blumbit

Typescript - Angular

Introducción a: Angular con Typescript









¿Qué es TypeScript?



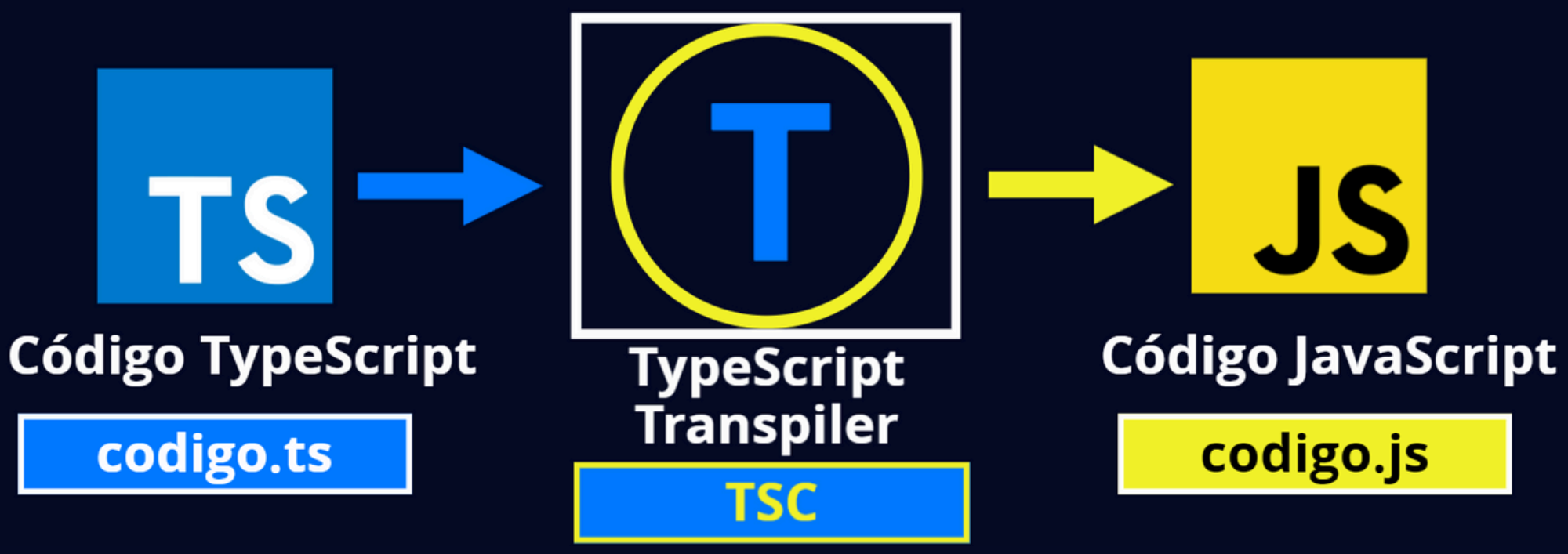
TypeScript es un lenguaje de programación Tipado de código abierto, basada en:



INTRODUCCIÓN A : ANGULAR CON TS TypeScript

¿Qué es TypeScript?

Es un superconjunto escrito de JavaScript y contiene todos sus elementos y se **compila** en JavaScript simple



Convertir el código de TypeScript (**codigo.ts**) a JavaScript (**codigo.js**)



Porqué usar TypeScript

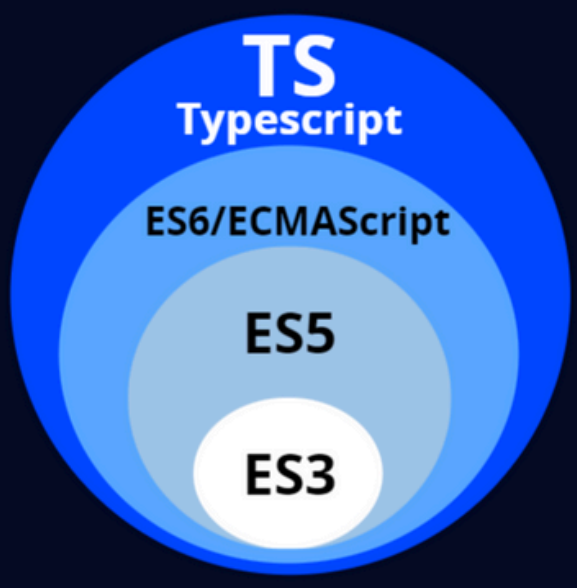
Potencia tu código JavaScript **(JavaScript con Super Poderes)**

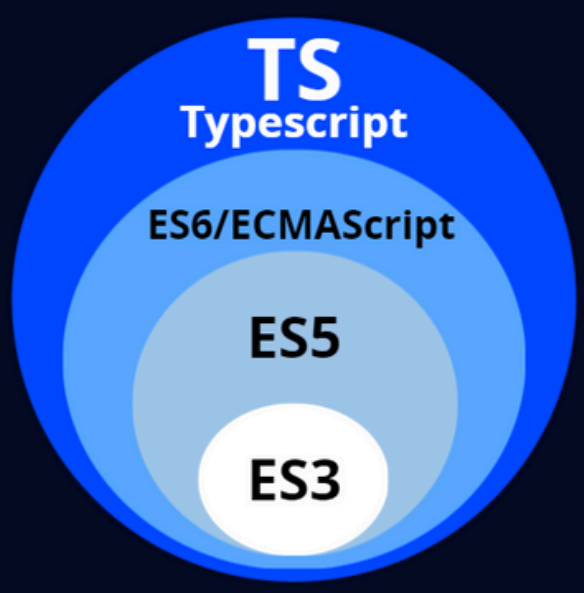
- Lenguaje de Programación Tipado
- Superconjunto de JavaScript
- Código Abierto
- Programación Orientada a Objetos
- Puede usar TypeScript para Backend
- Todos los beneficios de ES6 (ECMAScript 6), para una mayor productividad
- Lenguaje Interpretado **Compilador que convierte código TypeScript a código JavaScript**
- Comprobación de errores en tiempo de compilación

Tics

blumbit

Superconjunto del Lenguaje de Programación JavaScript





Javascript



Números

Puede usar TypeScript para desarrollo del lado del Servidor **(backend)**



Empresas que usan TypeScript

Microsoft Google Medium

airbnb slack shopify

2958+ compañias en: https://stackshare.io/typescript

Herramientas Necesarias

Descargar Node.js

https://nodejs.org/en/download/

Descargar la versión LTS Recomendado para la mayoría de los usuarios

Editor de codigo

Visual Studio Code o

Recomendado Visual Studio Code

Verificar la instalación de Node.js y NPM

```
> node --vesion # o node -v
v12.18.3

>npm --version # o npm -v
6.14.6
```

Tambien se puede utilizar NVM. lo podemos descargar en:
https://docs.microsoft.com/en-us/windows/nodejs/setup-on-windows
https://github.com/coreybutler/nvm-windows/releases

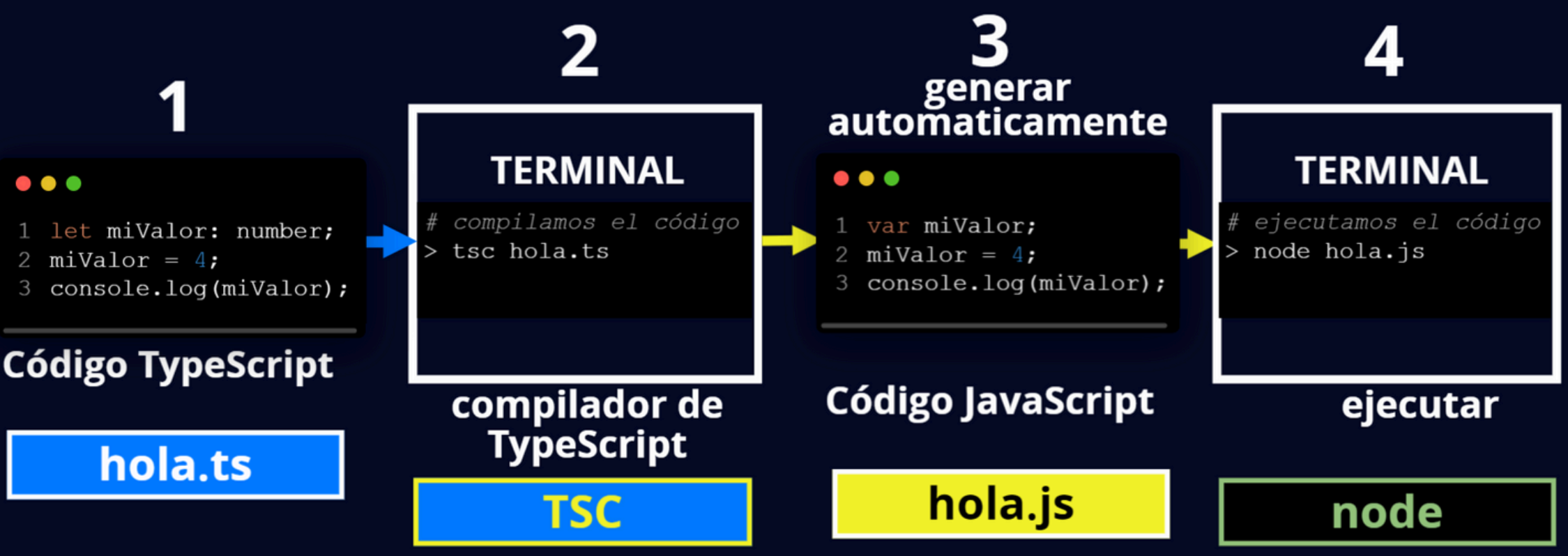
Instalación de typeScript

tsc: TypeScript Compiler

```
# 1. Podemos instalar TypeScript a través de npm
> npm install -g typescript

# 2. Verificamos la version del compilador
> tsc --version # o tsc -v
Version 4.0.3
```

Cómo utilizar el compilador tsc



compilador tsc con la opción --watch



Typescript - Angular

Introducción a: Angular con Typescript

Generación y Configuración del archivo tsconfig.json

Generamos en archivo tsconfig.json

```
# Generamos el archivo tsconfig.json
> tsc --init
```

Estructura Proyecto



Proyecto en TypeScript

habilitamos opciones de salida para nuestros archivos js en archivo tsconfig.json

```
{
  "compilerOptions": {
        ...
        "outDir": "./dist",  /* Redirect output structure to the directory. */
        ...
  }
}
```

Estructura Proyecto



Tipos de Datos en TypeScript

Tipo de dato Explícito

```
nombreVariable : TipoDeDato
```

El " : " permite asignar el tipo de dato

Tipo de dato Inferido

```
nombreVariable = Valor
```

TypeScript deduce el tipo de dato

valor inicial

```ts
// Asignación del tipo de dato manual (Explicito)
const pi: Number = 3.1416;

// Asignación del tipo de dato automatico (Inferido)
const pi = 3.1416;
```

Tipos de datos Primitivos en Typescript

```
edad : Number
```

El " : " permite asignar el tipo de dato

- Number
- Tuple
- Null
- String
- Enum
- Undefined
- Boolean
- Any
- Never
- Array
- Void
- Void

Funciones en TypeScript

```ts
// Funcion simple
function saludo() {
    console.log("Hola Mundo");
}

saludo();

// Funciones con parametros
function saludoRetorno(valor: string) {
    console.log("Hola, "+valor);
}
saludoRetorno("Cristian");

//Funciones con retorno
function suma(x: number, y:number): number {
    var s = x+y
    return s;
}
console.log(suma(6, 8));
```

Nombre de la Función

parámetros de entrada

tipo de dato

tipo de retorno de la función

lógica, proceso

Funciones en TypeScript

```ts
type Categorias = "ropa"|"muebles"|"electrodomesticos";

function guardarProducto(titulo: string, cantidad:number, precio?:number,categoria?: Categorias){
    console.log("Guardando Producto", titulo, cantidad, precio, categoria);
}

guardarProducto("Mesa",12);
guardarProducto("Silla", 20, 250.89, "muebles");
```

tipo de retorno de funcion: (void)

```ts
type Categorias = "ropa"|"muebles"|"electrodomesticos";

function guardarProducto2(titulo: string, cantidad:number, precio?:number,categoria?: Categorias): string{
    return `Guardando Producto ${titulo}, ${cantidad}, ${precio}, ${categoria}`;
}

console.log(guardarProducto2("Mesa",12));
console.log(guardarProducto2("Silla", 20, 250.89, "muebles"));
```

tipo de retorno de funcion: (string)

Interfaces

Una interface declara una serie de métodos y propiedades que deben ser implementados luego por una o más clases.

Las interfaces vienen a suplir la imposibilidad de herencia múltiple.

```ts
export interface Persona {
    nombre: string;
    apellidos: string;
    edad: number;
    otro?: any;
}
```

Clases

```ts
import { Persona }from './persona'

class PersonaComponent{
    nombre: string;
    apellidos: string;
    edad: number;
    otro?: any;

    constructor(datos:Persona){
        this.nombre = datos.nombre
        this.apellidos = datos.apellidos
        this.edad = datos.edad
        this.otro = datos.otro
    }

    mostrar(){
        return `nombre: ${this.nombre}, apellidos: ${this.apellidos}, edad: ${this.edad}`;
    }
}

let p1 = new PersonaComponent({nombre: "Juan", apellidos: "Perez", edad: 30})
console.log(p1.mostrar());
```

Una Clase es como una estructura predefinida que sirve como molde para crear Objetos.

En este (molde) se declaran atributos y métodos; un atributo puede ser una variable, y un método puede ser una función, si lo comparamos con Javascript.

blumbit

Typescript - Angular

Introducción a: Angular con Typescript



Tics

Instalación Angular

```
> npm install -g @angular/cli

> ng new blumbit

> cd blumbit

> ng serve -o
```



Angular CLI

herramienta de línea de comandos estándar para crear, depurar y publicar aplicaciones **Angular**

ng new

La CLI Angular facilita la creación de una aplicación que ya funciona, desde el primer momento.

ng serve

Pruebe fácilmente su aplicación localmente mientras se desarrolla.

ng generate

Genere:

componentes, rutas, servicios y pipes

con un simple comando.

La CLI también creará shells de prueba simples para todos estos.