Trabajo Practico Type Script

1- Convertir esta función JS a una función de flecha de TS

```
var resultadoDoble = function (a, b) { return (a + b) * 2; };
console.log(resultadoDoble(2, 2));
```

2- Convierte la siguiente función JS a una función TS, la función contendrá parámetros obligatorios, opcionales y otros serán por defecto si no le pasamos ninguno.
Deberás adaptarlos de la siguiente forma:
* nombre → obligatorio
* edad → opcional
* arma → por defecto u omisión será "pistola"

```
function getVillano( nombre, edad, arma ) {
  var mensaje;
  if( edad ) {
    mensaje = nombre + " tiene una edad de: " + edad + " y arma: " + arma;
  } else {
    mensaje = nombre + " tiene un " + edad
  }
};
```

3- Cree una función Type Script que devuelva el número de vocales en un string ingresado, el String contendrá letras minúsculas y espacios.
Ejemplo:
Ingreso: “esto es una prueba de codigo”
Salida: se encontraron 12 vocales

4- Dada la siguiente enumeración de Type Script

```
enum Movimiento {
  PIEDRA, PAPEL, TIJERAS
}
```

Y la clase Jugada

```
class Jugada{
      jugador1: Movimiento;
      jugador2: Movimiento;
}
```

Crea un programa que calcule quien gana más partidas al piedra, papel, tijera.
- La función recibe un listado que contiene Jugadas.
- Cada Jugada puede contener combinaciones de movimientos "R" (piedra), "P" (papel) o "S" (tijera).
- El resultado puede ser: "Gana 1", "Gana 2", "Empate"
- Ejemplo.
Entrada: [Jugada, Jugada, Jugada].
Resultado: "Empate", “Gana 2”, “Gana 1”.

5- Dado el siguiente código JSON, cree una interfaz que sirva después para validar el objeto.

```
var reptil = {
        nombre: "cobra",
        tipo: "serpiente",
        paisOrigen: "India",
        pesoPromedio: 2.5,
        esVenenosa: true
}
```

6- Deberá crear una clase **FiguraGeometrica** con dos propiedades numéricas decimales:

1 → Base

2 → Altura

También deberá tener un método para calcular el área (Base*Altura), este método deberá devolver un valor numérico.

A partir de la clase anterior deberá crear 2 clases hijas Rectángulo y Triangulo.

También deberá sobrescribir el método para calcular el área en el rectángulo: (Base*Altura) y en el triángulo: ((Base*Altura)/2)

7- Cree 2 archivos programa.html y programa.ts (el resultante de la compilación de este archivo será un 3 archivo programa.js)

El archivo programa.html tendrá el siguiente contenido:

```
<html>
<head>
</head>
<body>

<h1>Carga Datos Vehiculos</h1>

<form>
<input type="text" id="marca" placeholder="Ingrese marca del Vehiculo" />
<input type="text" id="modelo" placeholder="Ingrese modelo del Vehiculo" />
<input type="text" id="patente" placeholder="Ingrese patente del Vehiculo" />
<input type="button" value="Cargar Vehiculo" onclick="guardarVehiculo()" />
</form>

<h3>El listado de autos cargados es:</h3>
<div id="listadoVehiculos" />

<script type="text/javascript" src="programa.js" />

</body>
</html>
```

Codifique mediante Type Script la función guardarVehiculo() en la cual deberá crear una instancia de un clase vehículo que contendrá los atributos marca, modelo y patente, almacenar ese objeto en un array de vehículos y finalmente recorrer el array y cargar los vehículos en el div listadoVehiculos del archivo HTML haciendo uso de la etiqueta <li> de HTML y la función innerHTML

PSEUDOCODIGO

```
Clase Vehiculo{
        marca : string
        modelo : string
        patente : string
}

vehículos : Vehiculo[]
function guardarVehiculos(){
        Vehiculo v = new Vehiculo()
        v.marca = valor cargado en el input marca
        v.modelo = valor cargado en el input modelo
        v.patente = valor cargado en el input marca

        vehículos.push(v)

        listHTML:string = “<ul>”;
        for(recorro la lista de vehiculos){
                listHTML += “<li>” + marca + “ ” + modelo + “ ” + patente + “</li>”
        }
        listHTML:string = “</ul>”;
}
```

8- Dada la siguiente clase Type Script

```
class Cerveza {
  id: number
  uid: string
  brand: string
  name: string
  style: string
  hop: string
  yeast: string
  malts: string
  ibu: string
  alcohol: string
  blg: string
}
```

Y haciendo uso de la siguiente URL
https://random-data-api.com/api/beer/random_beer?size=25

Codifique un programa en Type Script que ejecute una petición a la URL anterior y cargue la respuesta en un array de objetos Cerveza.
Finalmente ejecute la iteración del array y muestre los datos brand, name, style, ibu y alcohol haciendo uso de un console.log