Typescript

Ut riviera maya

Investigación

pablo de la cruz koyoc tinal

2018

Contenido

[Typescript 2](#_Toc511070953)

[Definición (Fernandez Guerrra, 2016) 2](#_Toc511070954)

[¿Qué es un superset? 2](#_Toc511070955)

[Por qué TypeScript 2](#_Toc511070956)

[Qué necesitamos para usar TypeScript 2](#_Toc511070957)

[Editores y TypeScript 3](#_Toc511070958)

[TypeScript Playground 3](#_Toc511070959)

[Referencias 4](#_Toc511070960)

# Typescript

## Definición (Fernandez Guerrra, 2016)

TypeScript es un lenguaje de programación de código abierto con herramientas de programación orientada a objetos, muy favorable si se tienen proyectos grandes.

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.

TypeScript convierte su código en Javascript común. Es llamado también Superset de Javascript, lo que significa que, si el navegador está basado en Javascript, este nunca llegará a saber que el código original fue realizado con TypeScript y ejecutará el Javascript como lenguaje original.

## ¿Qué es un superset?

Se trata de un lenguaje escrito sobre otro lenguaje. En este caso Typescript es eso, un lenguaje basado en el original, ofreciéndonos grandes beneficios como el descrito anteriormente, aunque existen otros beneficios. Por ejemplo, mientras otros superset de JavaScript nos alejan del código original, Typescript, por el contrario, es muy similar a Javascript y a C# gracias a que su creador posee conocimientos de ambos lenguajes.

Actualmente Angular 2, uno de los frameworks más famosos de JavaScript, está siendo desarrollando en TypeScript, para lo cual conocer este lenguaje será fundamental para entender y darle un mejor uso a la nueva versión de Angular.

## Por qué TypeScript

En concreto TypeScript nos ofrece muchas de las utilidades que se necesitan en JavaScript para poder convertirlo en un lenguaje escalable, a la altura de las necesidades más exigentes. TypeScript nos ofrece muchas de las cosas que los desarrolladores de lenguajes más tradicionales vienen usando en su día a día.

Una de las diferencias fundamentales es que TypeScript es verdaderamente orientado a objetos, trayendo herramientas como la herencia, sobrecarga, etc. En resumen, tiene cosas que suenan a lenguajes como Java, C++, C#, etc. Otro ejemplo clave es el del tipado estático. En el caso de TypeScript este tipado estático es opcional, pero obviamente su uso es muy recomendado y es una de las principales utilidades que nos van a facilitar mucho el trabajo y la depuración de los programas.

## Qué necesitamos para usar TypeScript

Básicamente necesitamos descargar dos programas. El primero es NodeJS, no porque necesitemos desarrollar con Node, sino porque el compilador de TypeScript está desarrollado en NodeJS.

Desde el sitio de NodeJS encontramos las opciones para la instalación en nuestro sistema operativo, por medio de un típico instalador con su asistente (siguiente, siguiente…) Más información en el Manual de NodeJS.

Luego necesitarás el TSC (Command-line TypeScript Compiler), la herramienta que nos permite compilar un archivo TypeScript a Javascript nativo. Este software es el que está realizado con NodeJS y su instalación se realiza vía npm con el siguiente comando:

***npm install -g typescript***

## Editores y TypeScript

Una de las cosas que debemos de conseguir cuando vamos a programar con TypeScript es un editor que lo soporte. El motivo es que los editores son capaces de compilar el código TypeScript a la vez que se está escribiendo, informando de errores en el código en el instante de su creación, lo que ahorra mucho tiempo y facilita el flujo de desarrollo.

Te informará de asuntos como:

* Una variable con tipado estático a la que se le intenta cargar un dato de otro tipo
* Una función que devuelve un valor de un tipo distinto al que debería
* Una función a la que no se le están pasando los valores de tipos correctos, o los objetos de clases correctas
* Un objeto sobre el que se intentan invocar métodos privados
* Y la lista no para de crecer…

Además, al tipar las variables seremos capaces de obtener muchas más ayudas "intellisense", como métodos invocables sobre cadenas o números, métodos que contiene un objeto, etc.

Existen editores que ya incorporan las ayudas para la programación en TypeScript, como Visual Studio o Visual Studio Code (Este último un editor ligero y gratuito para cualquier uso). Otros editores como Vim, Atom, Sublime Text o Brackets necesitarán la instalación de plugins especiales para poder aportar todas las ayudas de TypeScript.

## TypeScript Playground

Un sitio interesante para probar TypeScript sin necesidad de instalar nada, o para revisar rápidamente cómo un código TS compilaría a Javascript, es el TypeScript Playground.

Encontrarás dos cajas de texto, en la primera escribes código TypeScript y en tiempo real verás cómo este código compila a Javascript en la caja de la derecha. Además, encontrarás ayudas en la caja de la izquierda, donde escribes TS, indicando en qué partes de tu código se encuentran problemas y sus motivos.

Esta es una herramienta muy útil tanto para aprender TypeScript como para probar puntualmente un código que os recomendamos experimentar ya mismo. Tiene un desplegable muy útil porque ya te muestra varios ejemplos de código ya escritos, si es que quieres ver cómo funciona el compilador sin tener ni idea de cómo se escribe en TS.

# Referencias

Caceres, M. (s.f.). *DevCode*. Obtenido de https://devcode.la/blog/que-es-typescript/

Fernandez Guerrra, E. (2 de junio de 2016). *desarrolloweb*. Obtenido de https://desarrolloweb.com/articulos/introduccion-a-typescript.html