

Lección 1: ¿Qué es Angular y cómo funciona?



Lección 1: ¿Qué es Angular y cómo funciona?

Lectura: Introducción a TypeScript

Introducción a TypeScript



A medida que empieces a conocer un poco más sobre Angular 2, e iniciemos la codificación de algunos elementos, te darás cuenta que existen algunos elementos en el lenguaje diferentes a lo que habíamos trabajado en JavaScript. La explicación para esto es que el lenguaje que usamos para codificar aplicaciones en Angular 2 técnicamente no es JavaScript sino TypeScript. ¿Sabes qué es?

¿Qué es TypeScript?

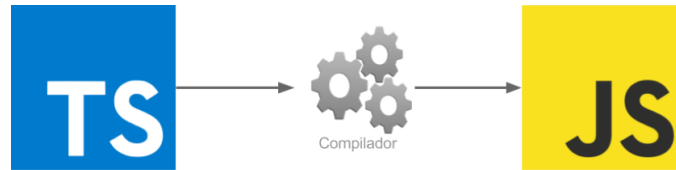
TypeScript es un lenguaje de programación creado por Microsoft y de código abierto, que comprende un set de características adicionales a JavaScript para la creación de scripts Web. En resumen, podríamos decir que es un complemento de JavaScript que añade unas cuantas herramientas adicionales de gran uso para los desarrolladores.

Compilación

¡Chatea con un tutor!



Tal vez te haya surgido la siguiente pregunta: si el único lenguaje que es interpretado por todos los navegadores es JavaScript, ¿cómo hace TypeScript para funcionar? La respuesta está en la misma pregunta, es decir, los archivos TypeScript se compilan a JavaScript plano para después ser interpretados por los navegadores. El compilador de TypeScript convierte el código propio de este lenguaje en un equivalente de JavaScript ECMAScript 3 o superior, esto quiere decir que todo código JavaScript funciona correctamente en un archivo TypeScript.



Diferencias entre TypeScript y JavaScript plano

Como ya lo mencionamos, TypeScript corresponde a un set adicional de herramientas para JavaScript. Dentro de las principales novedades que se encuentran en este lenguaje están:

- Imports: la manera de importar código desde otro módulo es muy sencilla, haciendo que la modularidad en los proyectos sea fluida y eficiente.

```
import { Component } from '@angular/core';
```

- Decoradores: los decoradores son funciones adjuntas a un objeto que realizan acciones en segundo plano y modifican el objeto al que están unidos para un fin específico.

```
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
```

- Clases: la forma en que TypeScript define las clases permite que la programación orientada a objetos sea muy sencilla y correcta.

```
export class AppComponent {
  title = 'Hola mundo!';
}
```

- Interfaces: elemento que no existe en JavaScript, sirve para forzar la implementación de atributos y métodos en una clase determinada.

¡Chatea con un tutor!



Lección 1: ¿Qué es Angular y cómo funciona?

```
interface Persona {  
  Nombre: string;  
  Apellido1: string;  
  Apellido2: string;  
}
```



- Tipos: como bien sabes, en JavaScript no se definen los tipos de datos para las variables creadas, y por lo tanto, éstas pueden tener cualquier tipo de valor. TypeScript añade un nivel adicional de seguridad en el código mediante la asignación de tipos a las variables o atributos de un objeto. De allí viene su nombre, ya que Type en inglés es Tipo.

```
function informacion(ciudad:string, habitantes:number){  
  return "Hay " + habitantes + " habitantes en " + ciudad;  
}
```

¿Por qué usar TypeScript?

Si el código TypeScript es compilado en JavaScript plano, ¿por qué no hacerlo todo en JavaScript desde el principio? En realidad hay muchas razones por las cuales se usa TypeScript en el desarrollo de aplicaciones con Angular 2:

1. Lenguaje fuertemente tipado: TypeScript añade tipos de datos a variables y a atributos de objetos, esto aumenta el nivel de seguridad en el código evitando que valores inapropiados

