



Anterior  
Desafio

Próximo

Conteúdo do Livro



## Infográfico



Praticamente, todas as linguagens de programação utilizadas atualmente têm a possibilidade de trabalhar com funções. Entender o conceito de função é fundamental, inclusive, para avançar para a programação orientada a objetos.



No Infográfico, a seguir, conheça as funções do Python.





Anterior  
**Desafio**

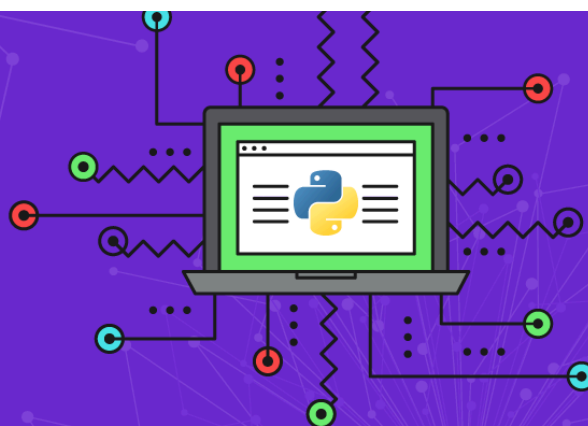
Próximo

**Conteúdo do Livro**



Seu principal objetivo é permitir organizar os programas em sub-rotinas que correspondam ao todo do sistema, mas que executem uma parte em específico. São como engrenagens que, acopladas, fazem o todo funcionar.

Em Python, para trabalhar com funções, é importante conhecer alguns conceitos relacionados a isso.



## Funções no Python

Em Python, subprogramas têm o nome de funções.

Onde:

- **Nome** é o nome da função.
- **args** são especificações de argumentos da função.
- Uma função pode ter 0, 1 ou mais argumentos.
- **Comando** contém as instruções a serem executadas quando a função é invocada.

### Formato geral:

```
def nome (arg, arg, ... arg):  
    comando  
    ...  
    comando
```

## Variáveis locais e globais

### Variáveis locais

Variáveis utilizadas somente nas funções em que foram definidas.

### Variáveis globais

Variáveis definidas fora de funções.

É possível, no código de uma função, ler o conteúdo de uma variável global. Para alterar uma variável global, ela precisa ser declarada no corpo da função.



Anterior  
**Desafio**

Próximo

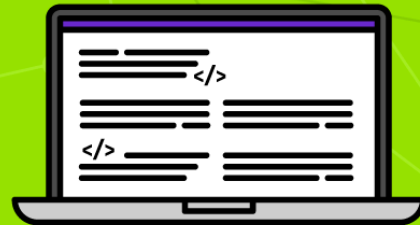
**Conteúdo do Livro**



iniciais do chamador. Essas variáveis, assim como outras definidas dentro da função, são ditas locais, isto é, só existem no lugar onde foram definidas.

Ao retornar ao ponto de chamada, as variáveis locais são descartadas. Se uma função define  $n$  argumentos, valores para todos eles devem ser passados pelo chamado.

**Exceção: argumentos com valores default.**



## Funções vars()

O dicionário obtido com a função vars() pode ser usado para ter acesso a todas as variáveis definidas em um escopo.

**Exemplo:**

```
vars()
{'__builtins__': <module '__builtin__' (built-in)>,
 '__name__': '__main__', '__doc__': None}
def f():
    x = 1
    print vars()
vars()
{'f': <function f at 0xb6e7f56c>, '__builtins__':
<module '__builtin__' (built-in)>, '__name__':
 '__main__', '__doc__': None}
f()
{'x': 1}
```

