Parte 2

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  | **var** arreglo = ["hola", "mundo", "como", "estas"];  console.log(obtenerLongitudes(arreglo)); |
|  | **function** obtenerLongitudes(arreglo) {  **var** longitudes = arreglo.map(**function**(elemento) {  **return** elemento.length;  });  **return** longitudes;  } |



|  |  |
| --- | --- |
|  | **var** arreglo = [1, 3, 5, 7, 9];  **var** numero = 5;  console.log ("Posición del número "+ encontrarNumero (arreglo, numero)); |
|  | **function** encontrarNumero (arreglo, numero) {  **var** indice = arreglo.indexOf(numero);  **return** indice;  } |

2.

|  |  |
| --- | --- |
|  | **saludo**("Hola")("Mundo"); |
|  | **function** saludo(saludoInicial) {  **return function**(saludoFinal) {  console.log(**`${**saludoInicial**} ${**saludoFinal**}`**);  };  } |
|  |  |

3.

4. Function.prototype.bind es un método utilizado en JavaScript para crear una nueva función, que cuando se manda a llamar tiene un contexto “this” asignado por uno mismo.

5. En JavaScript, al ser un lenguaje de tipado débil, no es necesario especificar el tipo de datos de las variables, lo que hace que se pueda modificar dicho tipo para operar con ellos a voluntad. La coerción puede ser una conversión implícita o explícita. Coerción implícita cuando JavaScript transforma un valor de un tipo de datos a otro automáticamente para que pueda usarse en una operación y la coerción explícita ocurre cuando un programador quiere hacer la conversión a su manera de un valor entre dos tipos de datos.

6. Una IIFE (Immediately Invoked Function Expression) es una función que se usa para ejecutar código tan pronto como se define. Se llama justo después de que se define y es una expresión de función encerrada entre paréntesis. Ejemplo de una IIFE:

|  |  |
| --- | --- |
| 1  2  3  4 | (**function**() {  **var** mensaje = "Hola desde una IIFE!";  console.log(mensaje);  })(); |

En el ejemplo anterior, se crea un IIFE que declara una variable de mensaje dentro de la función y la imprime en la consola. Debido a que la variable de mensaje se declara dentro del alcance de IIFE, no se puede acceder a ella fuera de la función.

7. null: es un valor especial que se utiliza para indicar intencionalmente que posee un objeto o valor. Para verificar si una variable es null, se puede utilizar el operador de triple igualdad (===) comparando la variable con null.

undefined: es un valor que se asigna automáticamente a las variables que no han sido inicializadas o que no tienen un valor asignado. Para verificar si una variable es undefined, se puede utilizar el operador de triple igualdad (===) comparando la variable con undefined.

undeclared: se refiere a una variable que no ha sido declarada previamente en ningún lado del código. Para verificar si una variable ha sido declarada, se puede utilizar la función typeof. Si la variable no ha sido declarada, la función typeof devuelve "undefined".

8. Los principales componentes de Angular son 7, lo cuales son:

* Templates: representan la vista de la aplicación y definen cómo se deben mostrar los datos.
* Directivas: son atributos que se utilizan en los elementos HTML para mejorar su comportamiento o modificar su apariencia.
* Componentes: son partes del aplicativo que están asociadas a una vista y una lógica específica que maneja la interacción con el usuario.
* Servicios: son clases que proporcionan funcionalidades compartidas en toda la aplicación, ejemplo la comunicación con un servidor.
* Inyección de dependencias: es un patrón de diseño que permite a las clases obtener las dependencias necesarias mientras se están ejecutando, en lugar de crearlas directamente.
* Módulos: son contenedores que agrupan componentes, servicios y otros módulos relacionados y les brindan un contexto de ejecución para ellos.
* Routing: es un mecanismo, utilizado para navegar entre diferentes páginas de la aplicación.

9. JSX es una extensión de la sintaxis de JavaScript que permite escribir HTML (o XML) en el mismo archivo que el código JavaScript. JSX permite escribir código de mantener, ya que permite crear componentes poseen tanto la lógica como la presentación visual en un mismo archivo.

10. En Angular, el "state" se refiere a un conjunto de datos que representan el estado actual de la aplicación. El estado puede incluir datos que se actualizan dinámicamente, como los valores de entrada del usuario, las respuestas del servidor, etc.

El método render () se utiliza para renderizar un elemento en el DOM después de haberlo modificado o creado dinámicamente. Este método se encuentra en el servicio Renderer2, que es una interfaz proporcionada por Angular para manipular el DOM de forma segura y eficiente.

11. El Router de Angular funciona de la siguiente manera:

* Primero se define un conjunto de rutas en la aplicación usando la clase Routes del módulo @angular/router. Cada ruta se define como un objeto el cual posee la URL de la ruta y el componente que se debe cargar cuando se navega a esa ruta.
* Se importa y se configura el módulo RouterModule en la aplicación para que sepa qué rutas se deben reconocer. Esto se hace utilizando la función forRoot() del módulo, que acepta un arreglo de rutas como argumento.
* Se incluye el componente router-outlet en el archivo HTML principal de la aplicación.
* Cuando se navega a una ruta en la aplicación, el Router analiza la URL y determina qué ruta se debe cargar. Luego, carga el componente correspondiente y lo muestra en el lugar del componente router-outlet.
* Cuando se navega a otra ruta en la aplicación, el Router repite el proceso de carga y renderizado de componentes.