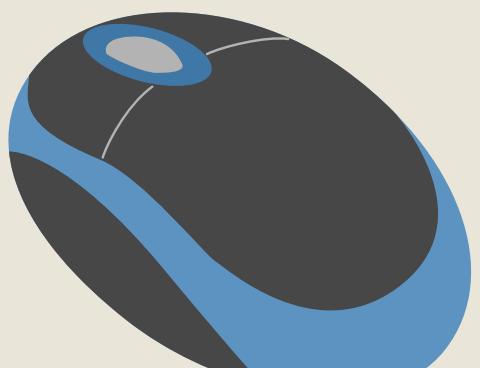
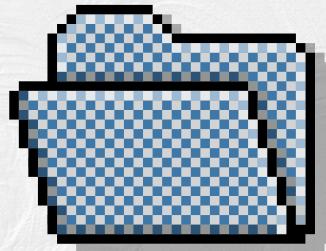
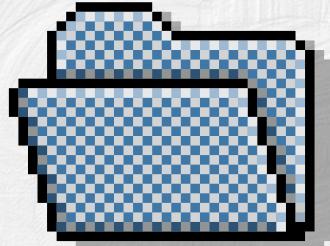
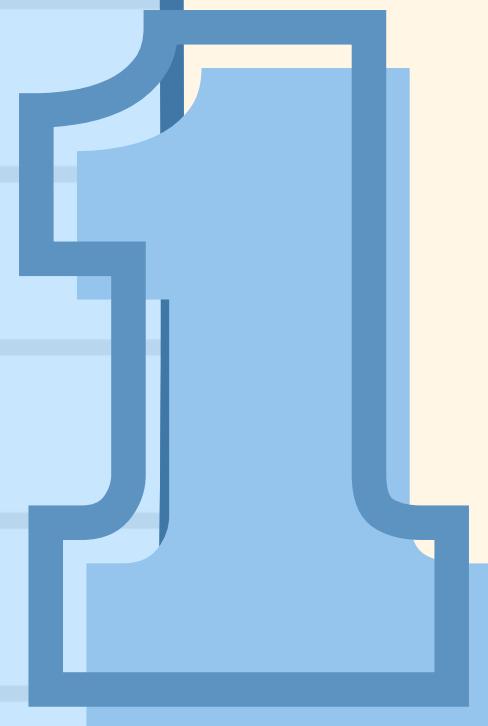


Eloize M. - N°03
Empresa de T.I.

folha de *pagamento*

Continuar





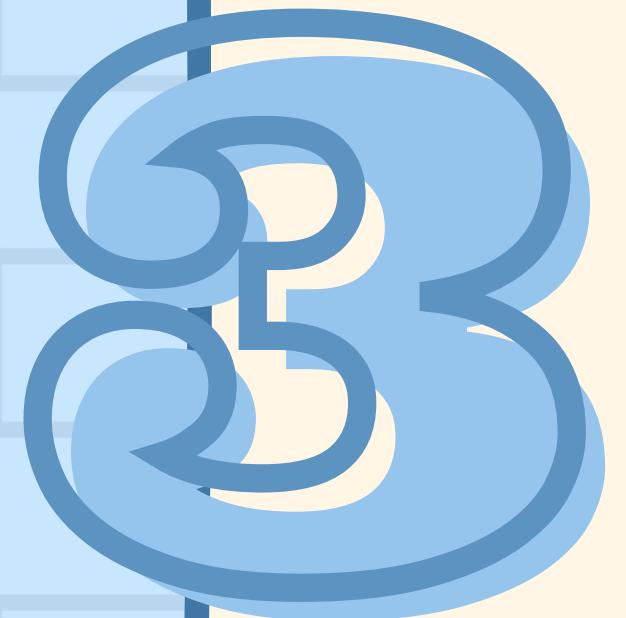
Adicione a pasta “backend” ou
“folhadepagamento” no
VS CODE

Passo 1



Crie a pasta “script.js” no
VS CODE

Passo 2



Adicione a lista de funcionários com
o seguinte código:

Passo 3



```
script.js  X  
script.js > ...  
1  const listaFuncionarios = [  
2      {id: 1, nome: 'Kemmilly Carvalho', setor: 'TI', salario: 7000, bonus: true},  
3      {id: 2, nome: 'Isabelly Oliveira', setor: 'TI', salario: 12000, bonus: false},  
4      {id: 3, nome: 'Maria Eduarda Dias ', setor: 'TI', salario: 10000, bonus: true},  
5      {id: 4, nome: 'Ana Laura Paulino', setor: 'TI', salario: 12000, bonus: false},  
6      {id: 5, nome: 'Isabella Rosa', setor: 'TI', salario: 7000, bonus: true},  
7  ]
```

CÓDIGO



Funcionários do setor "TI":
Todos os funcionários cadastrados
são do setor de T.I.

Passo 4



FILTER

O método `filter()` no JavaScript é uma função de alta ordem (HOF) usada para **selecionar elementos específicos** de um array, criando um novo array com apenas os itens que **passam em uma condição lógica**.

pesquisa

5

FILTER

Adicione o código para selecionar apenas os
funcionários que possuem bônus

Passo 5



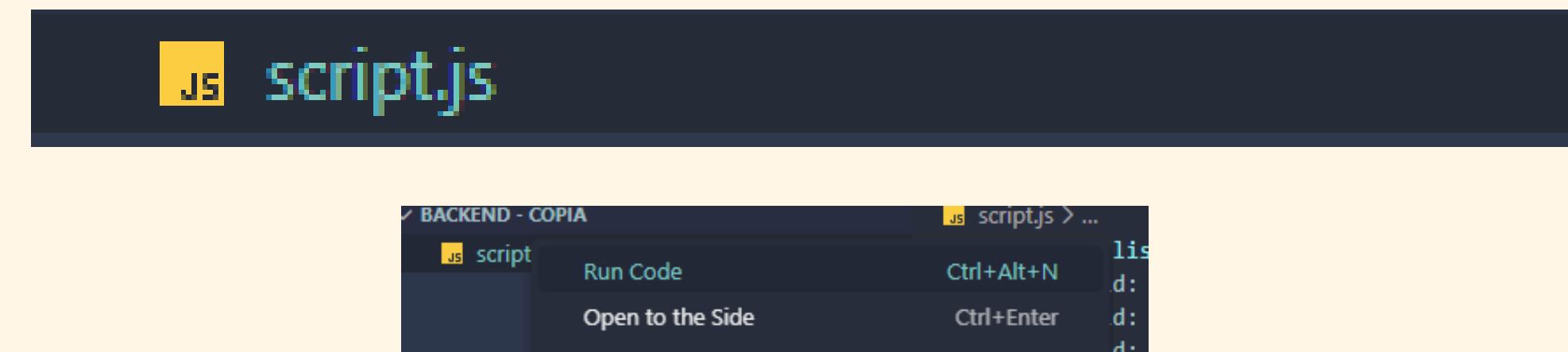
```
JS script.js > ...
1 const listaFuncionarios = [
2   {id: 1, nome: 'Kemmilly Carvalho', setor: 'TI', salario: 7000, bonus: true},
3   {id: 2, nome: 'Isabelly Oliveira', setor: 'TI', salario: 12000, bonus: false},
4   {id: 3, nome: 'Maria Eduarda Dias ', setor: 'TI', salario: 10000, bonus: true},
5   {id: 4, nome: 'Ana Laura Paulino', setor: 'TI', salario: 12000, bonus: false},
6   {id: 5, nome: 'Isabella Rosa', setor: 'TI', salario: 7000, bonus: true},
7 ]
8
9 // ///////////////FILTER
10 const funcionariosBonus = listaFuncionarios.filter(funcionarios => funcionarios.bonus);
11 console.log(funcionariosBonus);
12
```



```
9 // ///////////////FILTER
10 const funcionariosBonus = listaFuncionarios.filter(funcionarios => funcionarios.bonus);
11 console.log(funcionariosBonus);
```

cÓDIGO

Para testar o código basta clicar com o botão direito do mouse no arquivo “script.js” e depois em “Run Code”



Em resposta deverá abrir a mensagem a seguir:

Passo 6



```
[Done] exited with code=0 in 0.358 seconds
[Running] node "c:\Users\Aluno\Desktop\backend - Copia\script.js"
[
  {
    id: 1,
    nome: 'Kemmilly Carvalho',
    setor: 'TI',
    salario: 7000,
    bonus: true
  },
  {
    id: 3,
    nome: 'Maria Eduarda Dias ',
    setor: 'TI',
    salario: 10000,
    bonus: true
  },
  {
    id: 5,
    nome: 'Isabella Rosa',
    setor: 'TI',
    salario: 7000,
    bonus: true
  }
]
```

TESTE

Apenas 3
funcionários
possuem o bônus

MAP

O `.map()` no JavaScript é um método de array que cria uma nova lista aplicando uma função de transformação a cada elemento do array original, sem alterá-lo.

[pesquisa](#)



MAP

Adicione o código para aplicar um bônus de R\$500,00 no salário de cada funcionário que possuem o bônus ativo.

Teste em seguida.

Passo 7



```
13 // //////////////MAP: aplicar bonus de 500
14 const funcionariosComBonus = funcionariosBonus.map(funcionarios => {
15   return {
16     ...funcionarios,
17     salarioFinal: funcionarios.salario + 500
18   }
19 })
20 console.log(funcionariosComBonus)
21
```

cÓDIGO

```
[  
  {  
    id: 1,  
    nome: 'Kemmilly Carvalho',  
    setor: 'TI',  
    salario: 7000,  
    bonus: true,  
    salarioFinal: 7500  
  },  
  {  
    id: 3,  
    nome: 'Maria Eduarda Dias ',  
    setor: 'TI',  
    salario: 10000,  
    bonus: true,  
    salarioFinal: 10500  
  },  
  {  
    id: 5,  
    nome: 'Isabella Rosa',  
    setor: 'TI',  
    salario: 7000,  
    bonus: true,  
    salarioFinal: 7500  
  }  
]
```

Salário + bônus
= salário final

TESTE

REDUCE

O `reduce()` em JavaScript é um método de array utilizado para **reduzir uma lista de elementos a um único valor acumulado** (que pode ser um número, string, objeto ou outro array).

pesquisa

8

REDUCE

Adicione o código para criar uma folha de pagamento com a quantidade de funcionários que ganharam bônus e o total de dinheiro fornecido pela empresa.

Passo 8



```
 22 // /////REDUCE
23 const totalSetor = funcionariosComBonus.reduce((acumulador, item) =>{
24   return acumulador + item.salarioFinal;
25 }, 0);
26 console.log(`R$ ${totalSetor.toFixed(2)})`;
27
28 console.log(`--- FOLHA DE PAGAMENTO ---`);
29 console.log(`Funcionários com bônus de performance : ${funcionariosComBonus.length}`);
30 console.log(`Soma total da folha do setor: R$ ${totalSetor.toFixed(2)})`);
```

CÓDIGO



```
[  
 {  
   id: 1,  
   nome: 'Kemmilly Carvalho',  
   setor: 'TI',  
   salario: 7000,  
   bonus: true,  
   salarioFinal: 7500  
 },  
 {  
   id: 3,  
   nome: 'Maria Eduarda Dias ',  
   setor: 'TI',  
   salario: 10000,  
   bonus: true,  
   salarioFinal: 10500  
 },  
 {  
   id: 5,  
   nome: 'Isabella Rosa',  
   setor: 'TI',  
   salario: 7000,  
   bonus: true,  
   salarioFinal: 7500  
 ]
```

--- FOLHA DE PAGAMENTO ---
Funcionários com bônus de performance : 3
Soma total da folha do setor: R\$ 25500.00

TESTE

OBRIGADA

