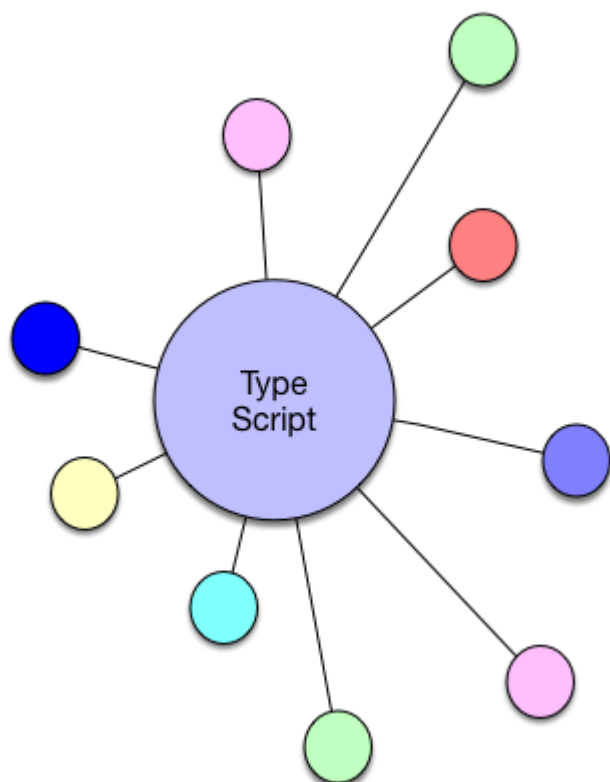


TypeScript es uno de los lenguajes de programación con mayor futuro. ¿Qué es TypeScript?. Basicamente es un superset de JavaScript que aborda de forma elegante muchas de las carencias que JavaScript tiene.



Typescript

Voy a aprovechar a comentar algunas de las cosas que más me gustan de este lenguaje.

1. Apoyado por fabricantes: Yo creo que esto es de lo más importante tanto Microsoft como Google apoyan de una forma fuerte este lenguaje.
2. Compilado: Es una de sus grandes ventajas al ser un lenguaje compilado/transpilado podemos detectar muchos de los errores antes de abordar la ejecución.
3. Visual Studio Code: Hay que reconocer que Microsoft ha hecho un gran trabajo con Visual

Studio Code .

4. Programación orientada a objeto: El soporte para el manejo de clases , interfaces , herencia, composición etc es fuerte y sólido algo fundamental en cualquier lenguaje moderno.
5. Moderno: Se trata de un lenguaje moderno que permite opciones interesantes como por ejemplo union types que permite que un tipo soporte dos opciones .

```
let mivariable : string | number
```

6. Decoradores/Anotaciones: El uso de decoradores o anotaciones (meta información) como nosotros lo llamamos en el mundo Java es clave para conseguir diseñar frameworks potentes.

```
//metadatos
@logClass
class Person {

    public name: string;
    public surname: string;
    constructor(name : string, surname : string) {
        this.name = name;
        this.surname = surname;
    }
}
```

7. Soporte de Genéricos: Los genéricos pueden parecer algo de poca importante pero mi experiencia es que ayudan a realizar refactorings importantes en nuestro código reduciendolo y permitiendo un diseño más solido de librerías y frameworks

```
class GenericClass <T>{
    items :T[] = [];
    pushData(val: T){
```

```
this.items.push(val);  
}  
  
getData(index: number):T{  
    return this.items[index];  
}  
}
```

8. Angular e Ionic : El hecho de que fabricantes estén apostando por diseñar frameworks basados en el lenguaje es otra de las garantías de futuro [Angular](#) e [Ionic](#) son dos de los más destacados pero cada día aparecen más como [type/orm](#).
9. Node.js y el lado Servidor: Al existir hoy por hoy una plataforma de JavaScript sólida como Node.js nos permitirá desarrollar aplicaciones también en el lado servidor abriendo puertas a muchas opciones.
10. Flexibilidad: JavaScript como lenguaje tiene una gran ubicuidad , quizás sea el más ubicuo de todos y habrá en muchas situaciones que queramos usarlo de forma prácticamente directa desde TypeScript . Por ejemplo en TypeScript una variable se define con un tipo concreto , algo que en Javascript no existe . Sin embargo podemos usar el tipo any y acercarnos a JavaScript clásico :

```
let mivariable : Person
```

```
let mivariable : any
```

Tenemos que estar muy atentos a este lenguaje ya que es uno de los que mayor futuro tiene.

Otros artículos relacionados:

1. [Angular async pipe y observables](#)
2. [Angular 5 Hello World y su funcionamiento](#)
3. [Angular 2 y el futuro de las arquitecturas web](#)