

Introducción a TypeScript

TypeScript

- Nuevo lenguaje
- Sintaxis basada en JavaScript
- Orientado a Objetos
- Tipado
- Se compila en JavaScript básico

TypeScript

- **Vista inicial**
- Partes de un fichero TS
- Visibilidad
- Objetos y Arrays

Vista inicial

Los componentes son
clases exportadas.

```
TS correo.component.ts x
1  import { Component, OnInit } from '@angular/core';
2
3  @Component({
4    selector: 'app-correo',
5    templateUrl: './correo.component.html',
6    styleUrls: ['./correo.component.scss']
7  })
8  export class CorreoComponent implements OnInit {
9
10     title: string;
11     public body: string;
12     private isStarted = false;
13
14     constructor() {
15         this.title = 'Correo';
16         this.body = 'Cuerpo del correo';
17     }
18
19     ngOnInit() {
20         this.isStarted = true;
21     }
22
23     private clickBoton() {
24         console.log("Click en botón");
25     }
26
27 }
28
```

TypeScript

- Vista inicial
- **Partes de un fichero TS**
- Visibilidad
- Objetos y Arrays

Import

Incluimos los
ficheros externos

```
TS correo.component.ts x
1  import { Component, OnInit } from '@angular/core';
2
3  @Component({
4    selector: 'app-correo',
5    templateUrl: './correo.component.html',
6    styleUrls: ['./correo.component.scss']
7  })
8  export class CorreoComponent implements OnInit {
9
10     title: string;
11     public body: string;
12     private isStarted = false;
13
14     constructor() {
15         this.title = 'Correo';
16         this.body = 'Cuerpo del correo';
17     }
18
19     ngOnInit() {
20         this.isStarted = true;
21     }
22
23     private clickBoton() {
24         console.log("Click en botón");
25     }
26
27 }
28
```

Cabecera

Nombre de la clase,
interfaces que debe
implementar o clases de
las que hereda.

```
TS correo.component.ts x
1  import { Component, OnInit } from '@angular/core';
2
3  @Component({
4    selector: 'app-correo',
5    templateUrl: './correo.component.html',
6    styleUrls: ['./correo.component.scss']
7  })
8  export class CorreoComponent implements OnInit {
9
10     title: string;
11     public body: string;
12     private isStarted = false;
13
14     constructor() {
15         this.title = 'Correo';
16         this.body = 'Cuerpo del correo';
17     }
18
19     ngOnInit() {
20         this.isStarted = true;
21     }
22
23     private clickBoton() {
24         console.log("Click en botón");
25     }
26
27 }
28
```

Variables

Deben declararse junto a su tipo al inicio de la clase

```
TS correo.component.ts x
1  import { Component, OnInit } from '@angular/core';
2
3  @Component({
4    selector: 'app-correo',
5    templateUrl: './correo.component.html',
6    styleUrls: ['./correo.component.scss']
7  })
8  export class CorreoComponent implements OnInit {
9
10     title: string;
11     public body: string;
12     private isStarted = false;
13
14     constructor() {
15         this.title = 'Correo';
16         this.body = 'Cuerpo del correo';
17     }
18
19     ngOnInit() {
20         this.isStarted = true;
21     }
22
23     private clickBoton() {
24         console.log("Click en botón");
25     }
26
27 }
28
```


Constructor

- Primera función en ejecutarse.
- Provee de los servicios necesarios al componente

```
TS correo.component.ts x
1  import { Component, OnInit } from '@angular/core';
2
3  @Component({
4    selector: 'app-correo',
5    templateUrl: './correo.component.html',
6    styleUrls: ['./correo.component.scss']
7  })
8  export class CorreoComponent implements OnInit {
9
10     title: string;
11     public body: string;
12     private isStarted = false;
13
14     constructor() {
15         this.title = 'Correo';
16         this.body = 'Cuerpo del correo';
17     }
18
19     ngOnInit() {
20         this.isStarted = true;
21     }
22
23     private clickBoton() {
24         console.log("Click en botón");
25     }
26
27 }
28
```

OnInit

- Función de ciclo de vida de Angular.
- Se ejecuta al iniciarse el componente en la vista

```
TS correo.component.ts x
1  import { Component, OnInit } from '@angular/core';
2
3  @Component({
4    selector: 'app-correo',
5    templateUrl: './correo.component.html',
6    styleUrls: ['./correo.component.scss']
7  })
8  export class CorreoComponent implements OnInit {
9
10     title: string;
11     public body: string;
12     private isStarted = false;
13
14     constructor() {
15         this.title = 'Correo';
16         this.body = 'Cuerpo del correo';
17     }
18
19     ngOnInit() {
20         this.isStarted = true;
21     }
22
23     private clickBoton() {
24         console.log("Click en botón");
25     }
26
27 }
28
```

Función

- Función simple
- Puede ser asociada a un evento en un template.

```
TS correo.component.ts x
1  import { Component, OnInit } from '@angular/core';
2
3  @Component({
4    selector: 'app-correo',
5    templateUrl: './correo.component.html',
6    styleUrls: ['./correo.component.scss']
7  })
8  export class CorreoComponent implements OnInit {
9
10     title: string;
11     public body: string;
12     private isStarted = false;
13
14     constructor() {
15         this.title = 'Correo';
16         this.body = 'Cuerpo del correo';
17     }
18
19     ngOnInit() {
20         this.isStarted = true;
21     }
22
23     private clickBoton() {
24         console.log("Click en botón");
25     }
26
27 }
28
```

TypeScript

- Vista inicial
- Partes de un fichero TS
- **Visibilidad**
- Objetos y Arrays

Público y privado

- Identificadores de visibilidad
- Aplicables a variables y funciones
- Privatizar evita que sea visible desde fuera al usar la clase

TypeScript

- Vista inicial
- Partes de un fichero TS
- Visibilidad
- **Objetos y Arrays**

Objetos {}

- Las clases, al ser exportadas a otras, se transforman en objetos JavaScript
- Mediante interfaces o clases podemos crear nuestros tipos de datos para usar en la aplicación
- Modelar nuestros objetos tiene muchas ventajas

Arrays []

- Tipos de datos iterables
- Contienen listas de objetos
- La base para cualquier transmisión de datos

¡Gracias!