

## Exercício 01 - Typescript

Prof. Ely Miranda

Aluno: Antônio Gabriel da Silva Moura

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

a) let a = 10;

a = "2";

**R: Não irá compilar, pois não há como uma variável do tipo "number", receber um atributo do tipo "string".**

b) let b: any = 10;

b = 2;

**R: Irá compilar corretamente, pois o tipo "any" permite que a variável "b" receba valores de qualquer outro tipo.**

c) let c: number = 10;

c = 2;

**R: Irá compilar corretamente, pois a variável "c" permite valores do tipo "number", então ela irá aceitar qualquer valor que se enquadre nesse tipo.**

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));

**R: O resultado exibido será 3. A soma da função será executada com sucesso, já que ambos os valores são do tipo "number".**

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

**R: O resultado exibido será 12. Pois ocorreu uma concatenação entre os valores por serem um "number" e uma "string".**

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

**R: O resultado exibido será NaN. Pois a soma da função não pode ser realizada por tentar somar um "number" com um valor indefinido.**

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;

```
enum siglaEstado{  
    Sigla1 = "PI",  
    Sigla3 = "CE",  
    Sigla2 = "MA"  
}  
  
for(let i in siglaEstado){  
    console.log(siglaEstado[i]);  
}
```

Resultado:

```
PI  
CE  
MA
```

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

```
enum siglaEstado{  
    Sigla1 = "PI",  
    Sigla3 = "CE",  
    Sigla2 = "MA"  
}  
  
for(let i in siglaEstado){  
    console.log(i);  
}
```

Resultado:

```
Sigla1  
Sigla2  
Sigla3
```

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().

```
enum diasSemana{

    Segunda = 1,
    Terca = 2,
    Quarta = 3,
    Quinta = 4,
    Sexta = 5,
    Sabado = 6,
    Domingo = 7
}

namespace diasSemana {
    export function isDiaUtil(diaSema: string) : boolean {
        if(diaSema == '6' || diaSema == '7'){
            return false;
        }
        return true;
    }
}

console.log(diasSemana.isDiaUtil('6'));
Resultado:
false
console.log(diasSemana.isDiaUtil('3'));
Resultado:
true
```

5) Crie uma função chamada `exibir` 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 quantidade de parâmetros conforme abaixo:

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

```
function exibir(...parameter: string[]){  
  
    return parameter  
}  
  
console.log(exibir("a", "b"))  
console.log(exibir("a", "b", "c"))  
console.log(exibir("a", "b", "c", "d"))
```

Resultado:

```
[ 'a', 'b' ]  
[ 'a', 'b', 'c' ]  
[ 'a', 'b', 'c', 'd' ]
```