# Typescript

* Typescript es un superset de Javascript, por lo tanto está escrito en Javascript.
* Es un agregado a JavaScript, no lo reemplaza. Algunas tareas las tendremos que hacer en Javascript.
* Agrega nuevas características:
  + Tipos
  + Clases
  + Interfaces
* Es transpilado a Javascript.
* Es el lenguaje con el que se programa para Angular4.

# Herramientas necesarias para utilizar Angular4

* Node.js
* Angular CLI

# Angular CLI

Es un set de herramientas de consola que sirve para crear proyectos y generar componentes, rutas, servicios, pipes, etc.

Permite correr nuestra app de forma local, hacer testing y preparar una app para producción.

<https://cli.angular.io/>

Para instalar Angular CLI se corre el siguiente comando:

**npm install -g @angular/cli**

# Crear un nuevo proyecto

**ng new <nombre de proyecto>**

Este comando crea la carpeta del proyecto y todas las subcarpetas y archivos necesarios.

# Correr proyecto

**ng serve**

Este comando (dentro de la carpeta de nuestro proyecto) transpila el código y corre un servidor en local.

# Módulos

Es la forma como se debe dividir un proyecto. Se declaran en app.module.ts, en la sección **imports** dentro de **@NgModule**.

# Componentes

Cada vista de la app deberá ser un componente distinto. Se declaran en app.module.ts, en la sección **declarations** dentro de **@NgModule**.

Cada componente deberá tener un **component.ts** en donde indica la información de nuestro componente. Esto se hace dentro del decorador **@Component**. Deberá tener obligatorio un **selector** y un **templateUrl,** opcional podemos tener un archivo de estilos en **styleUrls**. Para que se pueda utilizar ese componente se deberá exportar. Cuando se crea el proyecto con **ng new <nombre de proyecto>** se tiene por default un componente llamado app.component.ts.

# Data Binding

Comunicación entre el código que escribimos y lo que ve finalmente el cliente.

## Tipos

### String Interpolation {{}}

Tener TypeScript y presentar al usuario HTML.

**Ejemplo:**

**app.component.ts**

import { Component } from '@angular/core';

@Component({

selector: 'app-root',

templateUrl: './app.component.html',

styleUrls: ['./app.component.css']

})

export class AppComponent {

title = 'JazzoSquare';

a = 3;

b = 5;

}

**app.component.html**

<div style="text-align:center">

<h1>

Welcome to {{title}}!

</h1>

<h2>

{{a+b}}

</h2>

</div>

### Property Binding []

Tener información de HTML y que viaje a TypeScript.

**Ejemplo:**

**app.component.ts**

import { Component } from '@angular/core';

@Component({

selector: 'app-root',

templateUrl: './app.component.html',

styleUrls: ['./app.component.css']

})

export class AppComponent {

title = 'JazzoSquare';

listo = false;

constructor() {

setTimeout(()=>{this.listo = true;}, 3000);

}

}

**app.component.html**

<div style="text-align:center">

<h1>

Welcome to {{title}}!

</h1>

<hr/>

<button [disabled]="!listo">Click me!</button>

</div>

### Event Binding []

Tener evento en HTML y que viaje a TypeScript.

**Ejemplo:**

**app.component.ts**

import { Component } from '@angular/core';

@Component({

selector: 'app-root',

templateUrl: './app.component.html',

styleUrls: ['./app.component.css']

})

export class AppComponent {

title = 'JazzoSquare';

hacerAlgo() {

alert('Haciendo algo');

}

}

**app.component.html**

<div style="text-align:center">

<h1>

Welcome to {{title}}!

</h1>

<hr/>

<button (click)="hacerAlgo()">Click me!</button>

</div>

### Two Way Data Binding [()]

Comunicación de 2 vías.

**Ejemplo:**

**app.component.ts**

import { Component } from '@angular/core';

@Component({

selector: 'app-root',

templateUrl: './app.component.html',

styleUrls: ['./app.component.css']

})

export class AppComponent {

title = 'JazzoSquare';

nombre:string = '';

}

**app.component.html**

<div style="text-align:center">

<h1>

Welcome to {{title}}!

</h1>

<br/>

<input type="text" placeholder="Tu nombre" [(ngModel)]="nombre"/>

<br/>

Mi nombre es {{nombre}}

</div>

**app.module.ts**

import { BrowserModule } from '@angular/platform-browser';

import { NgModule } from '@angular/core';

import { AppComponent } from './app.component';

import { FormsModule } from '@angular/forms';

@NgModule({

declarations: [

AppComponent

],

imports: [

BrowserModule,

FormsModule

],

providers: [],

bootstrap: [AppComponent]

})

export class AppModule { }