TypeScript

Conocimientos Previos

Web Application Development

ADVERTENCIA

- Este no es un curso desde cero para aprender TypeScript ni JavaScript
- Solo se analizarán detalles del lenguaje que nos serán de utilidad para trabajar con el framework Angular.

¿Por qué TypeScript?

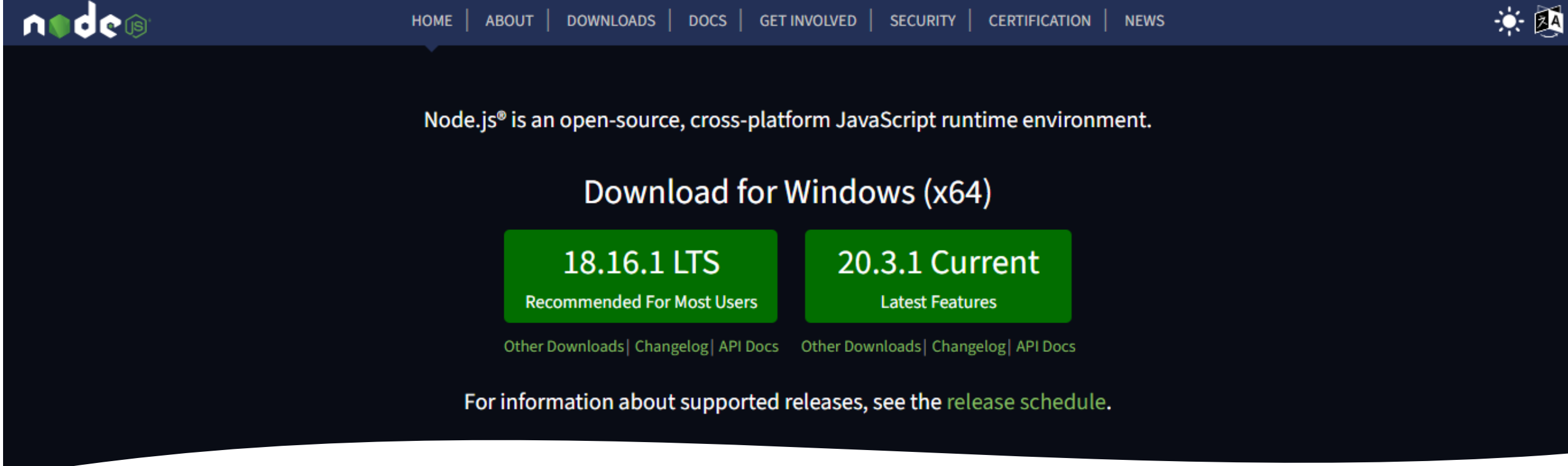
- Está basado en JavaScript.
- Antes de JS ES2015 era muy complicado crear aplicaciones complejas y a gran escala.
- Microsoft vio la necesidad de “robustecer” JavaScript.
- En 2012 es liberado al público TypeScript.
- Este lenguaje incluye muchas mejoras de lenguajes modernos que en ese entonces JavaScript no implementaba.

¿Por qué aprenderlo?

- Es fácil.
- Es completamente compatible con código JavaScript.
- Varios frameworks lo están implementando ya en sus códigos y lo soportan (Angular, React, Vue, entre otros).
- Se puede usar para crear grandes proyectos, en producción, manipular datos, entre muchas otras cosas.

TypeScript es Tipado Estático

- Una de sus ventajas es su tipado Estático
- Esto nos permite utilizar:
  - Interfaces
  - Conversiones de Datos
  - Genéricos
  - Detección de errores en tiempo de compilación y codificación (mediante editores de texto con detección de errores y autocompletado).



Instalación

Hay varias maneras de trabajar con TypeScript.

La manera más común es mediante el entorno de desarrollo “NodeJS”.

Se recomienda instalar la versión LTS (Long-Term Support).

Instalar TypeScript

Después de Instalar NodeJS debemos ejecutar el siguiente comando en la terminal del sistema para instalar TypeScript.

“npm install –g typescript”

- **npm** es el comando para llamar al manejador de paquetes de NodeJS.
- **Install -g** indica que se va a instalar el paquete de manera global.
- **Typescript** es el nombre del paquete a instalar.

El “Hola Mundo” en TypeScript

- Se recomienda usar Visual Studio Code ya que tiene soporte nativo para código TypeScript.
- Haremos la plantilla básica de un “index.html”

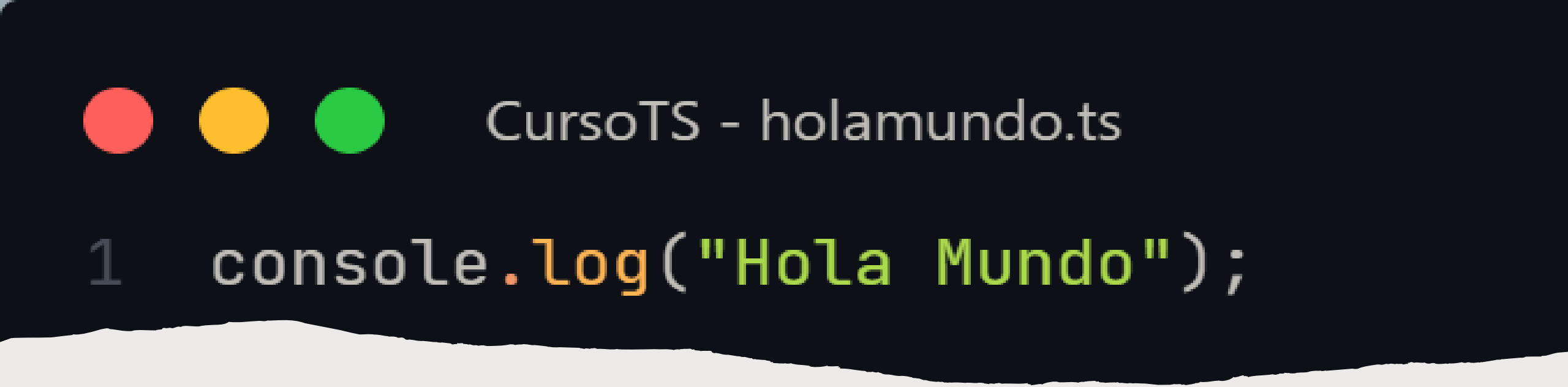
CursoTS - index.html

```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Hola Mundo</title>
</head>

<body>
    <h1>Hola Mundo</h1>
</body>
<script src="holamundo.js"></script>

</html>
```



- Posteriormente crearemos un archivo llamado “holamundo.ts”.
- “.ts” es la extensión de los archivos TypeScript.
- compilaremos este archivo TypeScript a un archivo JavaScript para que pueda ser utilizado en el navegador.

“tsc <nombre del archivo>.ts”

Por ejemplo:

“tsc holamundo.ts”

Este comando generara un archivo con el mismo nombre, pero “convertido” a un archivo JavaScript.

```
PS C:\Users\      \Desktop\CursoTS> tsc .\holamundo.ts
[10:19:19 PM] Starting compilation in watch mode...
[10:19:43 PM] Starting compilation in watch mode...

[10:19:45 PM] Found 0 errors. Watching for file changes.
```

- Para no tener que estar compilando cada vez que hacemos un cambio en un archivo vamos a implementar un observador (un watcher).
- Este observador estará al tanto de todos los cambios que se hacen en el archivo .ts y automáticamente hará la compilación a JavaScript.
- Para implementarlo agregaremos la bandera “—watch”

“tsc <nombre del archivo>.ts --watch”

Tipos de Dato, Variables y Constantes

CursoTS - holamundo.ts

```ts
// Numeros
// var <nombre de la Variable> : <tipo de Dato>;
var entero: number, decimal: number;
entero = 52;
decimal = 3.1415;

// TS identifica el tipo de dato al inicializarlo.
var numero = 20;

// Se pueden hacer todas los operaciones basicas
console.log(entero + decimal);
console.log(entero - decimal);
console.log(entero * decimal);
console.log(entero / decimal);
console.log(entero % decimal);
```

Tipos de Dato, Variables y Constantes

CursoTS - holamundo.ts

```ts
// Strings
var texto: string;
var nombre = "Soy fulano";
texto = "Hola Mundo";

// Se puede concatenar Strings
console.log(texto + " " + nombre);

```

Tipos de Dato, Variables y Constantes

CursoTS - holamundo.ts

```ts
// Booleans
var esBool: boolean = true;
var esFalso = false;
var numCompare = (123 + 4 == 45);
var numDif = (123 + 4 != 45);

// Algunas Operaciones Logicas
esBool && esFalso;
esBool || esFalso;
```

Tipos de Dato, Variables y Constantes

CursoTS - holamundo.ts

```
// Objetos
// Los objetos son unicos
// Si queremos crear varios iguales es necesario un I
nterface
var persona = {
    nombre: "Tomas",
    apellido: "Tercero",
    edad: 28,
    peso: 90
}

console.log(persona.nombre + " " + persona.apellido);
console.log("edad: ", persona.edad);
console.log("peso: ", persona.peso);
```

Tipos de Dato, Variables y Constantes

CursoTS - holamundo.ts

```ts
// Interface
interface Persona {
    nombre: string,
    apellido: string,
    edad: number,
    peso: number
}

var persona: Persona;

persona = {
    nombre: "Carlos",
    apellido: "Cuarto",
    edad: 16,
    peso: 70
}

var persona2: Persona = {
    nombre: "Carlos",
    apellido: "V",
    edad: 16,
    peso: 50
}

console.log(persona.nombre + " " + persona.apellido);
```

Tipos de Dato, Variables y Constantes

CursoTS - holamundo.ts

```ts
// Tipo Any

/*
 El tipo any es cuando no sabes el tipo
    de dato a trabajar.

 Es muy util cuando consumes un API y
    no sabes el tipo de dato que retorna.

 No se aconseja abusar de este tipo ya que
    perdemos todas las ventajas de TS
*/
var algo: any;

algo = 45;
algo = "soy un Texto";
algo = true;
algo = NaN;
algo = undefined;
```

Tipos de Dato, Variables y Constantes

CursoTS - holamundo.ts

```
1  // Arreglos
2
3  var listaNombres: Array<string> = ["Juana","Maria","Luca","Pedro"];
4  console.log(listaNombres[1]);
5
6  var listaNumeros: Array<number> = [0];
7  listaNumeros.push(300);
8
9  interface Alumno {
10     nombre: string,
11     edad: number
12 }
13
14 var listaAlumnos: Array<Alumno> = [];
15
16 listaAlumnos.push(
17     {
18         nombre: "Julia",
19         edad: 23
20     }
21 )
```

Funciones

CursoTS - holamundo.ts

```ts
// Funciones Simples
function mostrarHola() {
    console.log("Hola");
}

// Funciones con Parametros
function sumarDosNumeros(num1: number, num2: number) {
    console.log(num1 + num2);
}

// Funciones con Retorno
function restarDosNumeros(num1: number, num2: number): number {
    return num1 - num2;
}

/*
Por defecto, si no indicamos el tipo de retorno se
asigna el tipo "void".
*/
```

Clases

CursoTS - holamundo.ts

```ts
// Clases
// Los nombres de las clases son con PascalCase
class Alumno {
    // Por defecto los atributos y metodos son Publicos

    // Atributos
    public nombre: string;
    protected apellido: string;
    private edad: number;

    // Metodos
    constructor() {
        this.nombre = "";
        this.apellido = "";
        this.edad = 0;
    }

    private mostrarNombre() : void {
        console.log("Bienvenido(a) " + this.nombre + " " +this.apellido);
        console.log("Edad: "+ this.edad);

    }

    asignarValores(nombre: string, apellido: string, edad: number) {
        this.nombre = nombre;
        this.apellido = apellido;
        this.edad = edad;
        this.mostrarNombre();
    }
}
// Fin de la Clase

var alumno: Alumno = new Alumno();
alumno.asignarValores("Pepe","Piedra",300);
```