

# Operadores

Introdução a Programação

# Objetivo de Aprendizagem

- Conhecer operadores de uso geral do Python

# Agenda

- Módulos
- `math`
- Operadores
- Formatação de números
- Funções de texto padrão

# Módulos Externos

Um módulo Python é um arquivo contendo um conjunto de funções e declarações que pode ser incluído em um *script*. Um módulo pode ser entendido como uma biblioteca de códigos. O Python disponibiliza diversos módulos embutidos e existem ainda módulos de terceiros.

# Módulos Típicos

- Existem diversos módulos disponíveis
- Exemplos:
  - `math`
  - `platform`
  - `os`
  - Dentre muitos outros

# Como usar um módulo?

- Vamos utilizar como exemplo o módulo `math`
- O primeiro passo é a importação do módulo: `import math`

```
1 import math
2 print(math.pi)
```

# O que acontece ao executar os códigos?

```
1 import math
2 print(math.ceil(1.45))
3 print(math.floor(1.45))
4 print(math.factorial(3))
5 print(math.fabs(-1))
6 print(math.pow(2,3))
7 print(math.sqrt(2))
```



# Lista completa do Módulo Math

Mathematical functions

---

# Outras funções matemáticas

```
1 # retorna o valor absoluto de um número
2 abs(-3)
3 # retorna o valor máximo de uma sequência
4 max(1, 2, 3)
5 # retorna o valor mínimo de uma sequência
6 min(1, 2, 3)
7 # arredonda um float para o inteiro mais próximo
8 round(3.14)
9 # arredonda um float para um determinado número de casas
10 round(3.14, 1)
11 # retorna a soma de uma sequência de números
12 sum([1, 2, 3])
```

# Exemplo

```
1 top = max(5, 10, 25)
2 print(f"Máximo: {top}\nMínimo: {min(5, 10, 25)}")
```

# Operadores

# Operadores Aritméticos

Operador	Nome	Função
+	Adição	Realiza a soma de ambos operandos.
-	Subtração	Realiza a subtração de ambos operandos.
*	Multiplicação	Realiza a multiplicação de ambos operandos.
/	Divisão	Realiza a Divisão de ambos operandos.
//	Divisão inteira	Realiza a divisão entre operandos e a parte decimal de ambos operandos.
%	Módulo	Retorna o resto da divisão de ambos operandos.
**	Exponenciação	Retorna o resultado da elevação da potência pelo outro.

# Operações Aritméticas

```
1 a = 5
2 b = 3
3 c = a + b # soma de dois números
4 d = a - b # subtração de dois números
5 e = a * b # multiplicação de dois números
6 f = a / b # divisão de dois números
7 g = a // b # divisão inteira de dois números
8 g = a % b # módulo de dois números
9 h = a ** b # exponenciação de dois números
```

# Operadores de Comparação

Operador	Nome	Função
==	Igual a	Verifica se um valor é igual ao outro
!=	Diferente de	Verifica se um valor é diferente ao outro
>	Maior que	Verifica se um valor é maior que outro
>=	Maior ou igual	Verifica se um valor é maior ou igual ao outro
<	Menor que	Verifica se um valor é menor que outro
<=	Menor ou igual	Verifica se um valor é menor ou igual ao outro
**	Exponenciação	Retorna o resultado da elevação da potência pelo outro.

# Operações de Comparação

```
1  var = 5
2  if var == 5:
3      print('Os valores são iguais')
4  if var != 7:
5      print('O valor não é igual a 7')
6  if var > 2:
7      print('O valor da variável é maior de 2')
8  if var ≥ 5:
9      print('O valor da variável é maior ou igual a 5')
10 if var < 7:
11     print('O valor da variável é menor que 7')
12 if var ≤ 5:
13     print('O valor da variável é menor ou igual a 5')
```



# Operadores de Atribuição

Operador	Nome	Função
=	Atribuição	$x = 1$
+=	Incremento	$x = x + 1$
-=	Decremento	$x = x - 1$
*=		$x = x * 1$
/=		$x = x / 1$
%=		$x = x \% 1$

# Operações de Atribuição

```
1  n = 10
2  n += 1
3  print(n)
4  n += 5
5  print(n)
6  n -= 3
7  print(n)
8  n *= 2
9  print(n)
10 n /= 2
11 print(n)
12 n %= 12
13 print(n)
```

# Formatação

# Formatação de Números

## Casas decimais

```
1 import math
2 pi = math.pi
3 print(f"Pi: {pi}")
4 print(f"{pi:.1f}")
5 print(f"{pi:.3f}")
6 print(f"{pi:09.5f}")
7 print("{:.2f}".format(pi))
```

# Formatação de Números

## Percentuais

```
1 p = 0.215
2 print(f"Seu desconto é de {p:.2%}")
```

# Formatação de Strings

## Maiúsculas e Minúsculas

```
1 sigla = "mmc"
2 significado = "mínimo múltiplo comum"
3 print("{} = {}".format(sigla.upper(), significado.title()))
```

```
1 cmd1 = "PRINT"
2 cmd2 = "INPUT"
3 print("Alguns exemplos de comandos do Python: {} e {}".fo
```

# Formatação de Strings

Contagem de ocorrências

```
1 texto = "I love apples. Apples are my favorite fruits"
2 x = texto.count("apple")
3 print(x)
```

# Formatação de Strings

Verificação de dígito ou número

```
1  sn = "50800"
2  model = "A320"
3  sigla = "MMC"
4  x = sn.isdigit()
5  print(x)
6  x = model.isdigit()
7  print(x)
8  x = model.isalnum()
9  print(x)
10 x = sigla.isalpha()
11 print(x)
```



# Validação de Entrada

## Exemplo

```
1  usuario = input("Digite o nome de usuário (apenas letras)
2
3  if not usuario.isalpha():
4      print("Nome de usuário inválido. Use apenas letras (s
5  else:
6      senha= input("Digite a senha (apenas números): ")
7      if not senha.isdigit():
8          print("Senha inválida. Use apenas números (sem le
9  print("Usuário e senha válidos!")
```

# Comprimento de Strings

`len()`

- A função `len()` retorna o número de itens de um objeto
- Quando o objeto é uma string, retorna a quantidade de caracteres

```
1 fruits = ["apple", "banana", "cherry"]  
2 x = len(fruits)
```

```
1 msg = "Hello, world!!!"  
2 x = len(msg)
```

## `len()`

- Apenas faz sentido aplicar a função para tipos de dados iteráveis
  - Listas
  - Dicionários
  - Conjuntos
  - Strings
- Para tipos de dados que armazenam apenas um item `len()` apresenta erro

# Exemplo

```
1 nome = input("Digite seu nome: ")
2
3 print(f"\nNome em maiúsculas: {nome.upper()}")
4
5 tamanho = len(nome.strip())
6 print(f"Seu nome tem {tamanho} letras!")
7
8 if tamanho ≤ 4:
9     print("Seu nome é curto!")
10 elif tamanho ≤ 6:
11     print("Seu nome tem um tamanho médio.")
12 else:
13     print("Seu nome é longo!")
```

# Exercícios

1

Crie um *script* em Python uma Calculadora Geométrica para realizar cálculos de área e perímetro de um círculo ou retângulo, segundo opção do usuário. Caso escolha o círculo o usuário deve informar o raio, mas caso escolha o retângulo deve informar a base e altura do mesmo.

## 2

Crie um *script* em Python que realiza a criação de usuário para um sistema de banco de dados. Tanto o login quanto a senha possuem regras específicas que devem ser atendidas. O login do usuário deve possuir no mínimo 3 caracteres e no máximo 8, deve conter apenas caracteres do alfabeto e não possuir espaços em branco nem no início, nem no final. Já a senha deve ter pelo menos 6 caracteres e possuir caracteres do alfabeto ou numéricos. Mensagens de sucesso e erro devem ser exibidas ao usuário.


# Referências

- W3 Schools
- Python Math



# José Roberto Bezerra

[jbroberto@ifce.edu.br](mailto:jbroberto@ifce.edu.br)

Powered by  Slidify