**Typescript**

**José Santorcuato Tapia**

**2018**

***TypeScript****es un lenguaje de programación libre y de*[*código abierto*](https://es.wikipedia.org/wiki/C%C3%B3digo_abierto)*desarrollado y mantenido por*[*Microsoft*](https://es.wikipedia.org/wiki/Microsoft)*. Es un superconjunto de*[*JavaScript*](https://es.wikipedia.org/wiki/JavaScript)*, que esencialmente añade tipado estático y objetos basados en clases.*[*Anders Hejlsberg*](https://es.wikipedia.org/wiki/Anders_Hejlsberg)*, diseñador de*[*C#*](https://es.wikipedia.org/wiki/C)*y creador de*[*Delphi*](https://es.wikipedia.org/wiki/Delphi)*y*[*Turbo Pascal*](https://es.wikipedia.org/wiki/Turbo_Pascal)*, ha trabajado en el desarrollo de TypeScript.*[*1*](https://es.wikipedia.org/wiki/TypeScript#cite_note-1)*​ Typescript puede ser usado para desarrollar aplicaciones JavaScript que se ejecutarán en el lado del cliente o del servidor (*[*Node.js*](https://es.wikipedia.org/wiki/Node.js)*).*[Wikipedia](https://es.wikipedia.org/wiki/TypeScript)

**Instalación typescript**

En la sesión anterior instalamos @angular/cli, esto instalará automáticamente typescript

tsc –v

Evita instalarla por separado, destruye angular

**Algunas convenciones**

Las variables se declaran let para locales y var para globales

Ej:

**let** nombreSoft: string = `Angular`;

**let** saludo: string = `Hola, este es el curso de ${ nombreSoft }.

**Tipos de variables**

Any

number

number[]

Array<number>

string

enum

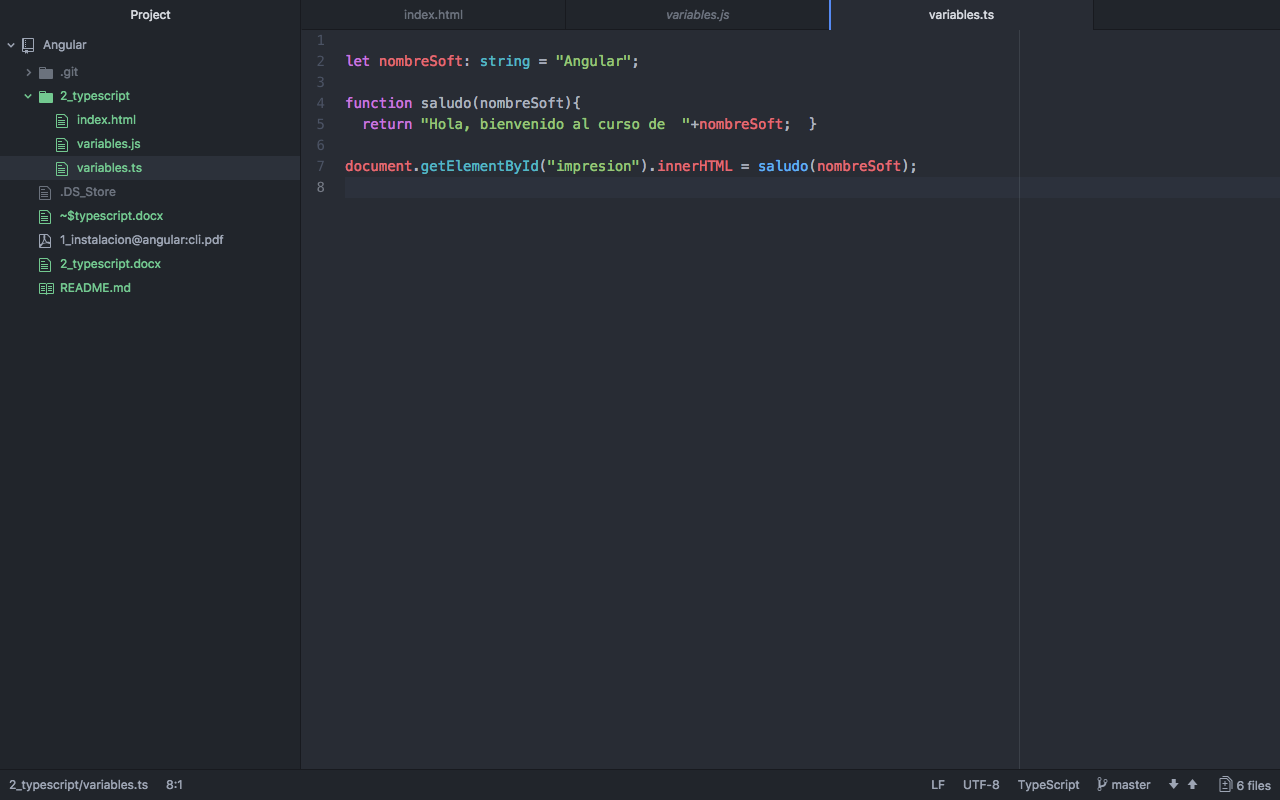
boolean

**Creación de proyecto typescript, estructura**

index.html

variables.ts

variables.js(se creará al ejecutar el .ts)



**index.html**

<!DOCTYPE html> <html lang="es">

<head>

<title></title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1"

</head>

<body>

<h1 id="impresion"></h1>

<script type="text/javascript" src="variables.js"></script>

</body> </html

**variables.ts**

let nombreSoft: string = "Angular";

function saludo(nombreSoft){

return "Hola, bienvenido al curso de "+nombreSoft; }

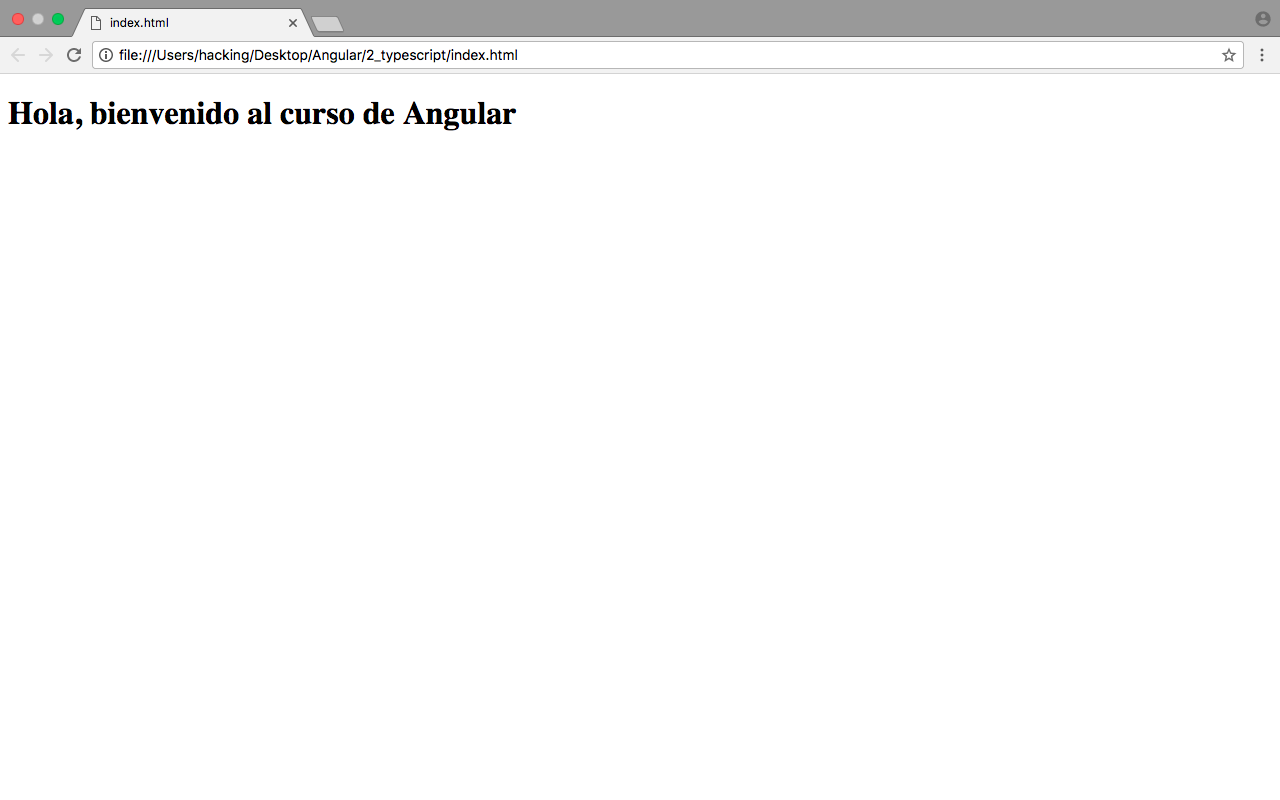
document.getElementById("impresion").innerHTML = saludo(nombreSoft);

Ejecutar en terminal

tsc variables.ts

Creará el .js

Abrir el index.htm



**Clases**

Las clases en Typescript tienen la siguiente estructura

**class Estudiante** **{**

//Variables

public nombre: string;

public edad: number;

//Metodos

public setNombre(nombre: string) {

this.nombre = nombre;

}

public getNombre() {

return this.nombre;

}

public setEdad(edad: number) {

this.edad = edad;

}

public getEdad() {

return this.edad;

}

**}**

var nuevoEstudiante = new Estudiante();

nuevoEstudiante.setNombre('Jose');

nuevoEstudiante.setEdad(42);

console.log(nuevoEstudiante.getNombre());

console.log(nuevoEstudiante.getEdad());

**Ahora con una pequeña variante, intenta que se pasen los datos a html**

document.getElementById("nombre").innerHTML = nuevoEstudiante.getNombre();

document.getElementById("edad").innerHTML = nuevoEstudiante.getEdad();

**Solución de errores**

Type 'number' is not assignable to type 'string'

document.getElementById("nombre").innerHTML = nuevoEstudiante.getNombre();

document.getElementById("edad").innerHTML = nuevoEstudiante.getEdad().toFixed(0);

**Constructor**

El constructor permite asignar valores a variables, a diferencia del ejemplo anterior

nuevoEstudiante.setNombre('Jose') ;

Con constructor queda

constructor(){

this.nombre = "Jose";

}

También permite parámetros

constructor(nombre){

this.nombre = "Jose";

}

**Interfaces**

**Las interfaces obligan a las clases a tener miembros, cumplir con miembros básicos, si implementa una interface y no cumple con los miembros arroja error**

**Ej:**

**interface DatosEstudiantes {**

**nombre: string;**

**direccion :string;**

**}**

**class Estudiante implements DatosEstudiantes {**

**//Variables**

**public nombre: string;**

**public edad: number;**

**constructor(){**

**this.nombre = 'Jose';**

**this.edad = 42;**

**}**

**//Metodos**

**public setNombre(nombre: string) {**

**this.nombre = nombre;**

**}**

**public getNombre() {**

**return this.nombre;**

**}**

**public setEdad(edad: number) {**

**this.edad = edad;**

**}**

**public getEdad() {**

**return this.edad;**

**}**

**}**

**var nuevoEstudiante = new Estudiante();**

**console.log(nuevoEstudiante.getNombre());**

**console.log(nuevoEstudiante.getEdad());**

**document.getElementById("nombre").innerHTML = nuevoEstudiante.getNombre();**

**document.getElementById("edad").innerHTML = nuevoEstudiante.getEdad().toFixed(0);**

**El constructor no tiene el miembro dirección…**

Class 'Estudiante' incorrectly implements interface 'DatosEstudiantes'.

  Property 'direccion' is missing in type 'Estudiante'