

## EVALUACIÓN DESARROLLADOR FRONT END

**Nombre: Leonardo Lira Cazares**

**Fecha: 02/03/2020**

Te pedimos que leas atentamente las siguientes indicaciones de la evaluación práctica, de tal modo que puedas resolver cada una de manera escrita.

1. Al ejecutar esta expresión ¿que debe devolver la consola?

```
const miArreglo = new Array(16).join('js'-1);  
console.log(miArreglo);
```

**R= (NaN) en 15 espacios, en otras palabras**

**"NaNNaNNaNNaNNaNNaNNaNNaNNaNNaNNaNNaNNaNNaNNaN"**

2. Sigue las instrucciones de la siguiente solicitud:

- Declara una función con el nombre reverse
- Esta función debe recibir un string como argumento
- La función debe devolver el mismo string pero en reversa. Es decir que, si recibe "Hola" como argumento, la función debe devolver "aloH".

```
function reverse(cadena) {  
  if (typeof(cadena)=== 'string') {  
    var tamano_cadena = cadena.length;  
    var nueva_cadena = "";  
    for (i = tamano_cadena; i >= 1; i--) {  
      nueva_cadena = nueva_cadena + cadena.slice(i - 1, i);  
    }  
    return nueva_cadena;  
  } else {  
    return false;  
  }  
}
```

- 3- ¿Qué es "scope"?

R= Scope, es el enlace o intermediario entre las vistas y los controladores en el framework angularjs, nos traslada o nos pasa los datos de las vistas al controlador o viceversa.

- 4.- Lea la expresión y conteste las siguientes preguntas:

```
Const miObjeto = {"Mi única propiedad": "Hola"};
```

- a) ¿Existe una forma de saber el valor de "Mi única propiedad"?

R= SI, miObjeto["Mi única propiedad"];

- b).-¿La variable miObjeto es válida? SI - NO ¿Por qué?:

R=Si, ya que un objeto se componen de un índice o identificador y su propiedad y en este caso el identificador "Mi única propiedad" y su propiedad "Hola", no es buena practica que existan espacios en los identificador o índices, ya que puede generar problemas a largo plazo en nuestras funciones.

5.-Utilice ES6 para reducir la siguiente función a una sola línea:

```
Function sumar (a,b){  
  Let suma = a + b;  
  Return suma;  
};  
Sumar(2,2);
```

R= var sumar = (a, b) => { return a + b }

6- La función anterior (sumar) recibe dos parámetros (a y b). ¿El cliente ha solicitado que la función sea capaz de recibir "n" parámetros de tal forma que pueda ser invocada de la siguiente forma:

Sumar (2,3,6,9,7,4,5,3,7,2,4,7,42,1,2,2,6,7,8);

Por favor modifique la función para que pueda realizar dicha función y el cliente quede satisfecho.

```
R= function sumar () {  
  var resultado = 0;  
  for (i = 0; i < arguments.length; i++) {  
    resultado += arguments[i];  
  }  
  return resultado;  
}
```