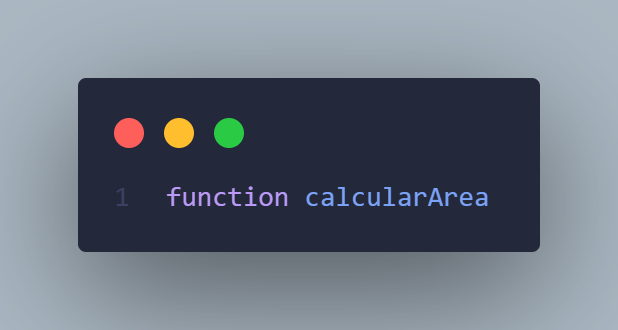
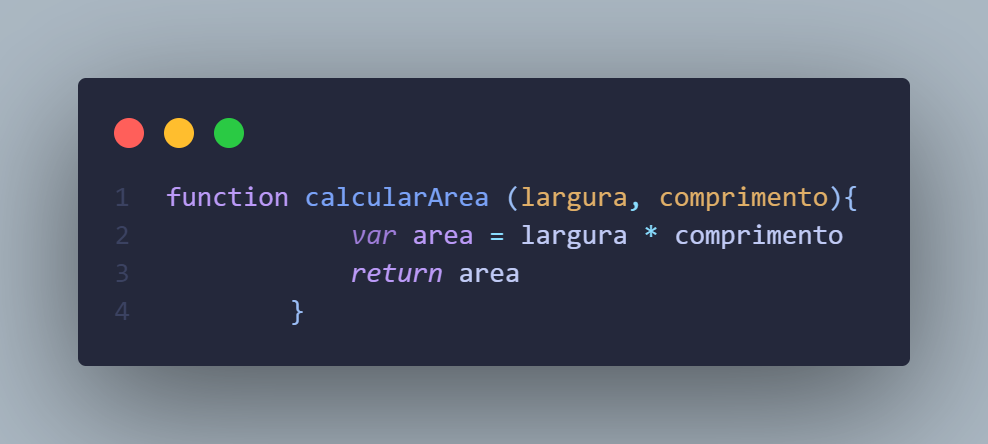
Função

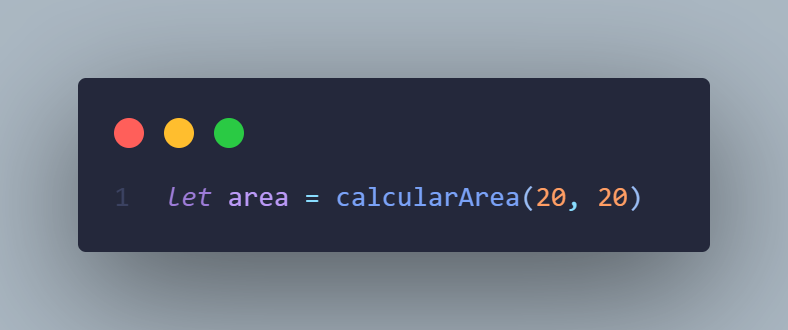
A função são blocos de código responsáveis para executar um conjunto específico de comandos, normalmente são comandos que são reutilizados no código, o nome de uma função é delimitada com o que ele faz



Nesse exemplo o nome atribuído a função é calcularArea, normalmente o nome é formatado por um verbo e um substantivo indicando o que a função faz, logo depois denominamos os parâmetros que ficam entre (), são definidos com os dados que são necessários para fazer a função, nesse caso necessitamos do comprimento e da largura para fazer esse cálculo.



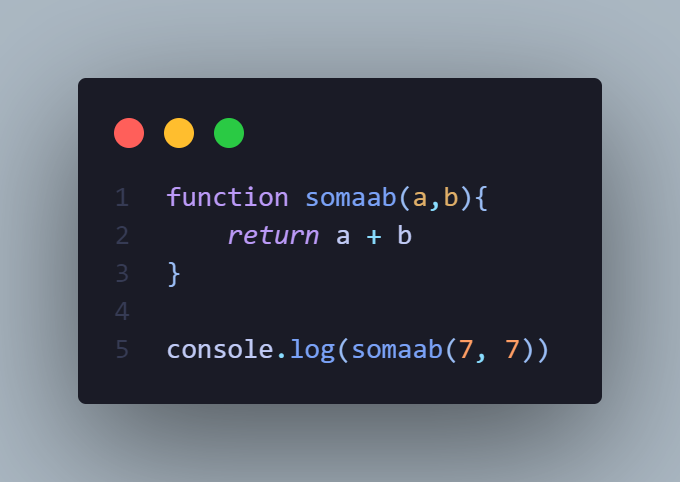
Aqui estamos definido que nosso parâmetros, são largura e comprimento, e dentro da função quando ela for chamada, ela vai definir uma variável e o valor dessa variável é a largura vezes o comprimento, e depois vai retorna o valor dessa variável.



Nesse exemplo acima, criamos uma variável e o valor dessa variável, é o valor do retorno da função, foi defino que os parâmetros são 20, a função vai ser executada fazendo 20 \* 20, e vai retorna esse valor para nossa variável que está executando a função.

**Flexibilidade de parâmetros**

A flexibilidade de parâmetro é a forma que nossa função retorna, com diferentes tipos de situações.



Nesse modelo essa definido que nosso parâmetros dentro do primeiro console.log que a e b é 7, e logo vai somar os dois número, mas o que acontece se tiver mais parâmetros?



Nesse exemplo acima, temos 4 parâmetros, sendo que foi definido só dois, nesse exemplo, a função ignora os parâmetros depois dos dois primeiros, ou seja o resultado ainda vai ser 14, e se deixar só um número? ou deixar sem nenhum parâmetro?

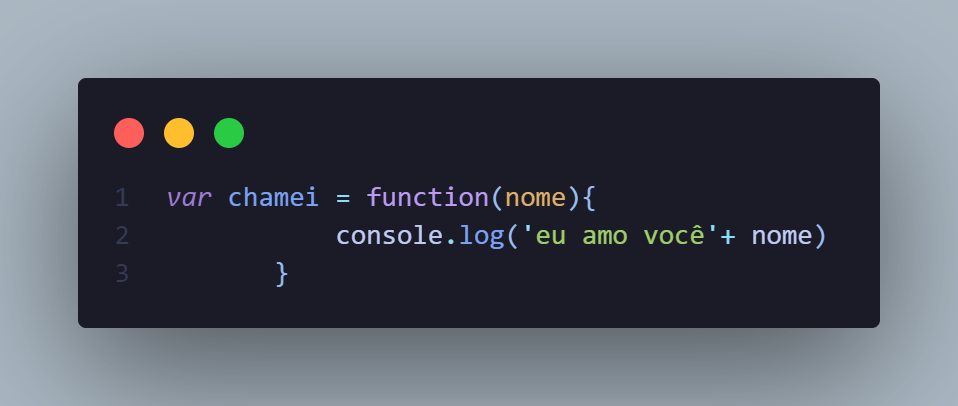


Nesse exemplo acima vemos que temos só o parâmetro 7, então nesse caso esse parâmetro vai ser definido como o valor de a, enquanto b vai ter o valor undefined, e o valor que vai retorna para nós é NaN, Not a Number, ou seja, não é um número, caso não seja definido parâmetro para um ou ambos, o resultado vai ser NaN.

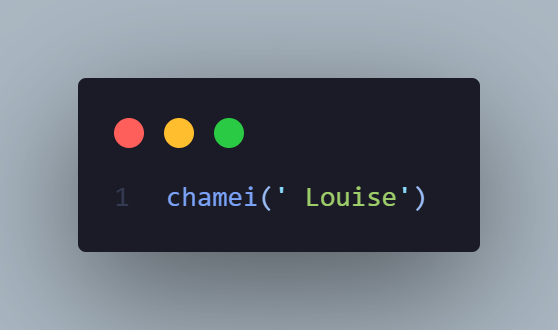
4

Função Anônima e Técnica de wrapper

As funções anônimas, são funções que não possui nome, para chamar essas função precisamos de outros elementos



Nesse exemplo Estamos atribuindo o valor da variável, assim quando chamar a variável tem que passar o valor do parâmetro junto como nesse exemplo:



Isso é um técnica chamada de Wrapper, que é embrulhar, ou seja a função está embrulhada em outro comando

Função de CallBack

Funções de callback são funções que são passadas como argumentos para outras funções, para que possam ser executadas dentro da função que as recebe. Isso permite que o comportamento de uma função seja personalizado ou adiado até que um determinado evento ou ação ocorra:



Nesse exemplo, temos a função artigo, que recebe três parâmetros: id, callbacksucess e callbackerro. As funções callbacksucess e callbackerro são funções de callback, ou seja, são passadas como argumentos para serem executadas dentro da função artigo.

Dentro da função artigo, usamos uma estrutura condicional (if). Se o id for igual a 1, a função callbacksucess é chamada com os parâmetros "Funções" e "Aqui vamos ver as funções de callback", representando o título e a descrição. Se o id for diferente de 1 (caso contrário), a função callbackerro é chamada com o parâmetro "Erro ao recuperar dados", para tratar possíveis erros.

Depois, definimos as variáveis callbacksucess e callbackerro, e a cada uma delas atribuímos funções anônimas:

* A função atribuída a callbacksucess adiciona um título (<h1>) ao corpo da página, seguido de uma linha horizontal (<hr>) e um parágrafo (<p>) com a descrição fornecida.
* A função atribuída a callbackerro exibe um alerta (alert) com a mensagem de erro.

Na última linha, chamamos a função artigo com o valor 1 para o parâmetro id, e passamos as funções callbacksucess e callbackerro como argumentos, para que sejam executadas conforme a lógica definida na função artigo.