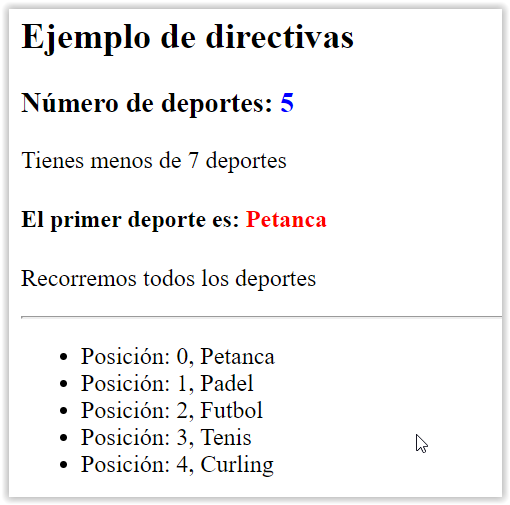
**DIRECTIVAS DE ANGULAR**

* Una directiva nos permite poder evaluar datos dentro del código HTML, como, por ejemplo, un IF.
* También podemos utilizar directivas de bucles.
* Para visualizar las directivas vamos a crear una aplicación dónde dibujaremos los deportes que tendremos en Array.
* Tendremos un Array de string y lo recorreremos dentro del código HTML.
* Todo será estático por ahora.



Creamos un nuevo **component** llamado **Deportes**

**DEPORTES.COMPONENT.TS**

import { Component, OnInit } from "@angular/core";

@Component({

  selector: "app-deportes",

  templateUrl: "./deportes.component.html",

  styleUrls: ["./deportes.component.css"]

})

export class DeportesComponent implements OnInit {

  //DECLARAMOS LOS DEPORTES COMO UN ARRAY

  public deportes: Array<string>;

  constructor() {

    //INICIALIZAMOS LOS DEPORTES QUE DESEAMOS EN LA VISTA

    this.deportes = ["Petanca", "Padel", "Futbol", "Tenis", "Curling"];

  }

  ngOnInit(): void {}

}

Y dibujamos los deportes en la vista.

**DEPORTES.COMPONENT.HTML**

<h2>Ejemplo de directivas</h2>

<h3>

  Número de deportes:

  <span style="color:blue">

    {{ deportes.length }}

  </span>

</h3>

<!-- PODEMOS PREGUNTAR POR EL NUMERO DE DEPORTES CON DIRECTIVA-->

<div \*ngIf="deportes.length <= 7">

  <p>Tienes menos de 7 deportes</p>

</div>

<h4>

  El primer deporte es:

  <span style="color:red">

    {{ deportes[0] }}

  </span>

</h4>

<p>Recorremos todos los deportes</p>

<hr />

<ul>

  <li \*ngFor="let deporte of deportes; let i = index">

    Posición: {{ i }}, {{ deporte }}

  </li>

</ul>

Otra directiva que tenemos es un ELSE dentro de la página HTML. Para realizar else, se utilizan templates, es decir, parte de código que leerá en el caso de que no se cumpla una condición.

Vamos a realizar el else cuando los deportes sean mayores a 7.

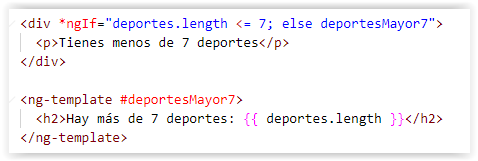
Lo primero será crearnos una plantilla ngTemplate para indicar lo que deseamos visualizar en la página para el else.

<ng-template #deportesMayor7>

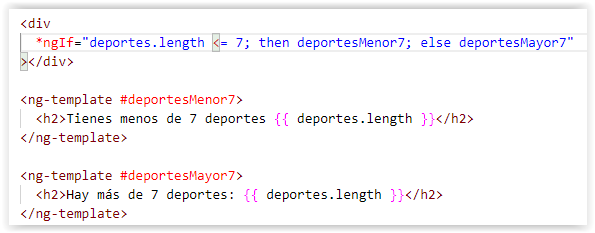
  <h2>Hay más de 7 deportes: {{ deportes.length }}</h2>

</ng-template>

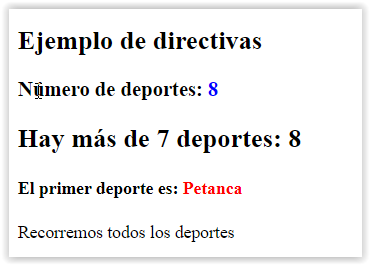
Después, modificamos la directiva ngIf añadiendo un else:



También podemos dividir el código con templates en cada If:



Y veremos el resultado añadiendo más deportes o quitando:



A continuación, vamos a ver otro tipo de directivas, que tienen que ver con el estilo de los objetos HTML.

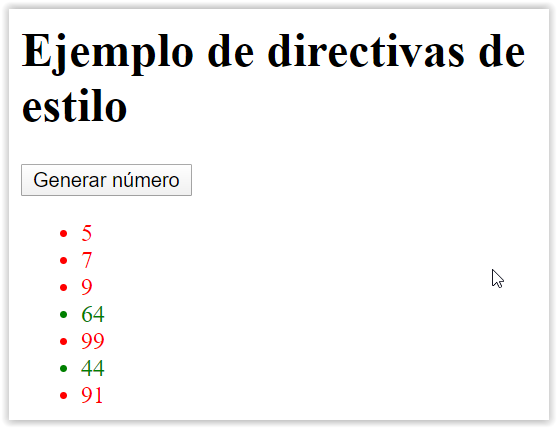
Tendremos un array de números, dónde iremos rellenando números de forma aleatoria con un botón.

Mostraremos los números en una lista, por ejemplo.

Si el número generado es **par**, lo dibujaremos de color verde, si el número es **impar**, lo dibujaremos de color rojo.

Para el ejemplo, utilizaremos una directiva llamada **ngStyle**

Creamos un nuevo componente llamado **NumerosAleatorios**.



**NUMEROALEATORIOS.COMPONENT.TS**

import { Component, OnInit } from "@angular/core";

@Component({

  selector: "app-numeros-aleatorios",

  templateUrl: "./numeros-aleatorios.component.html",

  styleUrls: ["./numeros-aleatorios.component.css"]

})

export class NumerosAleatoriosComponent implements OnInit {

  //TENDREMOS UN ARRAY INICIAL DE NUMEROS

  public numeros: Array<number>;

  constructor() {

    //CREAMOS UNOS NUMEROS INICIALES EN EL

    //ARRAY

    this.numeros = [5, 7, 9];

  }

  //CREAMOS UN METODO PARA GENERAR UN NUMERO

  //ALEATORIO DENTRO DEL ARRAY

  generarNumero() {

    var aleat = Math.floor(Math.random() \* 100) + 1;

    this.numeros.push(aleat);

    console.log(aleat);

  }

  ngOnInit(): void {}

}

**NUMEROALEATORIOS.COMPONENT.HTML**

<h1>Ejemplo de directivas de estilo</h1>

<button (click)="generarNumero()">Generar número</button>

<ul>

  <li

    \*ngFor="let num of numeros"

    [style.color]="num % 2 == 0 ? 'green' : 'red'"

  >

    {{ num }}

  </li>

</ul>

La otra forma que tenemos de trabajar es con estilos de clase css.



Vamos a crearnos un estilo para los números verdes y otro para los rojos dentro del archivo **numerosaleatorios.component.css**

**NUMEROSALEATORIOS.COMPONENT.CSS**

.rojo {

  color: red;

  font-weight: bold;

}

.verde {

  color: green;

  font-weight: bold;

}

Y modificamos el código HTML del componente para indicar la clase css.

**NUMEROALEATORIOS.COMPONENT.HTML**

<h1>Ejemplo de directivas de estilo</h1>

<button (click)="generarNumero()">Generar número</button>

<ul>

  <li

    \*ngFor="let num of numeros"

    [class.verde]="num % 2 == 0"

    [class.rojo]="num % 2 == 1"

  >

    {{ num }}

  </li>

</ul>

También podemos utilizar la sintaxis ngClass, que con un objeto JSON nos permite incluir las clases y las condiciones de forma más limpia.

<h1>Ejemplo de directivas de estilo</h1>

<button (click)="generarNumero()">Generar número</button>

<ul>

  <li

    \*ngFor="let num of numeros"

    [ngClass]="{

      verde: num % 2 == 0,

      rojo: num % 2 == 1

    }"

  >

    {{ num }}

  </li>

</ul>