



JavaScript Básico - Prática

Desafio: Validador de CPF I

O que estamos usando?

Organização é importante

Tag `<script>`

Lidando com Eventos dentro do JS

Desafio: Validador de CPF II

Capturando valores no JS

if(resultadoValidacao)

Minhas alterações

Desafio: Validador de CPF II

Validando a função CPF

Desafio: Validador de CPF III

Resto da divisão %

Console

Desafio: Validador de CPF III

Limpando as validações na tela força bruta

Desafio: Validador de CPF I

O que estamos usando?

- Bootstrap
- JQuery
- Importamos todos no final do arquivo.

Organização é importante

- Deixe seu arquivo identado.

Tag `<script>`

- Conectando o HTML com o JavaScript
- Colocamos ao final do nosso `body`

```
<script src="/script.js"></script>
```

Lidando com Eventos dentro do JS

```
const cpfButton = document.querySelector('.btn')

function cpfValidation () {

}

cpfButton.addEventListener('click', () => {
  console.log('Oi')
})
```

Desafio: Validador de CPF II

Capturando valores no JS

- Devemos indicar um ID para o `input` e depois capturamos o valor de dentro dele.

```
const cpfButton = document.querySelector('.btn')

function cpfValidation () {
  const cpf = document.getElementById('cpf_digitado').value

  console.log(cpf)
}

cpfButton.addEventListener('click', cpfValidation)
```

if(resultadoValidacao)

- Não precisamos ser redundantes `if(resultadoValidacao == true)`
- Já checa se o resultado é verdadeiro.

Minhas alterações

- Shorthand com operador ternário
- Se verdadeiro executa a primeira `?`, se falso executa a segunda `:`

```
validationResult ? document.getElementById('success').style.display = 'block'  
: document.getElementById('error').style.display = 'block'
```

Desafio: Validador de CPF II

Validando a função CPF

- Tratamos String como posições de um Array.

CharAt

- Retorna a posição do índice na lista.

```
function isCpfValid(cpf) {  
  if(cpf.length !== 11) {  
    return false  
  }  
  
  let numbers = cpf.substring(0, 9)  
  let chars = cpf.substring(9)  
  
  let increment = 0  
  for (let i = 10; i > 1; i--) {  
    increment += numbers.charAt(10 - i) * i  
  }  
}
```

Desafio: Validador de CPF III

Resto da divisão %

- Vamos pegar o resto da divisão.

Console

- O console é seu melhor amigo para debuggar.

```

let increment = 0
for (let i = 10; i > 1; i--) {
    increment += numbers.charAt(10 - i) * i
}

let result = (increment % 11) < 2 ? 0 : 11 - (increment % 11)

if(result !== chars.charAt(0)) {
    return false
}
}

```

Desafio: Validador de CPF III

```

increment = 0
numbers = cpf.substring(0, 10)

for (let k = 11; k > 1; k--) {
    increment += numbers.charAt(11 - k) * k
}

result = increment % 11 < 2 ? 0 : 11 - (increment % 11)

if (result !== chars.charAt(1)) {
    return false
}

return true

```

Limpendo as validações na tela força bruta

```

document.getElementById('success').style.display = 'none'
document.getElementById('error').style.display = 'none'

```