

Definição de função como argumento

Se uma função é um valor/expressão então ela também deve herdar o **poder de ser passada como argumento** para outra função.

```
1  const subtrair = (x, y) => x - y
2  const somar = (x, y) => x + y
3  const multiplicar = (x, y) => x*y
4  const dividir = (x, y) => x/y
5  const concatenar = (x, y, sep=" ") => x+sep+y
6  const iniciais = (x, y) => x[0]+y[0]
7  const negativo = subtrair(0,y)
8
9  const exec = (f, x, y) => f(x, y)
10
11 const res1 = exec(subtrair, 50, 25)
12 const res2 = exec(somar, 50, 25)
13 const res3 = exec(multiplicar, 50, 25)
14 const res4 = exec(dividir, 50, 25)
15 const res5 = exec(concatenar, "Isaac", "Newton")
16 const res6 = exec(iniciais, "Isaac", "Newton")
17 const res7 = exec(negativo,30)
18
19 console.log(res1) //25
20 console.log(res2) //75
21 console.log(res3) //1250
22 console.log(res4) //...
23 console.log(res5)
24 console.log(res6)
25 console.log(res7)
```



```
const subtrair = (x, y) => x - y
const somar = (x, y) => x + y
const multiplicar = (x, y) => x*y
const dividir = (x, y) => x/y
const concatenar = (x, y, sep=" ") => x+sep+y
const iniciais = (x, y) => x[0]+y[0]
```

```
const exec = (f, x, y) => f(x, y)
```

Função como argumento

```
const res1 = exec(subtrair, 50, 25)
const res2 = exec(somar, 50, 25)
const res3 = exec(multiplicar, 50, 25)
const res4 = exec(dividir, 50, 25)
const res5 = exec(concatenar, "Isaac", "Newton")
const res6 = exec(iniciais, "Isaac", "Newton")
```