

Téc em Desenvolvimento  
de Sistemas Bilíngue

Desenvolver Código

Orientado a Objetos

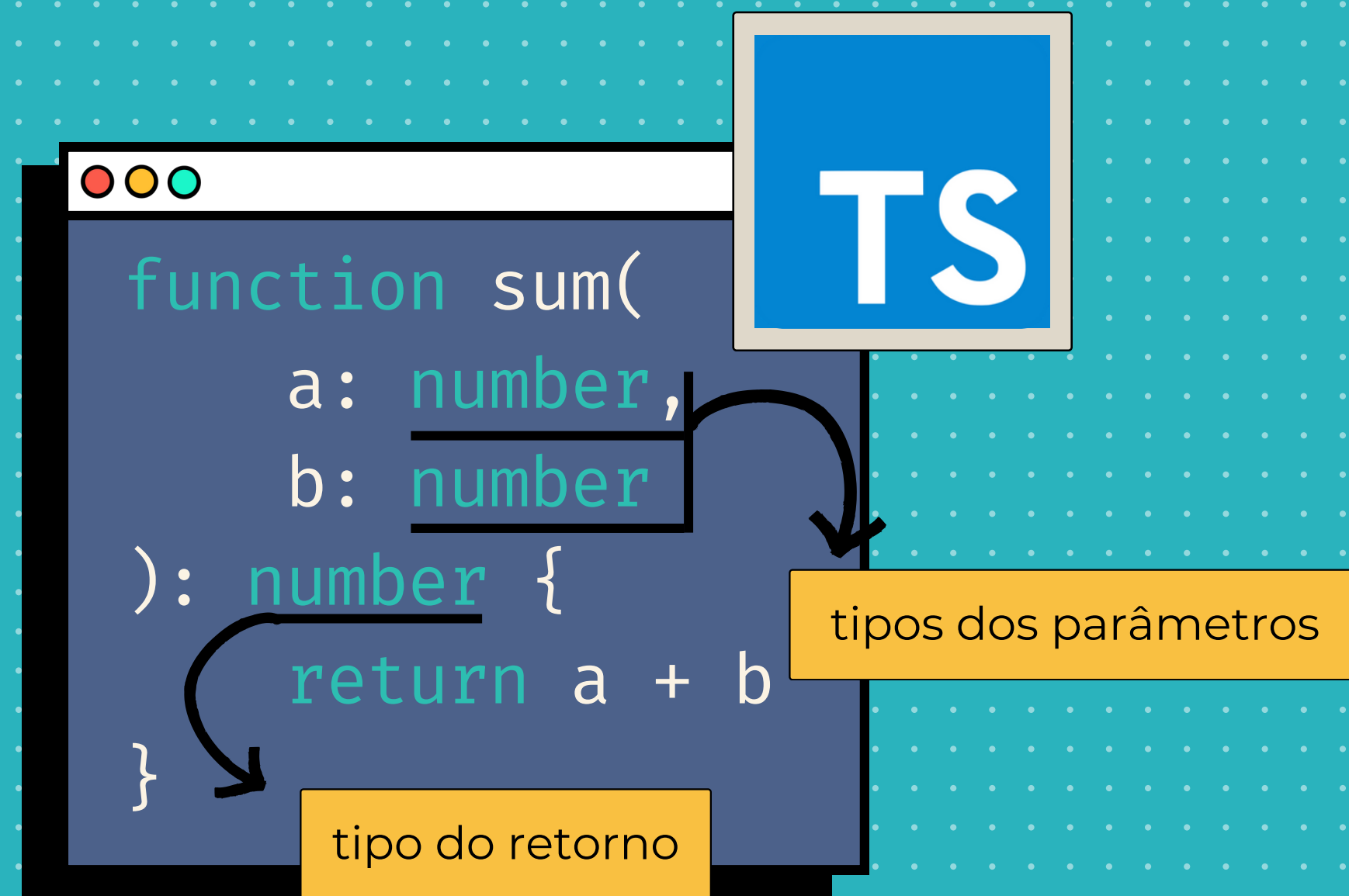
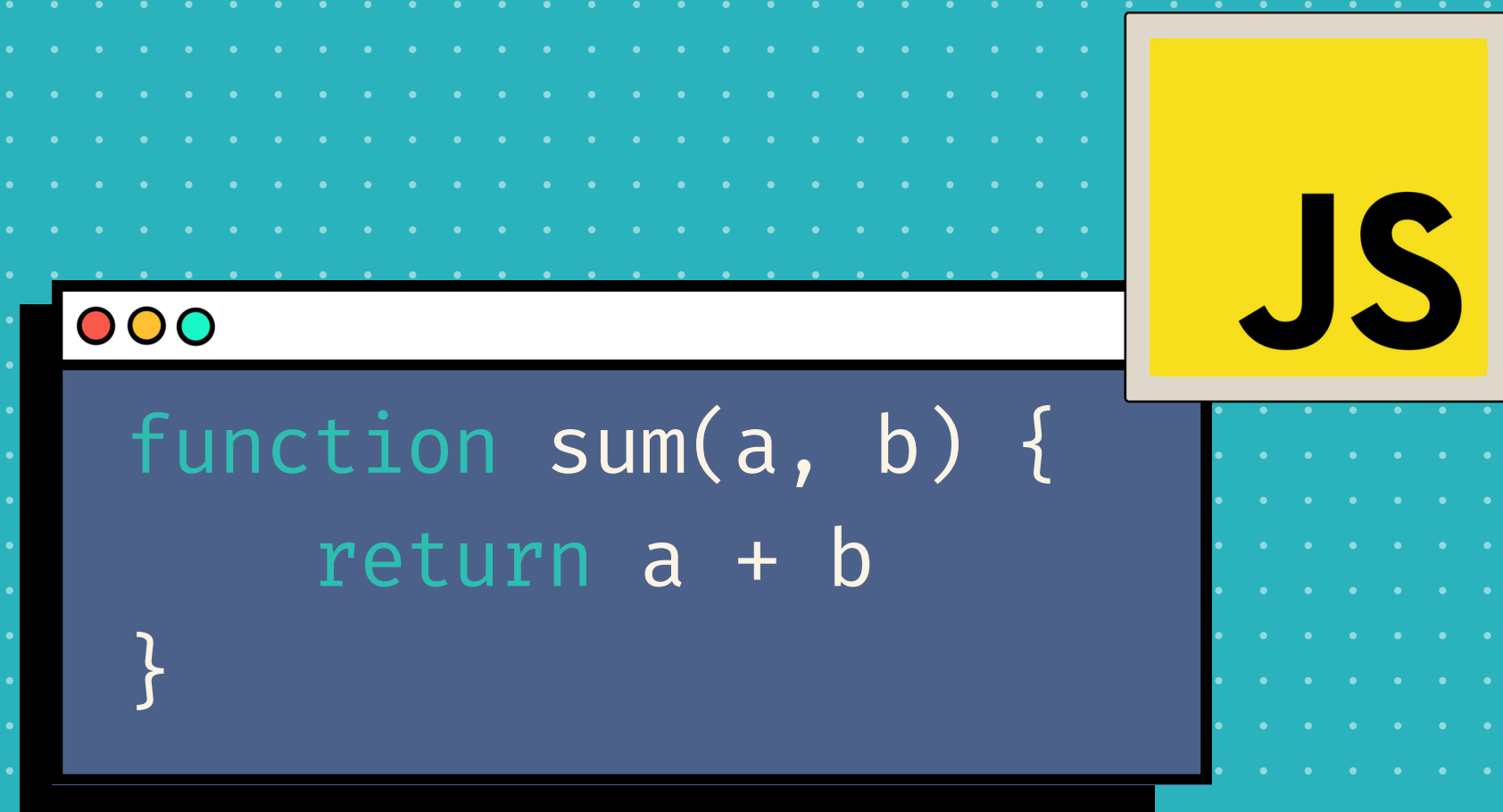
UC4 | Prof. Vitor Hugo Lopes

## Funções Tipadas

No **Javascript**, podemos criar uma função que recebe dois números e chamá-la passando duas strings, por exemplo. Isso abre margem para bugs!

No **Typescript**, podemos tipar sua entrada (parâmetros) e saída (retorno)

```
function sum(a, b) {  
  return a + b  
}
```



## Parâmetros Opcionais

Nem sempre todos os dados que a função recebe são obrigatórios. Eventualmente precisaremos utilizar parâmetros opcionais, representados pelo ponto de interrogação.

```
function sayHello(name?: string): void {  
  console.log("Hello,", name || "World")  
}
```



parâmetro opcional

tipo informado quando  
não há retorno

```
function sayHello(name?: string): void {  
  console.log("Hello, ", name || "World")  
}
```

```
function applyDiscount(  
  price: number,  
  discount: number = 0.05  
): number {  
  return price * (1 - discount)  
}
```

quando não enviado o valor de  
"discount" a função usará "0.05"

## Tipo Any

Há um tipo especial, que deve ser evitado, mas as vezes é a única opção: **any**. Indica que a variável pode assumir qualquer valor

```
let aux: any
aux = "aux"
aux = 0
aux = true
```

## tsconfig.json

Em um projeto com vários arquivos, o processo de transpilação que vimos até agora é muito pouco prático.

Por isso, vamos criar um arquivo de configuração chamado **tsconfig.json** dentro da pasta do nosso projeto, usando o comando **npx tsc --init**



## tsconfig.json

Configurando esse arquivo só precisamos usar o comando **npx tsc** no script, e, depois, rodar os javascripts que forem criados.

```
"start": "npx tsc && node ./build/index.js"
```

# tsconfig.json

```
{  
  "compilerOptions": {  
    "target": "es6",           /* Specify ECMAScript target version */  
    "module": "commonjs",     /* Specify module code generation */  
    "sourceMap": true,        /* Generates corresponding '.map' file. */  
    "outDir": "./build",      /* Redirect output structure to the directory. */  
    "rootDir": "./src",       /* Specify the root directory of input files. */  
    "removeComments": true,   /* Do not emit comments to output. */  
    "noImplicitAny": true,    /* Raise error on declarations with an implied 'any' type. */  
  }  
}
```

## #DicaDoProf

Se você tem mais de um exercício na mesma pasta pode criar diferentes starts dentro do package.json. Por exemplo, considere que temos uma lista com 3 questões:

```
"start1": "npx tsc && node ./build/index1.js",  
"start2": "npx tsc && node ./build/index2.js",  
"start3": "npx tsc && node ./build/index3.js"
```



Obrigado!

Dúvidas? Escreva para [vhlopes@senacrs.com.br](mailto:vhlopes@senacrs.com.br)