**SESION DOS**

**TIPO DE DATOS TYPESCRIPT**

Los tipos de datos mas utilizados en typescript son los siguientes:

|  |  |  |
| --- | --- | --- |
| **TIPO DE DATO** | **DESCRIPCION** | **EJEMPLO** |
| let | Recibe cualquier tipo de datos, es utilizado como variable local (dentro del cuerpo de un método). | let miVariable = “semillero”; |
| any | Recibe cualquier tipo de dato, pude ser usado como un objeto con diferentes atributos. | let miVarible : any = {}; |
| number | Recibe cualquier tipo de valor numérico, entero, decimal. | let miVariable : number = 100; |
| string | Recibe cadenas alfanuméricas. | Let miVariable : string = “semillero2019”; |
| Array ó [] | Recibe un arreglo de cualquier tipo de dato. | let miVariable : Array<number> = [1,2,3,4,5] |
| boolean | Recibe datos de tipo lógico, true y false | var miVariable : boolean = true; |
| null | Es utilizado para definir un tipo de dato como nulo. | let miVariable : string = null; |
| void | Tipo de dato vacio, se utiliza para retornar vacio en métodos. | public miMetodo() : void {} |
| undefined | Tipo de dato para definir un tipo de dato con undefined, pero no es nulo. | let miVariable = undefined; |
| var | Tipo de dato utilizado para declara variables globales, recibe cualquier tipo de dato. | var myVariable : Array<any>;  myVariable = new Array<any>;  myVariable.push(1); |
| Map | Es un mapa que contiene una llave y un valor y recibe cualquier tipo de dato. | let miMapa : Map<string,string>;  miMapa = new Map<string,string>();  miMapa.set(“1”, “semillero”); |
| Date | Almacena una fecha | Let mifecha = new Date(); |
| Declarar una clase que se puede instanciar | Es una clase que se puede instanciar y ser utilizada como un tipo de dato. | export class comicDTO {} |

**VARIABLES GLOBALES Y LOCALES**

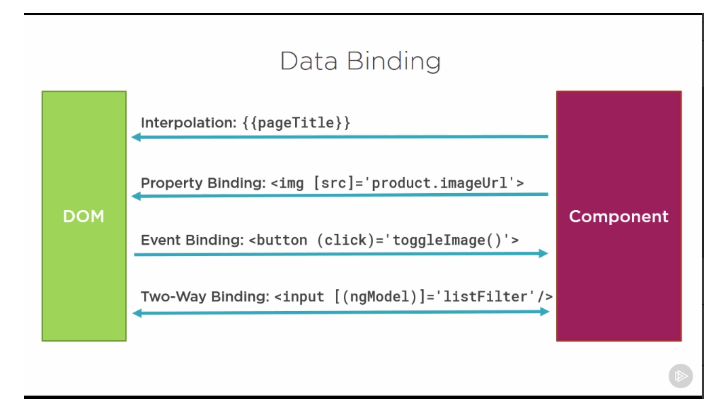
**Variables locales:**

Se declaran dentro del cuerpo de un método y su scope o nivel de alcanza esta solo dentro del método o del bloque en donde se declare.

**Variables globales:**

Se declaran en el cuerpo de clase, y pueden ser accedidas desde cualquier bloque de código implementado dentro de esta, para acceder a la variable se hace uso de la palabra clave (this.variable).

**DATA BINDING E INTERPOLACION**



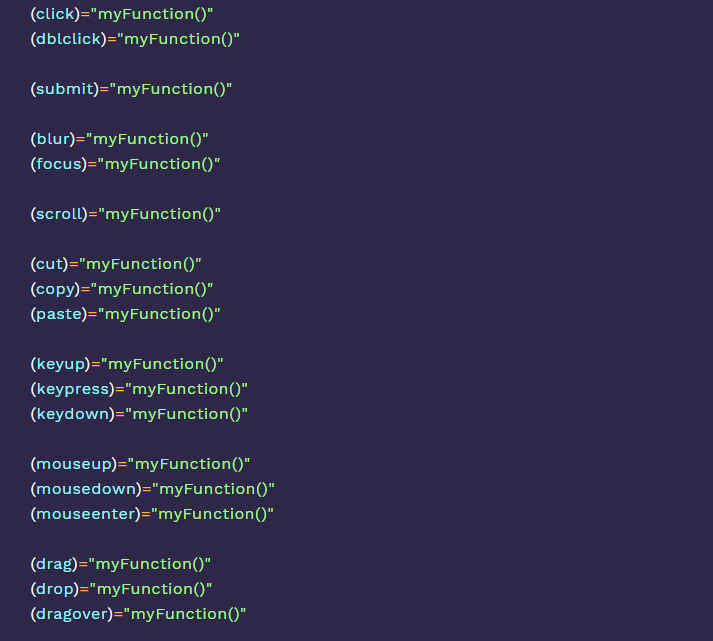
**OPERADORES TYPESCRIPT**

Básicamente son los mismos que se manejan en java, haciendo excepción de algunos que se usan de manera especial o su sintaxis es diferente.

|  |  |  |
| --- | --- | --- |
| **OPERADOR** | **DESCRIPCION** | **EJEMPLO** |
| This | Permite acceder a una variable global o a un metodo declarado. | this.miVariable;  this.miMetodo(); |
| For each | Permite recorrer una lista. | miLista.forEach(objeto =>{  console.log(objeto);  }); |
| Delete | Permite eliminar un atributo de un objeto | let persona = {  id : 1,  nombre: “Diana”,  edad : 21  profesión : “Diseñador grafico”  }  delete persona.profesion; |
| For, while, switch, if else | Operadores básicos. |  |

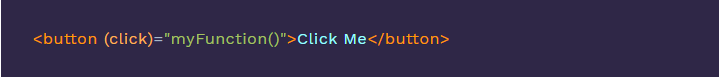
**EVENTOS BINDING**

Los eventos son utilizados vara realizar o ejecutar acciones en el componente.

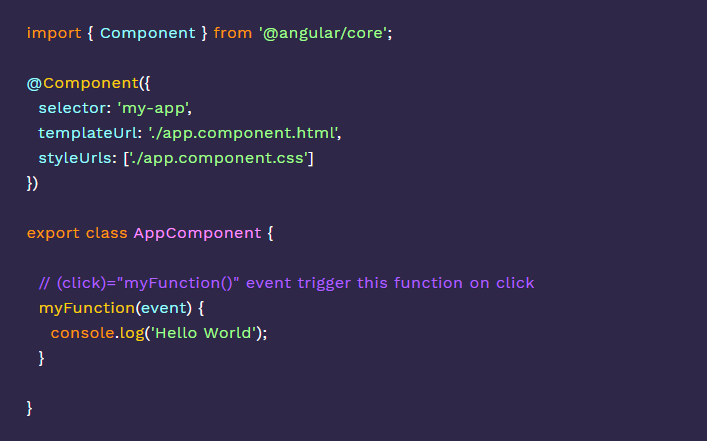


Ejemplo:

Invocación desde HTML.



Declaración del método o acción que se va ejecutar (revisar métodos con retorno y sin retorno).



Ejercicio.

1. Se requiere crear un objeto que contenga las siguientes propiedades:

* id - numero entero
* nombre - alfanumerico
* editorial - alfanumerico
* tematica - alfanumerico
* numeroPaginas - numero entero;
* precio - decimal
* autores - alfanumerico
* aColor - operador de desicion
* fechaVenta - fecha;
* estado - alfanumerico;

1. Una vez creado el objeto almacenar en una lista 5 instancias de este objeto (ejemplo).

id = 1

nombre = batman

editorial = marvel

tematica = accion

numeroPaginas = 50

precio = 60,500.99

autores = Pedrito barrera

aColor = true

fechaVenta = 10-10-2019

estado – activo

1. Imprimir la lista para ser mostrada en html en uno de los componentes que han creado.
2. Crear un botón que invoque un evento enviándole como parámetro el numero 3, para eliminar esa posición de la lista de comics y mostrar en un string el nombre del comic que se elimino, mostralo en la página html, con el mensaje “Se ha eliminado el comic nombreDelComic”, mediante interpolacion.
3. El anterior mensaje debe ser mostrado solo cuando se elimine, dentro de un <div>, controlar los posibles errores en otro div

La etiqueta div para mostrar u ocultarla usar \*ngIf=”mostrarMensaje” ejemplo:

<div ngIf=”mostarMensaje”>aquí va el mensaje</div>

NOTA INVESTIGACION: Imprimir los comics en una etiqueta de viñeta.