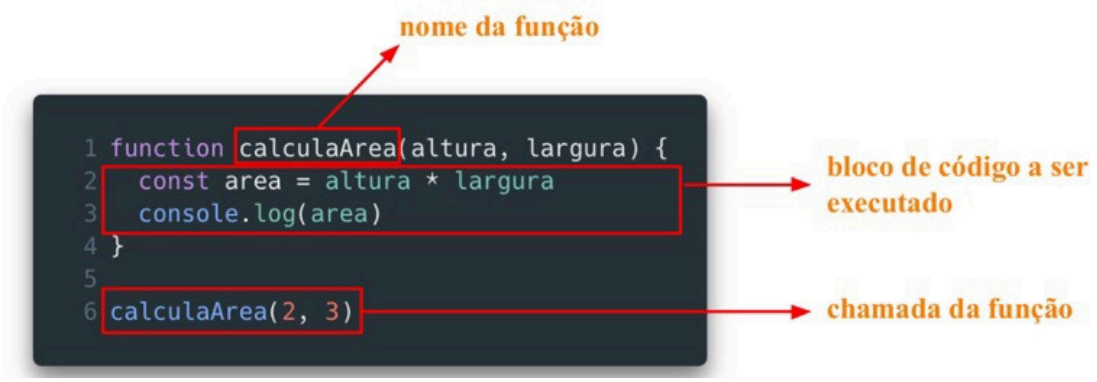
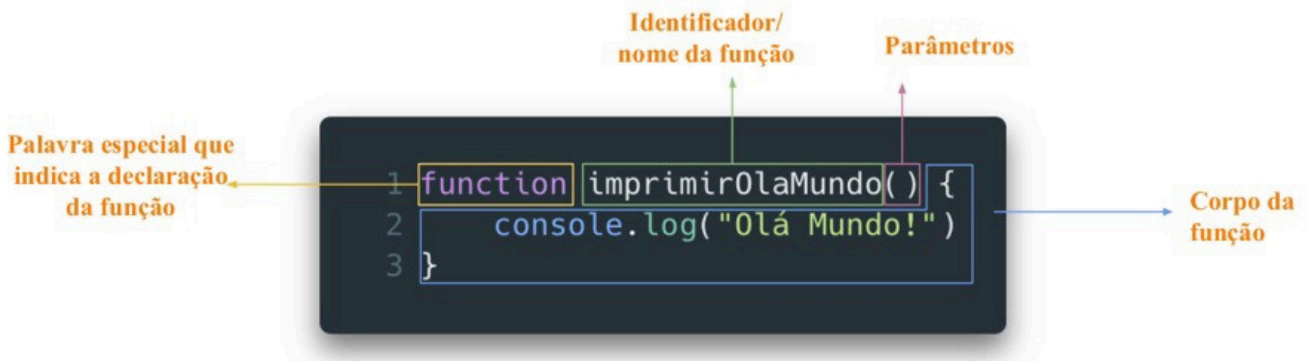
AULA 6 - FUNÇÕES

- Uma função é um **bloco de códigos** o que pode ser chamado (ou invocado) a partir do seu nome. Permite reutilizar variáveis.



- O primeiro passo para criar uma função é **declará-la**
- A declaração **atribui** um **bloco de código** à um **identificador** (ou um nome)



Chamando uma função

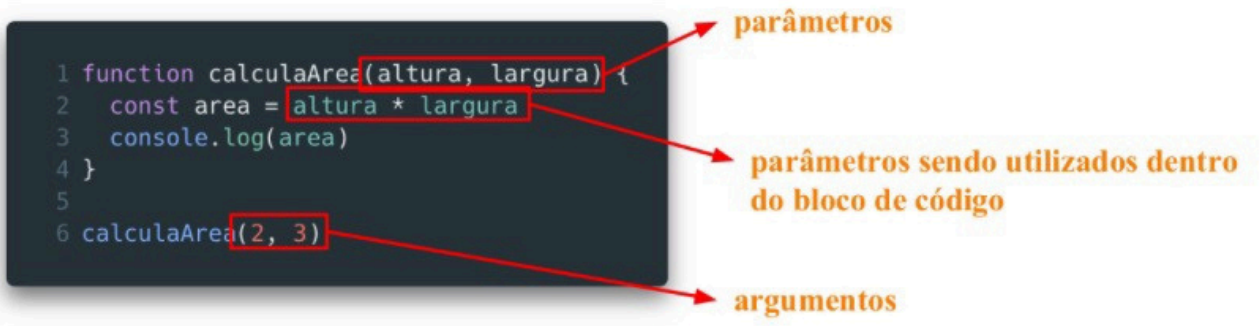
- Podemos chamar, invocar ou executar uma função usando o seu identificador. Quando fazemos isso, o bloco de código definido na declaração é executado.

Declaração vs. Execução

- Só declarar a função **não executa** o código
- Você pode **chamar/invocar** e **executar** a função quantas vezes quiser
- O JavaScript permite executar a função **antes** da sua declaração. Porém, isso deixa o código confuso
- Priorize declarar a função primeiro, e posteriamente executa-lá

Parâmetros e Argumentos

Funções podem receber **entradas**, e se receberem, devem ser usadas no bloco do código dentro da função.

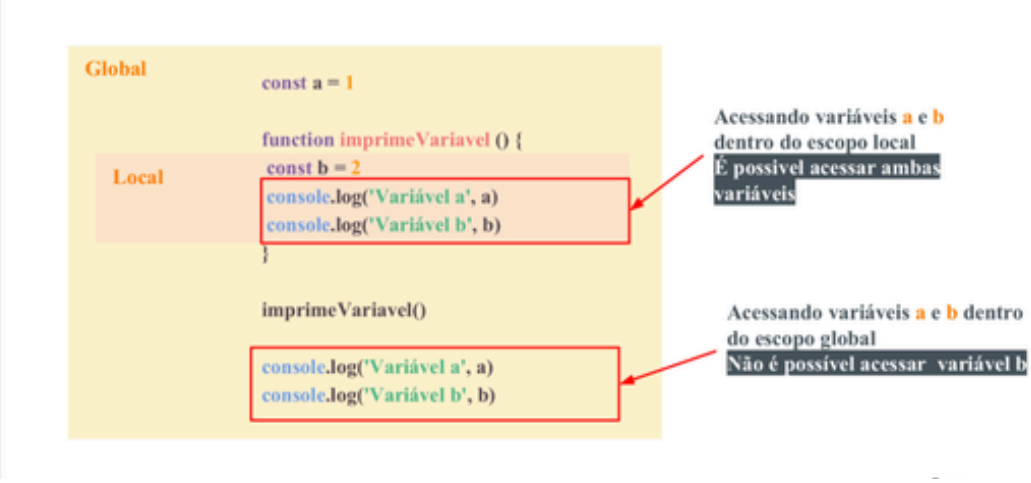


- **Parâmetros** são como **variáveis** criadas na declaração da função, onde podemos guardar os argumentos (valores) a serem enviados para a função.
- **Argumentos** são os **valores** (strings, numbers, booleanos) passados na chamada da função. Cada parâmetro recebe seu valor dos argumentos, seguindo a mesma ordem.

Escopo { }

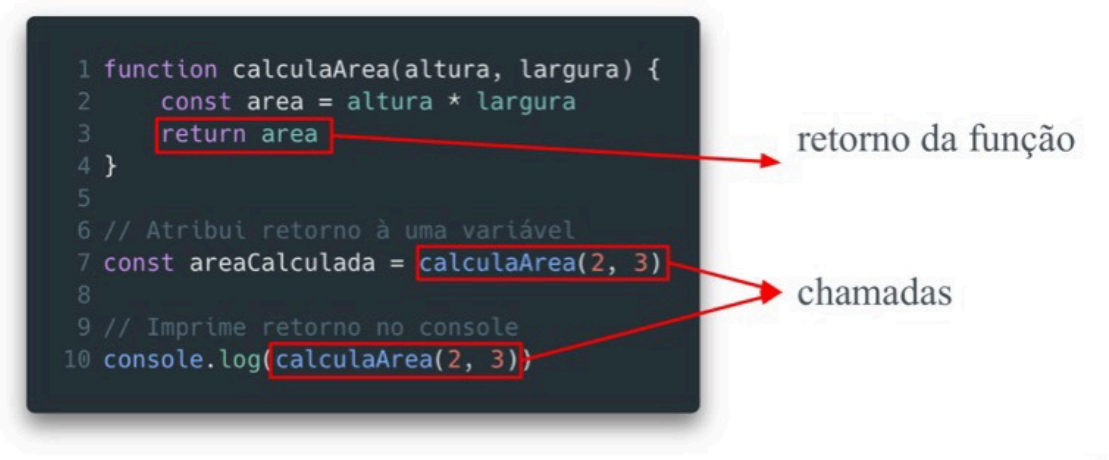
> O escopo determina quais variáveis serão acessíveis ao rodarmos o código.

- No Javascript temos dois tipos de escopo:
- **Escopo Global**: variáveis no escopo global podem ser acessadas de qualquer lugar do código.
- **Escopo Local**: variáveis no escopo local somente podem ser acessadas dentro do escopo em que foram declaradas.
- As variáveis definidas dentro de uma **função** possuem **escopo local**



Retorno

Funções podem gerar **saídas**, que podem ser acessadas após a execução.



- O retorno acontece usando a palavra chave ***return***, seguida pela variável/valor a ser retornado

  Uma função só pode retornar **um valor**

  Quando a função retorna algo, sua **execução é interrompida**
  - Ou seja, o código escrito após o **return** não é executado

Imprimir vs. Retornar

- Quando pede-se para imprimir algo, utilizamos o **console.log()**
- Quando pede-se para retornar algo, utilizamos o **return**

- Funciona como uma caixa preta que pode receber **valores de entrada** (input/parâmetros/argumentos) e pode devolver **valores de saída** (output/resultado)