help of JavaScript code, browsers can read the TypeScript code and display the output.

Advantage of TypeScript over JavaScript

* TypeScript always highlights errors at compilation time during the time of development, whereas JavaScript points out errors at the runtime.
* TypeScript supports strongly typed or static typing, whereas this is not in JavaScript.
* TypeScript runs on any browser or JavaScript engine.
* Great tooling supports with IntelliSense, which provides active hints as the code is added.
* It has a namespace concept by defining a module.

Disadvantage of TypeScript over JavaScript

* TypeScript takes a long time to compile the code.
* TypeScript does not support abstract classes.
* If we run the TypeScript application in the browser, a compilation step is required to transform TypeScript into JavaScript.

# 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://nodejs.org/en/download/>

To verify the installation was successful, enter the following command in the Terminal Window.How to find Nth Highest Salary in SQL

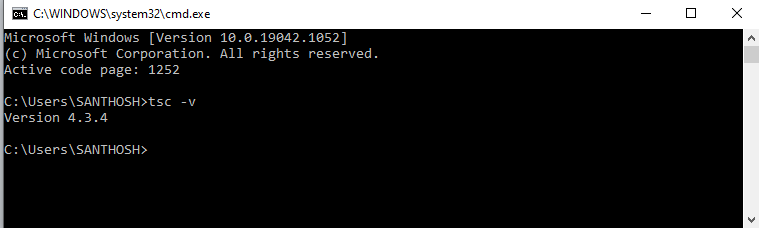
1. $ node -v
2. $ npm -v



**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.



## 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)