



¿Qué es TypeScript?

TypeScript es un lenguaje de programación de código abierto desarrollado por Microsoft, el cual cuenta con herramientas de programación orientada a objetos, muy favorable si se tienen proyectos grandes. Anders Hejlsberg, arquitecto principal del desarrollo del lenguaje de programación C#, es el principal participante en el desarrollo de este lenguaje.

Características

Principalmente se caracteriza por implementar las ventajas de un lenguaje orientado a objetos y las ventajas de JavaScript, permitiéndonos usar ambos a la vez (podríamos escribir la mitad en JavaScript y la otra en el lenguaje propio de TypeScript) y por supuesto el código que tuviésemos de JavaScript no se verá alterado, por lo que no hará falta modificar ninguna librería.

Este lenguaje de programación está especialmente indicado para desarrollar proyectos de una cierta envergadura, teniendo en cuenta la integración con distintos editores e IDE. Añade comprobación de tipos, de manera que permite utilizar IntelliSense para facilitar el desarrollo utilizando Visual Studio.

TypeScript, por supuesto, es open source y ofrecen herramientas para integrar el lenguaje de programación en distintas herramientas (Visual Studio, Sublime Text, Vi y Emacs), así como un paquete para Node.js. Las características que TypeScript agrega al desarrollo de JavaScript son pequeñas, pero generan grandes beneficios para los desarrolladores de .NET, que están familiarizados con características similares en los lenguajes que usan para el desarrollo habitual de aplicaciones de Windows.

Ventajas

- Se trata de un superconjunto de JavaScript
- Cada programa TypeScript tiene su correspondiente en JavaScript y viceversa
- Opcionalmente, puede proporcionar un mapeado de las fuentes que permita su depuración
- Incluye ya la inferencia de tipo en la declaración de variables y funciones. Se trataría de una de las columnas vertebrales del nuevo lenguaje
- Define declaraciones ambientes que son visibles en el ámbito del archivo TS, pero no se propagan en el archivo JS correspondiente, para así poder asimilarse a declaraciones explícitas en navegadores o en librerías específicas, como por ejemplo JQuery
- El lenguaje en sí es débilmente tipado –tipificado, si se prefiere- y se permiten declaraciones en tiempo de ejecución, de modo que no se atiene a una rigidez de declaraciones tales como en C# o Java
- Un módulo es un objeto totalmente anónimo, todo lo que se encuentre dentro de él estará oculto al mundo exterior, a menos que se haga visible de manera explícita. Dentro se pueden definir variables, funciones, objetos, clases e interfaces. En el correspondiente archivo JS se traducirán en componentes de un objeto encapsulado
- El soporte de clases abarca bastante de lo que cabría esperar –dentro de su simplicidad, que no es momento de tratar-, herencia, accesibilidad pública y privada y sobrecarga, básicamente.

Ejemplo hacer un hola mundo.

TypeScript Documentation Download Connect Playground

v3.6.3 Config Examples What's new Run Export Shortcuts About

```
1 const message: string = 'hello world';
2 console.log(message);
```

```
1 "use strict";
2 const message = 'hello world';
3 console.log(message);
```

Ejemplo de hacer una clase.

```
class Animal {
  name: string;
  constructor(theName: string) { this.name = theName; }
  move(distanceInMeters: number = 0) {
    console.log(`${this.name} moved ${distanceInMeters}m.`);
  }
}

class Snake extends Animal {
  constructor(name: string) { super(name); }
  move(distanceInMeters = 5) {
    console.log("Slithering...");
    super.move(distanceInMeters);
  }
}

class Horse extends Animal {
  constructor(name: string) { super(name); }
  move(distanceInMeters = 45) {
    console.log("Galloping...");
    super.move(distanceInMeters);
  }
}

let sam = new Snake("Sammy the Python");
let tom: Animal = new Horse("Tommy the Palomino");

sam.move();
tom.move(34);
```

```
1 "use strict";
2 class Animal {
3   constructor(theName) { this.name = theName; }
4   move(distanceInMeters = 0) {
5     console.log(`${this.name} moved ${distanceInMeters}m.`);
6   }
7 }
8 class Snake extends Animal {
9   constructor(name) { super(name); }
10  move(distanceInMeters = 5) {
11    console.log("Slithering...");
12    super.move(distanceInMeters);
13  }
14 }
15 class Horse extends Animal {
16   constructor(name) { super(name); }
17   move(distanceInMeters = 45) {
18     console.log("Galloping...");
19     super.move(distanceInMeters);
20   }
21 }
22 let sam = new Snake("Sammy the Python");
23 let tom = new Horse("Tommy the Palomino");
24 sam.move();
25 tom.move(34);
26
```

Conclusión

Aprender Typescript te introduce a un lenguaje de programación increíble, con herramientas para desarrollo aún mejores. Particularmente Visual Studio Code ofrecerá un número interesante de mejoras a tu flujo de desarrollo.

Además, algo de lo increíble de usar Typescript es que puedes usarlo en diferentes entornos y para el desarrollo de Aplicaciones nativas, híbridas, web, de escritorio e incluso servidores web.