

Em Python, parâmetros e argumentos são conceitos relacionados, mas não são a mesma coisa. Aqui está uma explicação de cada um e como eles se relacionam:

Parâmetros

Definição: Parâmetros são variáveis listadas na definição de uma função. Eles agem como placeholders que recebem valores quando a função é chamada.

Exemplo:

python

Copiar código

```
def soma(a, b): # a e b são parâmetros
```

```
    return a + b
```

Aqui, a e b são parâmetros da função soma.

Argumentos

Definição: Argumentos são os valores reais que você passa para uma função quando a chama. Eles são atribuídos aos parâmetros da função.

Exemplo:

python

Copiar código

```
resultado = soma(3, 5) # 3 e 5 são argumentos
```

Aqui, 3 e 5 são argumentos passados para a função soma, que então atribui esses valores aos parâmetros a e b, respectivamente.

Resumo da Diferença

Parâmetros são usados na definição da função.

Argumentos são usados quando você chama a função.

Exemplos Adicionais

Função com Parâmetros e Argumentos

python

Copiar código

```
def saudacao(nome, idade): # nome e idade são parâmetros  
    return f"Olá, {nome}! Você tem {idade} anos."
```

```
mensagem = saudacao("Ana", 30) # "Ana" e 30 são argumentos  
print(mensagem) # Saída: Olá, Ana! Você tem 30 anos.
```

Parâmetros Padrão

Você pode definir valores padrão para parâmetros, que são usados se nenhum argumento correspondente for fornecido:

python

Copiar código

```
def saudacao(nome="Visitante", idade=0): # Valores padrão para os parâmetros  
    return f"Olá, {nome}! Você tem {idade} anos."
```

```
# Chamando a função sem fornecer argumentos
```

```
print(saudacao()) # Saída: Olá, Visitante! Você tem 0 anos.
```

```
# Chamando a função com argumentos
```

```
print(saudacao("Carlos", 25)) # Saída: Olá, Carlos! Você tem 25 anos.
```

Argumentos Nomeados

Você pode passar argumentos pelo nome para tornar o código mais legível e evitar erros:

python

Copiar código

```
print(saudacao(nome="João", idade=45)) # Argumentos nomeados
```

Argumentos Variáveis

Você pode usar `*args` e `**kwargs` para passar um número variável de argumentos:

python

Copiar código

```
def soma(*args): # *args captura argumentos posicionais variáveis
    return sum(args)
```

```
print(soma(1, 2, 3, 4)) # Saída: 10
```

```
def info_pessoa(**kwargs): # **kwargs captura argumentos nomeados variáveis
    for chave, valor in kwargs.items():
        print(f"{chave}: {valor}")
```

```
info_pessoa(nome="Maria", idade=28, cidade="São Paulo")
```

Essas ferramentas oferecem flexibilidade ao definir e chamar funções em Python.