



## Módulo 2.4 - Funções

5000



### Funções

- Função é um bloco de código que só vai ser executado quando chamado.
- Uma função também pode retornar dados.
- Para executar uma função, coloque o nome (da função) seguido de parênteses.

### Exemplo

```
def funcao():  
    print('dentro da função')  
  
funcao() # <- Executando a função  
  
# Retorno  
> dentro da função
```

### Parâmetros

- Uma função pode receber múltiplos argumentos nos parâmetros dela ou não receber nenhum.
- Os argumentos são passados nos parênteses, que estão logo após o nome da função.
- Você também pode passar alguns parâmetros padrão para sua função.

### Exemplo

Utilizando apenas um argumento.



## Módulo 2.4 - Funções

5000



### Funções

- Função é um bloco de código que só vai ser executado quando chamado.
- Uma função também pode retornar dados.
- Para executar uma função, coloque o nome (da função) seguido de parênteses.

### Exemplo

```
def funcao():  
    print('dentro da função')  
  
funcao() # <- Executando a função  
  
# Retorno  
> dentro da função
```

### Parâmetros

- Uma função pode receber múltiplos argumentos nos parâmetros dela ou não receber nenhum.
- Os argumentos são passados nos parênteses, que estão logo após o nome da função.
- Você também pode passar alguns parâmetros padrão para sua função.

### Exemplo

Utilizando apenas um argumento.



## Módulo 2.4 - Funções

5000



### Funções

- Função é um bloco de código que só vai ser executado quando chamado.
- Uma função também pode retornar dados.
- Para executar uma função, coloque o nome (da função) seguido de parênteses.

### Exemplo

```
def funcao():  
    print('dentro da função')  
  
funcao() # <- Executando a função  
  
# Retorno  
> dentro da função
```

### Parâmetros

- Uma função pode receber múltiplos argumentos nos parâmetros dela ou não receber nenhum.
- Os argumentos são passados nos parênteses, que estão logo após o nome da função.
- Você também pode passar alguns parâmetros padrão para sua função.

### Exemplo

Utilizando apenas um argumento.



## Módulo 2.4 - Funções

5000



### Funções

- Função é um bloco de código que só vai ser executado quando chamado.
- Uma função também pode retornar dados.
- Para executar uma função, coloque o nome (da função) seguido de parênteses.

### Exemplo

```
def funcao():  
    print('dentro da função')  
  
funcao() # <- Executando a função  
  
# Retorno  
> dentro da função
```

### Parâmetros

- Uma função pode receber múltiplos argumentos nos parâmetros dela ou não receber nenhum.
- Os argumentos são passados nos parênteses, que estão logo após o nome da função.
- Você também pode passar alguns parâmetros padrão para sua função.

### Exemplo

Utilizando apenas um argumento.



## Módulo 2.4 - Funções

5000



### Funções

- Função é um bloco de código que só vai ser executado quando chamado.
- Uma função também pode retornar dados.
- Para executar uma função, coloque o nome (da função) seguido de parênteses.

### Exemplo

```
def funcao():  
    print('dentro da função')  
  
funcao() # <- Executando a função  
  
# Retorno  
> dentro da função
```

### Parâmetros

- Uma função pode receber múltiplos argumentos nos parâmetros dela ou não receber nenhum.
- Os argumentos são passados nos parênteses, que estão logo após o nome da função.
- Você também pode passar alguns parâmetros padrão para sua função.

### Exemplo

Utilizando apenas um argumento.



## Módulo 2.4 - Funções

5000



### Funções

- Função é um bloco de código que só vai ser executado quando chamado.
- Uma função também pode retornar dados.
- Para executar uma função, coloque o nome (da função) seguido de parênteses.

### Exemplo

```
def funcao():  
    print('dentro da função')  
  
funcao() # <- Executando a função  
  
# Retorno  
> dentro da função
```

### Parâmetros

- Uma função pode receber múltiplos argumentos nos parâmetros dela ou não receber nenhum.
- Os argumentos são passados nos parênteses, que estão logo após o nome da função.
- Você também pode passar alguns parâmetros padrão para sua função.

### Exemplo

Utilizando apenas um argumento.