Difference between TypeScript and ES6

TypeScript

TypeScript is an **open-source** pure object-oriented programing language. It is a strongly typed **superset** of JavaScript which compiles to plain JavaScript. TypeScript is developed and maintained by **Microsoft** under the **Apache 2** license. It is not directly run on the browser. It needs a compiler to compile and generate in JavaScript file. TypeScript source file is in ".ts" extension. We can use any valid "**.js**" file by renaming it to ".ts" file. TypeScript is the ES6 version of JavaScript with some additional features.

History of TypeScript

**Anders Hejlsberg** developed TypeScript. It was first introduced for the public in the month of **1 October 2012**. After two years of internal development at Microsoft, the new version of TypeScript 0.9 was released in 2013. The current version of TypeScript is **TypeScript 3.4.5** which was released on **24 April 2019**.

ES6

ECMAScript (ES) is a **scripting language** specification standardized by **ECMA international**. It was created to standardize JavaScript. The ES scripting language contains many implementations, and the most popular is **JavaScript**. The developers use *ECMAScript* mostly for **client-side scripting** of World Wide Web (WWW).

The **sixth** edition of ECMAScript standard is ECMAScript6 or ES6 and later renamed as **ECMAScript 2015**. It is a major enhancement to the JavaScript language, which allows us to write programs for complex applications. It adds many features intended to make large-scale software development easier. The most common ES6 web-browsers are **Chrome** and **Firefox**. A **transpiler** converts the ES6 based code into **ES5** which is supported many browsers. TypeScript is a transpiler. Grunt, Gulp, and Babel are some other transpilers to compile the modules. Therefore, TypeScript supports ES6.

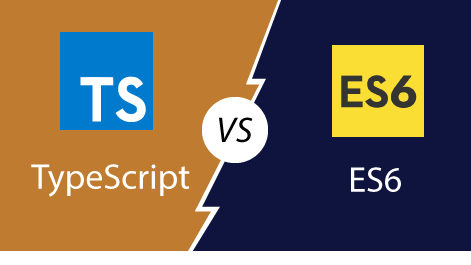
History

JavaScript introduced in **May 1995** by **Brendan Eich**. It was initially called **Mocha**, a name chosen by **Netscape** founder **Marc Andreessen**, and later renamed to **LiveScript**. At the same time, **Sun Microsystems** owned the trademark for JavaScript. In **December 1995**, Netscape acquired a trademark license and renamed it to JavaScript.

In between **1996** and **1997**, Netscape took JavaScript to the ECMA standards organization to maintain a specification for the language. In **June 1997**, the ECMA Technical Committee 39 (TC39) was created to continue to evolve the language, eventually releasing **ECMA-262 Ed.1**.

The first standard version of JavaScript was ECMAScript 1 was released on **June 1997**. After a year later, ECMAScript 2 was released, which contains only minor changes to keep a parallel ISO standard for JavaScript. In **December 1999**, ECMAScript 3 was released, which introduces a lot of popular features of JavaScript. In **December 2009**, ECMAScript or ES6 was published and subsequently renamed to ECMAScript 2015.

TypeScript vs. ES6



|  |  |  |
| --- | --- | --- |
|  | **TypeScript** | **ES6** |
| **Definition** | TypeScript is a free and open-source pure object-oriented programming language. It is developed and maintained by Microsoft. | ES6 is a version of ECMAScript (ES), which is a scripting language specification standardized by ECMA international. |
| **Explanation** | Typescript is to eradicate the development errors. | ES6 is comparatively more flexible in development time. |
| **Data-Types** | TypeScript supports all primitive data types. | ES6 does not support all data types. |
| **Features** | TypeScript contains features such as generics and type annotations, Inference, Enums, and Interfaces. | ES6 does not support these features. |
| **Scope** | Typescript has three scopes.   1. Global Scope 2. Class Scope 3. Local Scope | ES6 has two scopes.   1. Global Scope 2. Local Scope |
| **Decision-Making** | 1. if Statement 2. if-else Statement 3. else...if and nested if statements 4. switch Statement | 1. if Statement 2. if-else Statement 3. The else- if ladder/nested if statements. 4. switch?case Statement |
| **Modules** | TypeScript Modules are of two types:   1. Internal 2. External modules | We can classify the ES6 modules in two ways:   1. Importing a module 2. Exporting a module |
| **Loop** | Typescript and ES6 both are having same loops.   1. Definite 2. Indefinite | Typescript and ES6 both are having same loops.   1. Definite 2. Indefinite |
| **Why choose** | The developers choose TypeScript:   * Typesafe * JavaScript superset * Powerful type system, including generics & JS features. * Aligned with ES development for compatibility. * Structural, rather than nominal, subtyping. * Compile-time errors. * Starts and ends with JavaScript. | The developers choose ES6:   * ES6 code is shorter than traditional JS * Module System Standardized * Extremely compact * Destructuring Assignment |
| **Company using** | The list of companies which uses TypeScript are:   * Slack * Asana * CircleCI * Intuit * Swat.io * Avocode | The list of companies which uses ES6 are:   * Slack * StackShare * eBay * Asana * Intuit * Swat.io |