TypeScript - Tipos básicos

Objetivos: Anotar tipos primitivos, arreglos y cualquier tipo. Identificar cuando ocurre la verificación de tipos. Luego analizaremos la salida del transpiler

Ejercicio 1.1

Dado el siguiente codigo:

1. Coloca el cursor sobre los garabatos rojos para inspeccionar los errores de TS
2. Coloca el cursor sobre las variables para inspeccionar sus tipos
3. Arregle el error en la linea 2 cambiando el valor de pi al tipo esperado

```
let pi = '3.14159';
let tau = pi * 2;

console.log('[Ejercicio 1.1]', `${tau} es ${pi} veces el dos.`);
```

Ejercicio 1.2

Dado el siguiente codigo:

1. Inspeccionar el tipo de ‘torta‘
2. Añadir una anotación de tipo explícito a ‘torta‘
3. Intenta asignar tipos inválidos, por diversión

```
let torta;
torta = 'arandanos';

console.log('[Ejercicio 1.2]', `Me gusta comer torta con sabor a ${torta}.`);
```

Ejercicio 1.3

Dado el siguiente codigo:

1. Inspeccione el error, luego corríjalo

```
let esPablo: boolean;

console.log('[Ejercicio 1.3]', `${esPablo ? 'Oh, hola Pablo' : 'Quien sos vos?'}`);
```


Ejercicio 1.4

Dado el siguiente codigo:

1. Añadir anotaciones de tipo (lo más explícitas posible)
2. Solucionar errores (si corresponde)

```
const entero = 6;
const decimal = 6.66;
const hexadecimal = 0xf00d;
const binario = 0b1010011010;
const octal = 0o744;
const ceroNegativo = -0;
const enRealiadadNumero = NaN;
const mayorNumero = Number.MAX_VALUE;
const elNumeroMasGrande = Infinity;

const miembros: any[] = [
  entero,
  decimal,
  hexadecimal,
  binario,
  octal,
  ceroNegativo,
  enRealiadadNumero,
  mayorNumero,
  elNumeroMasGrande
];

miembros[0] = '12345';

console.log('[Ejercicio 1.4]', miembros);
```

Ejercicio 1.5

Dado el siguiente codigo:

1. Añadir anotaciones de tipo (lo más explícitas posible)
2. Solucionar errores (si corresponde)

```
const secuencia = Array.from(Array(10).keys());
const animales = ['pinguino', 'oso hormiguero', 'zorro', 'jirafa'];
const cadenasYNumeros = [1, 'uno', 2, 'dos', 3, 'tres'];
const todosMisArreglos = [secuencia, animales, cadenasYNumeros];

console.log('Ejercicio 1.5', todosMisArreglos);
```


Ejercicio 1.6

Queremos representar un elemento de inventario como una estructura donde la primera entrada es el nombre del artículo y la segunda es la cantidad.

Dado el siguiente codigo:

1. Añadir anotaciones de tipo (lo más explícitas posible)
2. Solucionar errores (si corresponde)

```
const elementoInventario = ['tuerca', 11];

// despues lo desestructuramos
const [nombre, cantidad] = elementoInventario;

const mensaje = agregarInventario(nombre, cantidad);

console.log('[Ejercicio 1.6]', mensaje);

function agregarInventario(nombre: string, cantidad: number): string {
  return `Se agregaron ${cantidad} ${nombre}s al inventario.`;
}
```

