

Variáveis criadas dentro de uma função só podem ser usadas dentro dela justamente porque ela foi criada com aquele escopo, de que, qualquer variável -> var, let e const se for declarado dentro de uma função vai ser o escopo daquela função.

```
function nome(parametros) {  
}
```

Quando invocamos o “return”, a função para de ser executada. Ele vai retornar o valor que vem após ele, você pode retornar um valor ou não e sempre que você retorna significa que a função ela vai rodar e depois ela vai ter um valor que você pode utilizar em outras operações.

```
function nome(parametros) {  
    return;  
}
```

## Função Anônima

Uma variável pode armazenar uma função.

Abrir o VS CODE

Ela não tem um nome, por isso chamamos de função anônima. Ela só está associada a uma variável mas ela não possui um nome.

```
const sum = function (a, b) {  
    return a + b;  
}  
  
sum(1, 42);  
sum(9, 10);
```

## Função Autoinvocável

Uma função anônima entre (), seguida por outro par de (), que representa sua chamada.

```
(  
    function() {  
        let name = "Leonardo";  
        return name;  
    }  
) ();
```

Ela pode ter um nome, mas como ela vai ser executada logo após não é necessário.

Também pode ser usada com parâmetros ou armazenada em uma variável.

```
(  
  function(a, b) {  
    return a + b;  
  }  
) (1, 4);
```

```
const sum = (  
  function() {  
    return a + b;  
  }  
) (1, 5);  
  
console.log(sum);
```

## Callbacks

Uma função passada como argumento para outra função. Utilizando-os você tem maior controle da ordem de chamadas.

```
const calc = function(operacao, num1, num2) {  
  return operacao(num1, num2);  
}  
  
const soma = function(num1, num2) {  
  return num1 + num2;  
}  
  
const sub = function(num1, num2) {  
  return num1 - num2;  
}  
  
const resultSoma = calc(soma, 2, 53);  
const resultSub = calc(sub, 22, 53);  
console.log(resultSub);  
console.log(resultSoma);
```

operacao vai ser uma função, então ela vai executar essa função com os 2 parâmetros num1 e num2 que ela recebeu.

O callback não vai ser executado em ordem de declaração, a gente pode controlar qual a gente quer chamar, quando a gente quer chamar. Dentro da função callback eu podia ter toda uma série de códigos e chamar a operação só num determinado momento.