

# Operacoes Basicas em Python

## 1 Operacoes matematicas

### 1.1 Adicao, subtracao, multiplicacao, divisao

Python suporta as operacoes aritmeticas basicas:

```
1 a = 10
2 b = 3
3
4 print(a + b) # Adicao: 13
5 print(a - b) # Subtracao: 7
6 print(a * b) # Multiplicacao: 30
7 print(a / b) # Divisao: 3.3333333333333335
```

### 1.2 Divisao inteira e modulo

A divisao inteira retorna o quociente sem a parte fracionaria, enquanto o modulo retorna o resto da divisao:

```
1 print(a // b) # Divisao inteira: 3
2 print(a % b)  # Modulo: 1
```

### 1.3 Potenciacao e raiz quadrada

Para potenciacao, usamos o operador \*\*. Para raiz quadrada, podemos usar a potencia de 0.5 ou a funcao sqrt() do modulo math:

```
1 import math
2
3 print(a ** 2)          # Potenciacao: 100
4 print(math.sqrt(a))    # Raiz quadrada: 3.1622776601683795
```

## 1.4 Precedencia de operadores

A ordem de precedencia em Python segue a ordem matematica padrao: parenteses, exponenciacao, multiplicacao/divisao, adicao/subtracao.

```
1 print(2 + 3 * 4)      # 14
2 print((2 + 3) * 4)    # 20
```

## 2 Operacoes com strings

### 2.1 Concatenacao

Strings podem ser concatenadas usando o operador +:

```
1 str1 = "Hello"
2 str2 = "World"
3 print(str1 + " " + str2) # Hello World
```

### 2.2 Repeticao

Strings podem ser repetidas usando o operador \*:

```
1 print("Ha" * 3) # HaHaHa
```

### 2.3 Indexacao e fatiamento basicos

Podemos acessar caracteres individuais ou fatias de uma string:

```
1 s = "Python"
2 print(s[0])      # P
3 print(s[1:4])    # yth
4 print(s[::-1])   # nohtyP (inverte a string)
```

## 3 Operacoes de comparacao

### 3.1 Igualdade, desigualdade

```
1 print(5 == 5)    # True
2 print(5 != 6)    # True
```

### 3.2 Maior que, menor que, maior ou igual, menor ou igual

```
1 print(5 > 3)      # True
2 print(5 < 3)      # False
3 print(5 >= 5)     # True
4 print(5 <= 4)     # False
```

### 3.3 Comparacoes com diferentes tipos de dados

Python permite comparacoes entre diferentes tipos de dados em alguns casos:

```
1 print(5 == 5.0)   # True
2 print("5" == 5)   # False
3 print(True == 1)  # True
4 print(False == 0) # True
```

## 4 Operacoes logicas

### 4.1 AND, OR, NOT

```
1 print(True and False) # False
2 print(True or False)  # True
3 print(not True)       # False
```

### 4.2 Uso com booleanos e outros tipos de dados

Operacoes logicas podem ser usadas com outros tipos de dados:

```
1 print(5 and 3)      # 3
2 print(0 or 2)       # 2
3 print(not [])       # True
```

## 5 Conversao entre tipos de dados

### 5.1 Funcoes int(), float(), str(), bool()

Python fornece funcoes para converter entre tipos de dados:

```

1 print(int("5"))      # 5
2 print(float("3.14")) # 3.14
3 print(str(42))       # "42"
4 print(bool(1))       # True

```

## 5.2 Casos de uso e possiveis erros

Cuidado com conversoes que podem gerar erros:

```

1 try:
2     print(int("3.14")) # Erro: nao pode converter float
3         string para int
4 except ValueError as e:
5     print(f"Erro: {e}")
6
7 print(int(3.14)) # 3 (trunca o float)
8 print(bool("")) # False (string vazia e considerada False)

```

## 6 Conclusao e exercicios praticos

### 6.1 Recapitulacao das operacoes aprendidas

Nesta aula, cobrimos:

- Operacoes matematicas basicas e avancadas
- Manipulacao de strings
- Operacoes de comparacao
- Operacoes logicas
- Conversao entre tipos de dados

### 6.2 Sugestao de exercicios para praticar

- 1. Calcule a area de um circulo dado o raio.
- 2. Crie uma funcao que verifica se uma palavra e um palindromo.
- 3. Implemente um conversor de temperatura de Celsius para Fahrenheit.
- 4. Crie um programa que determine se um ano e bissexto.
- 5. Implemente uma calculadora simples que aceita dois numeros e uma operacao.

Pratique estes exercicios para solidificar seu entendimento das operacoes basicas em Python!