

TypeScript

Introducción a TypeScript

CertiDevs

Índice de contenidos

1. ¿Qué es TypeScript?	1
2. Propósito de TypeScript	1
3. Diferencias entre TypeScript y JavaScript	1
4. ¿Qué valor aporta TypeScript?	2
5. ¿Por qué Angular utiliza TypeScript en lugar de JavaScript?	2
6. Comparación con otros lenguajes	3
7. Instalación TypeScript	3
8. Compilador TypeScript	4
9. Recursos de aprendizaje	4

1. ¿Qué es TypeScript?

TypeScript es un lenguaje de programación de código abierto desarrollado y mantenido por **Microsoft**.

Fue diseñado por Anders Hejlsberg, quien también es conocido por su trabajo en el desarrollo de Turbo Pascal, Delphi y C#.

TypeScript fue anunciado por primera vez en octubre de 2012 y la primera versión estable fue lanzada en **junio de 2014**.

TypeScript es una **extensión tipada** de **JavaScript** que busca mejorar la experiencia de desarrollo y proporcionar características adicionales y de mayor seguridad en comparación con JavaScript puro o JavaScript vanilla.

2. Propósito de TypeScript

TypeScript es un superconjunto o superset de JavaScript, lo que significa que incluye todas las características de JavaScript y agrega otras adicionales, principalmente un **sistema de tipos estático**. Este sistema de tipos permite a los desarrolladores especificar el tipo de las variables, funciones y objetos en su código. Esto a su vez facilita la detección temprana de errores y mejora la legibilidad y el mantenimiento del código.

El **objetivo de TypeScript** es abordar los problemas y limitaciones de JavaScript, especialmente en lo que respecta a la escalabilidad y el mantenimiento del código en aplicaciones grandes y complejas. Aunque **JavaScript** ha evolucionado significativamente a lo largo de los años y ha adoptado muchas características modernas de programación, sigue siendo un lenguaje de programación **dinámico** y **débilmente tipado**, lo que puede conducir a problemas de mantenimiento y errores difíciles de detectar en aplicaciones más grandes.

TypeScript es **compatible** con todas las características de JavaScript y también incluye características adicionales, como interfaces, clases abstractas, decoradores y espacios de nombres (namespaces).

TypeScript es creado para ayudar a los desarrolladores a escribir código JavaScript más escalable y fácil de mantener.

3. Diferencias entre TypeScript y JavaScript

TypeScript es un superconjunto de JavaScript, lo que significa que incluye todas las características de JavaScript y también proporciona algunas mejoras y características adicionales:

- **Sistema de tipos estático:** TypeScript permite especificar tipos de datos para variables, funciones y objetos, lo que ayuda a detectar errores de tipo en tiempo de compilación en lugar de tiempo de ejecución.
- **Características de programación orientada a objetos (OOP) mejoradas:** TypeScript proporciona soporte avanzado para clases, interfaces, clases abstractas, herencia y otras

características de programación orientada a objetos.

- **Decoradores:** TypeScript admite decoradores, que son una función que se utiliza para modificar y mejorar clases, métodos, propiedades y parámetros.
- **Espacios de nombres:** TypeScript ofrece espacios de nombres para organizar y encapsular código relacionado en unidades lógicas.

4. ¿Qué valor aporta TypeScript?

TypeScript ofrece varios beneficios en comparación con JavaScript puro:

- **Detección temprana de errores:** El sistema de tipos de TypeScript permite identificar errores de tipo durante la compilación, lo que reduce la cantidad de errores en tiempo de ejecución y mejora la calidad del código.
- **Mejor legibilidad y mantenimiento del código:** El uso de tipos y otras características adicionales proporcionadas por TypeScript hace que el código sea más fácil de leer y mantener.
- **Escalabilidad:** TypeScript es particularmente útil en proyectos más grandes y complejos, donde un sistema de tipos y una estructura de código organizada pueden facilitar la gestión y el desarrollo del código.
- **Herramientas de desarrollo mejoradas:** TypeScript mejora el soporte para herramientas de desarrollo, como autocompletado, navegación por el código y refactorización.

5. ¿Por qué Angular utiliza TypeScript en lugar de JavaScript?

Angular es un framework de desarrollo web popular desarrollado por Google. A partir de Angular 2, el equipo de Angular decidió adoptar TypeScript como el lenguaje de programación principal para el desarrollo de aplicaciones Angular. Algunas de las razones por las que Angular utiliza TypeScript en lugar de JavaScript incluyen:

- **Mejor detección de errores:** El sistema de tipos estático de TypeScript permite identificar errores de tipo durante la compilación, lo que resulta en menos errores en tiempo de ejecución y una mayor calidad del código en general.
- **Desarrollo más eficiente:** TypeScript proporciona mejoras significativas en herramientas de desarrollo, como autocompletado, navegación por el código y refactorización. Estas mejoras pueden aumentar la productividad y facilitar el proceso de desarrollo.
- **Mejor legibilidad y mantenimiento del código:** El uso de tipos y otras características proporcionadas por TypeScript hace que el código sea más fácil de leer y mantener, lo cual es especialmente importante en aplicaciones más grandes y complejas, como las que se desarrollan con Angular.
- **Adopción de características futuras de JavaScript:** TypeScript adopta y proporciona soporte para características de JavaScript que aún no están ampliamente disponibles en todos los navegadores y entornos de ejecución. Esto permite a los desarrolladores de Angular utilizar las últimas características de JavaScript antes de que estén ampliamente disponibles.

6. Comparación con otros lenguajes

TypeScript se compara favorablemente con otros lenguajes de programación web populares, como JavaScript y Dart. Aunque TypeScript se basa en JavaScript y comparte muchas características con él, ofrece varias ventajas sobre JavaScript puro, como un sistema de tipos estático opcional, mejor soporte para programación orientada a objetos y características futuras de JavaScript.

Dart es otro lenguaje de programación web desarrollado por Google. Aunque Dart ofrece algunas características similares a TypeScript, como un sistema de tipos estático y soporte para programación orientada a objetos, TypeScript es más popular y ampliamente utilizado en la comunidad de desarrollo web. Además, TypeScript es más compatible con el ecosistema de JavaScript existente, ya que puede usar directamente bibliotecas y marcos de JavaScript en su código TypeScript.

En resumen, TypeScript es una extensión de JavaScript que proporciona un sistema de tipos estático opcional y características adicionales para mejorar la escalabilidad, legibilidad y mantenimiento del código. TypeScript es especialmente útil en aplicaciones más grandes y complejas, y es el lenguaje de programación principal utilizado por el framework Angular.

7. Instalación TypeScript

1. **Crear un nuevo proyecto:** Crea una nueva carpeta para tu proyecto y ábrela en Visual Studio Code.
2. **Inicializar el proyecto npm:** Abre la terminal integrada en VSCode y ejecuta el siguiente comando para generar un archivo `package.json`:

```
npm init -y # Si no tienes un archivo package.json en tu proyecto
```

1. **Instalar TypeScript a nivel de proyecto:** Ejecuta el siguiente comando para instalar TypeScript como una dependencia de desarrollo:

```
npm install -g typescript
```

1. **Instalar ts-node:** [ts-node](<https://github.com/TypeStrong/ts-node>) es una herramienta que permite ejecutar archivos TypeScript directamente sin necesidad de compilarlos previamente. Instala `ts-node` como una dependencia de desarrollo con el siguiente comando:

```
npm install --save-dev ts-node
```

1. Configurar Code Runner: abrir el archivo `settings.json` y donde `code-runner` `typescript` añadir:

```
"code-runner.executorMap": {  
  "typescript": "./node_modules/.bin/ts-node"
```

```
}
```

8. Compilador TypeScript

El compilador de TypeScript (tsc) se utiliza para compilar archivos TypeScript en archivos JavaScript compatibles con navegadores y entornos de ejecución de Node.js.

El compilador de TypeScript convierte el código TypeScript en código JavaScript compatible con versiones específicas de ECMAScript, como ES5 o ES6. Durante el proceso de compilación, el compilador también verifica si hay errores de tipo y otras inconsistencias en el código.

Para compilar un archivo TypeScript en un archivo JavaScript, ejecute el siguiente comando en su terminal:

```
tsc myfile.ts
```

Esto generará un archivo `myfile.js` en la misma ubicación que el archivo TypeScript original. Puede configurar el proceso de compilación utilizando un archivo de configuración `tsconfig.json` que especifique las opciones del compilador, como la versión de ECMAScript de destino, el manejo de errores y las rutas de salida de los archivos compilados.

9. Recursos de aprendizaje

Ver la [documentación oficial de TypeScript](#).