

## Typescript-Exercícios-01

1) Verifique nas alternativas abaixo quais compilam ou não. Explique o motivo:

a) `let a = 10;`  
`a = "2";`

**Resposta:** Não compila, pois a foi definido com tipo number e logo em seguida a recebe um valor do tipo string ou seja um tipo incompatível com o definido.

b) `let b: any = 10;`  
`b = 2;`

**Resposta:** Compila, pois b foi definido por qualquer tipo ou seja qualquer tipo que ele vim a receber em uma atribuição vai ser compilado com o tipo de b que no caso e tipo number resultando no valor 2 já que foi a última atribuição dada a b.

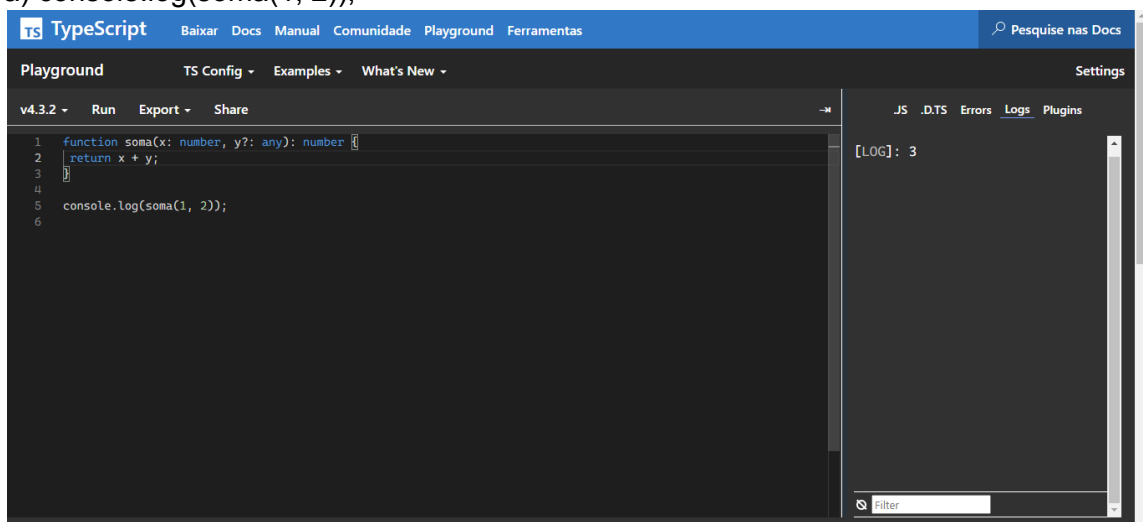
c) `let c: number = 10;`  
`c = 2;`

**Resposta:** Compila, pois c foi definido como uma variável do tipo number que recebe o valor 10 depois recebe o valor dois sendo esse seu valor final não ocorre erro pois c e 2 são do mesmo tipo(Number).

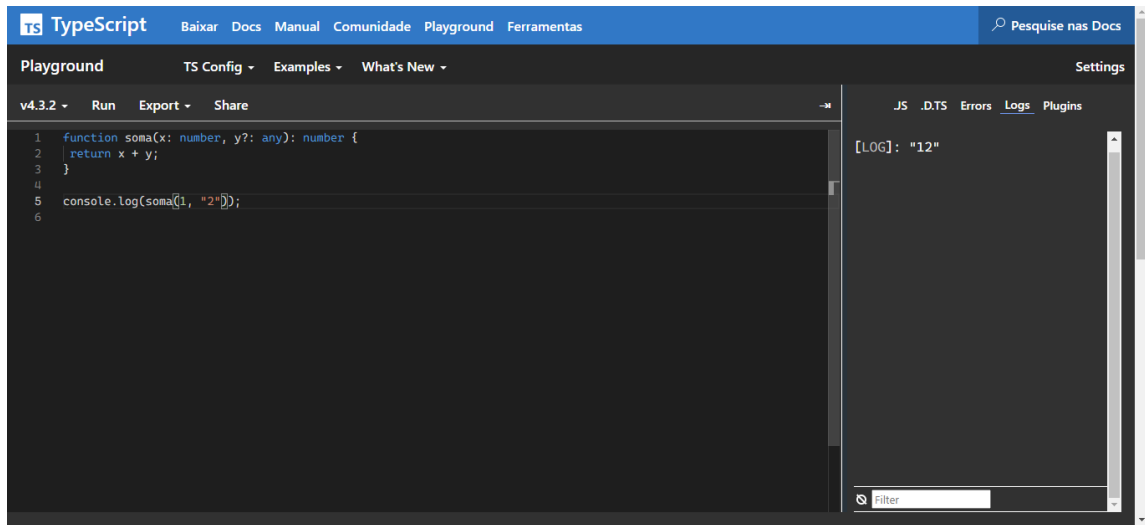
2) Dada a função soma abaixo, tente executar os scripts das alternativas e exiba os eventuais resultados:

```
function soma(x: number, y?: any): number {  
  return x + y  
}
```

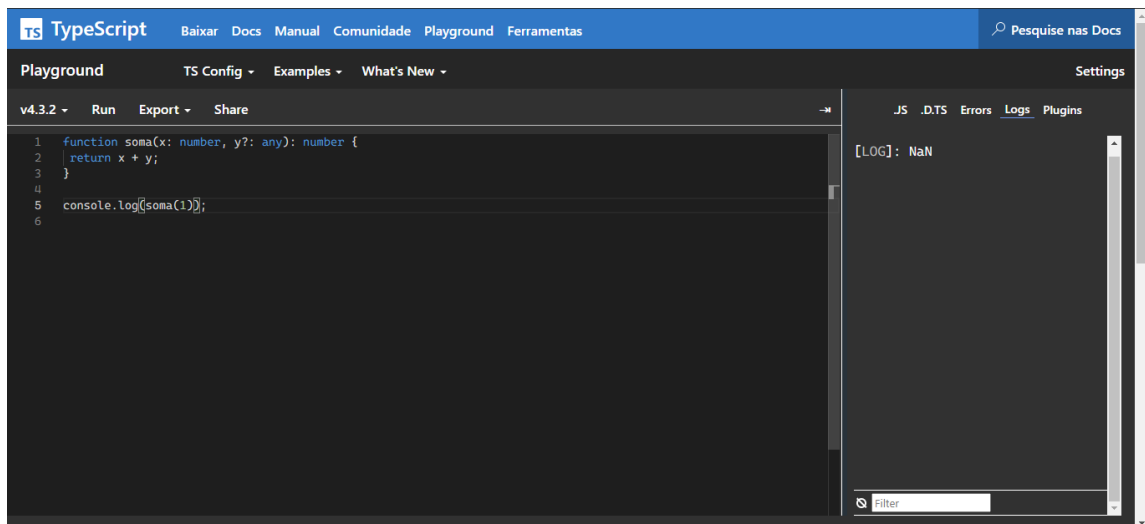
a) `console.log(soma(1, 2));`



b) `console.log(soma(1, "2"));`



c) `console.log(soma(1));`



3) Crie uma enum com as siglas dos estados "PI", "CE", "MA" e implemente as duas alternativas abaixo:

a) Crie um laço usando for para imprimir esses valores;

```
1 enum Estados {  
2   est_1 = "PI", est_2 = "CE", est_3 = "MA"  
3 }  
4 for (let estado in Estados) {  
5   let es = estado;  
6   console.log(es);  
7 }  
8
```

The right sidebar shows the logs: [LOG]: "est\_1", [LOG]: "est\_2", [LOG]: "est\_3".

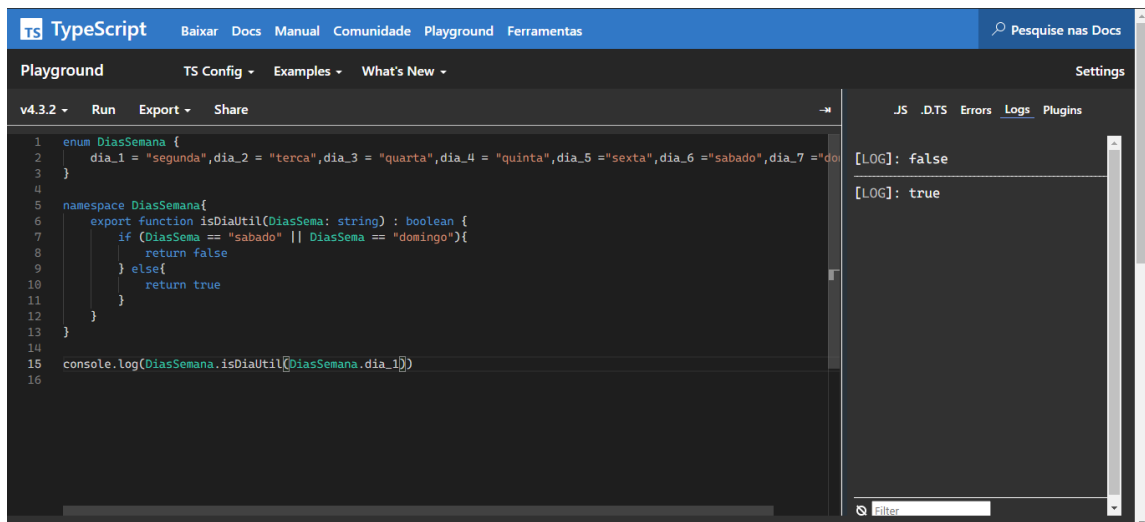
b) Crie um laço que imprima os índices dessa enum e diga o que aconteceu.

```
1 enum Estados {  
2   "PI", "CE", "MA"  
3 }  
4 for ([let estado in Estados]) {  
5   let es = estado;  
6   console.log(es);  
7 }  
8
```

The right sidebar shows the logs: [LOG]: "0", [LOG]: "1", [LOG]: "2", [LOG]: "PI", [LOG]: "CE", [LOG]: "MA".

**Resposta:** O programa percorre o enum e armazena o index de cada string na variável es em cada volta e em seguida faz o print de cada index na ordem do loop.

- 4) Sobre enums, implemente o seguinte:
- a) Crie uma enum chamada DiasSemana com os valores representando os dias da semana segunda a domingo;
  - b) Crie um namespace com mesmo nome e dentro dele crie uma função chamada isDiaUtil recebe um parâmetro do tipo DiasSema e retorna false se for um sábado ou domingo e retorna true caso contrário;
  - c) Escreva também um script que declara uma variável do tipo da enum e que testa a função DiasSemana.isDiaUtil().

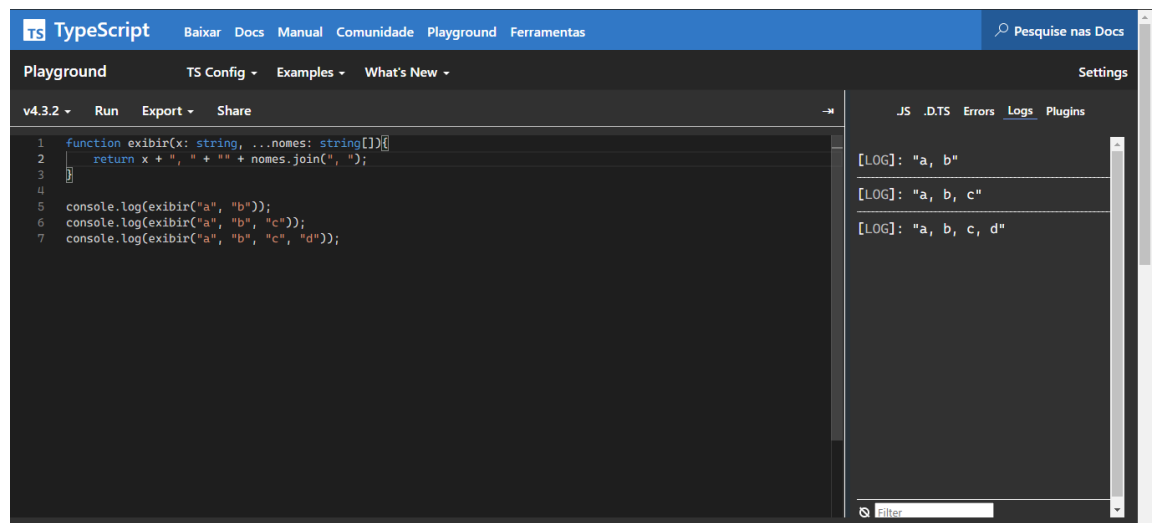


- 5) Crie uma função chamada `exibir` que receba como parâmetro um “rest parameter” representando strings. A função deve exibir no log cada um dos elementos do “rest parameter”. Chame a função usando diferentes quantidades de parâmetros conforme abaixo:

```

exibir("a", "b");
exibir("a", "b", "c");
exibir("a", "b", "c", "d");

```

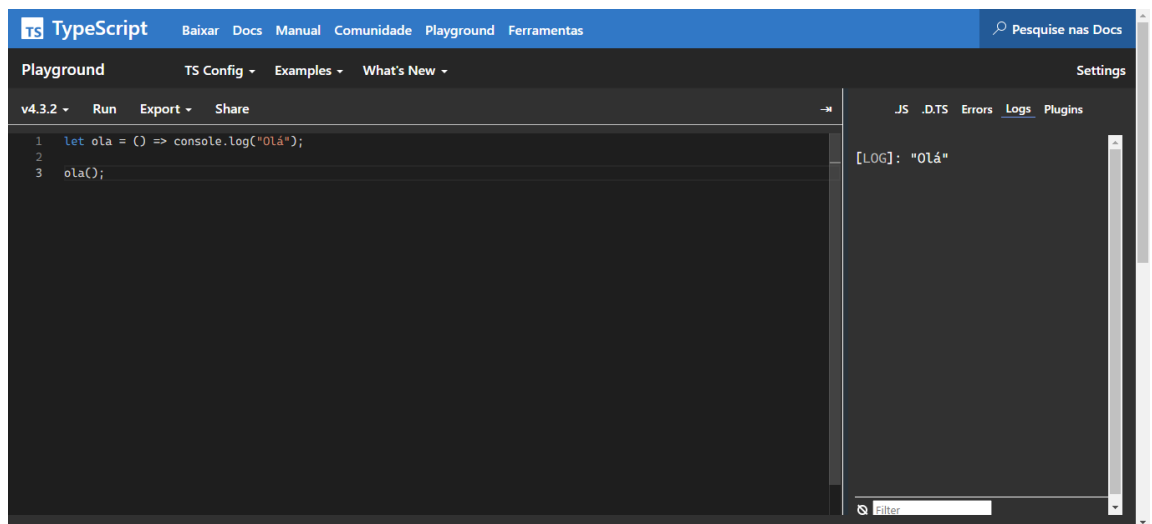


- 6) Converta em arrow function a seguinte função:

```

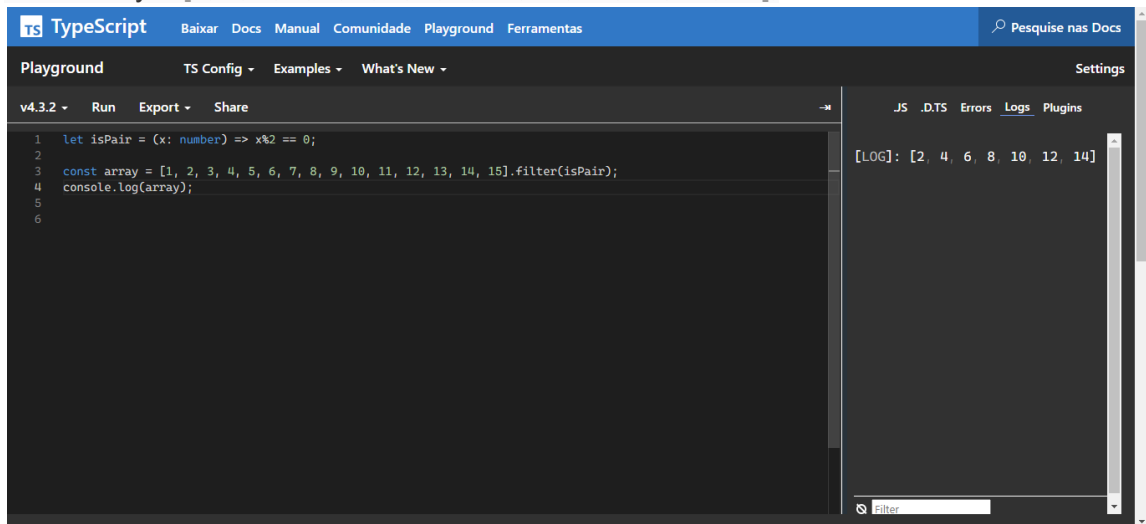
function ola() {
  console.log("Olá");
}

```



- 7) Dado método filter dos arrays, crie uma implementação usando arrow function que filtre todos os elementos pares do array abaixo:

`const array = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15];`



- 8) Crie uma classe chamada MeuNumero tenha:

- Um atributo chamado numero que seja somente leitura e inicializado no construtor;
- Um método chamado getInteiro que retorna a parte inteira do atributo numero;
- Um método chamado getDecimal que retorne a parte decimal do atributo numero.

Dica: utilize a função `Math.floor(n)`

- Instancie uma classe MeuNumero e teste os métodos da classe.

The screenshot shows the TypeScript Playground interface. The main editor contains the following code:

```
1 class MeuNumero {
2   readonly numero: number;
3
4   constructor(numero: number) {
5     this.numero = numero;
6   }
7
8   getInteiro() {
9     let integer_number = Math.floor(this.numero);
10    return integer_number;
11  }
12
13  getDecimal() {
14    let decimal_number = this.numero - Math.floor(this.numero);
15    return decimal_number;
16  }
17 }
18
19 var numero = new MeuNumero(6.25)
20 console.log(numero.getInteiro())
21 console.log(numero.getDecimal())
22
23
```

On the right side, the 'Logs' tab is selected, showing the output of the code execution:

```
[LOG]: 6
[LOG]: 0.25
```

9) Crie uma classe chamada Transacao que tenha:

- Um atributo chamado valor e um outro chamado desconto, ambos somente leitura;
- Um método que calcule e retorne o valor do desconto aplicado ao valor original:  $\text{valor} * (1 - \text{desconto}/100)$ .
- Crie métodos de acesso get para ambos os atributos.
- Instancie uma classe Transacao e teste seus métodos