



Computational Thinking With Python

Prof. Gilberto Alexandre das Neves
profgilberto.neves@fiap.com.br

Módulos

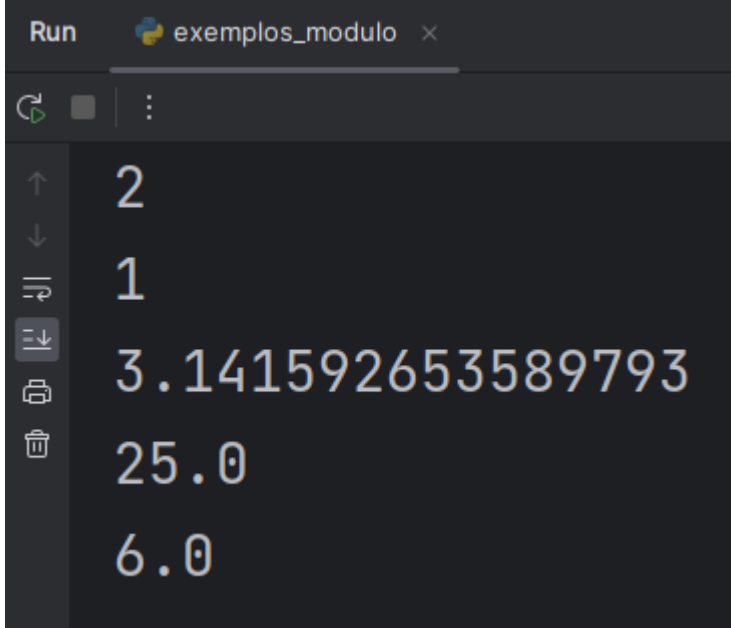
O Python possui o módulo **math** que contém várias funções matemáticas para uso.

Para usar estas funções, primeiramente temos que importá-las no início o código, usando **import**.

Veja algumas das funções do módulo **math**:

- **math.ceil(x)** – arredonda o número x para cima.
- **math.floor(x)** – arredonda o número x para baixo.
- **math.pi** – retorna o valor de π .
- **math.pow(x, y)** – eleva o valor x por y .
- **math.sqrt(x)** – calcula a raiz quadrada de x .

```
1 import math
2 numero = 1.4
3 print(math.ceil(numero))
4 print(math.floor(numero))
5 print(math.pi)
6 print(math.pow(5, 2))
7 print(math.sqrt(36))
8
```



The screenshot shows a Python IDE window titled "exemplos_modulo". The output of the code is displayed in a list on the right side of the window. The output values are: 2, 1, 3.141592653589793, 25.0, and 6.0. The output is displayed in a light blue font on a dark background.

```
Run exemplos_modulo x
```

```
2
1
3.141592653589793
25.0
6.0
```

O módulo **random** disponibiliza funções para obtenção de valores aleatórios.

Para usar o módulo **random** , primeiramente temos que importa-la no início do código, usando **import**.

Veja algumas das funções do módulo **random**:

- **random.randint(x, y)** – sorteia um número aleatório entre x e y .
- **random.choice(*lista*)** – sorteia um valor aleatório da *lista*.

```
1 import random
2 frutas = ['maçã', 'kiwi', 'pêra', 'amora']
3 # sorteando um número entre 10-50
4 sorteia_numero = random.randint(10, 50)
5 print("Número sorteado: {}".format(sorteia_numero))
6 # sorteando uma fruta da lista
7 sorteia_fruta = random.choice(frutas)
8 print("Fruta sorteada: {}".format(sorteia_fruta))
```

Além de usar os módulos do Python, podemos criar e usar nossos próprios módulos.

Vamos a uma exemplo: crie um programa (chamado **area.py**) com funções com passagem de parâmetros e retorno de valores (resultados) capazes de calcular as áreas de um retângulo, triângulo e círculo.

Vamos criar agora um outro programa que exiba um menu perguntado ao usuário qual área ele deseja calcular, pedir as informações necessária e utilizar a função de nosso programa anterior.

```
1  # função cálculo da área do retângulo
2  def area_retangulo(lado, altura):
3      return lado * altura
4
5
6  # função cálculo da área do triângulo
7  def area_triangulo(lado, altura):
8      return lado * altura / 2
9
10
11 # função cálculo da área do círculo
12 def area_circulo(raio):
13     import math
14     return math.pi * math.pow(raio, 2)
```



```
1 import area
2 print("Escolha qual área deseja calcular")
3 print("(1) Retângulo")
4 print("(2) Triângulo")
5 print("(3) Círculo")
6 escolha = int(input("Número da escolha: "))
7 if escolha == 1:
8     lado = float(input("Digite valor de lado: "))
9     altura = float(input("Digite valor da altura: "))
10    resultado = area.area_retangulo(lado, altura)
11    print("Área do retângulo: {:.2f}".format(resultado))
12 elif escolha == 2:
13     lado = float(input("Digite valor de lado: "))
14     altura = float(input("Digite valor da altura: "))
15     resultado = area.area_triangulo(lado, altura)
16     print("Área do triângulo: {:.2f}".format(resultado))
17 elif escolha == 3:
18     raio = float(input("Digite valor do raio: "))
19     resultado = area.area_circulo(raio)
20     print("Área do círculo: {:.2f}".format(resultado))
21 else:
22     print("Opção de escolha inválida")
```

Exercício

Monte um programa do jogo **Jokenpô** (pedra, papel e tesoura). Neste jogo, o usuário jogará contra o computador.



- Sorteio aleatório da jogada o computador:
 - Armazene as opções em uma Lista
 - Crie um programa (para importar como um módulo) com uma função que sorteie dessa lista uma jogada para o computador e retorne o que foi sorteado.
- No programa principal, peça para o usuário escolher seu movimento.
- Faça a verificação de quem ganhou e exiba uma mensagem e as jogadas de cada participante.
 - Criar aqui uma função (no arquivo importado anterior) que faça essa verificação e retorne o resultado de quem venceu.

- Coloque tudo isso em uma estrutura de repetição condicional, permitindo que o usuário continue a jogar
 - Pergunte se ele deseja continuar jogando
 - Em caso afirmativo repita novamente o sorteio do computador, a escolha de nova jogada do usuário e a verificação de quem venceu.
 - Em caso negativo para continuar jogando, encerre o programa.
- Exiba ao final de tudo (quando o usuário não quer mais jogar) um placar final com a pontuação do computador (total de vezes que ele venceu) e a pontuação do usuário (total de vezes que ele venceu).





Introdução à programação com Python. Nilo Menezes. Novatec, 2019.

Curso Intensivo de Python: Uma introdução prática e baseada em projetos à programação. Eric Matthes. Novatec, 2016.

Até breve!