



UNIVERSIDADE
FEDERAL DO PIAUÍ

AULA 04: BIBLIOTECA EM PYTHON

Prof. Dr. Jaclason Veras

jaclason@ufpi.edu.br

INTRODUÇÃO



UNIVERSIDADE
FEDERAL DO PIAUÍ

- As bibliotecas no Python são conjuntos de código pré-escrito que permite realizar diversas tarefas sem a necessidade de escrever o código do zero.
- São componentes essenciais da linguagem de programação Python, pois fornecem funcionalidades adicionais que expandem as capacidades da linguagem base.

INTRODUÇÃO



UNIVERSIDADE
FEDERAL DO PIAUÍ

- As bibliotecas desempenham um papel fundamental no desenvolvimento de projetos em Python. Permitem agregar funcionalidades específicas aos nossos programas e nos ajudam a resolver problemas complexos de forma mais eficiente.

INTRODUÇÃO



UNIVERSIDADE
FEDERAL DO PIAUÍ

- Ao utilizar bibliotecas no desenvolvimento de projetos em Python, podemos:
 - **Acelerar o desenvolvimento:** Ao invés de reinventar a roda e escrever código do zero, podemos importar uma biblioteca que já possui a funcionalidade desejada. Isso nos permite economizar tempo e esforço, acelerando o processo de desenvolvimento.

INTRODUÇÃO



UNIVERSIDADE
FEDERAL DO PIAUÍ

- Ao utilizar bibliotecas no desenvolvimento de projetos em Python, podemos:
 - **Aproveitar o conhecimento coletivo:** Muitas bibliotecas são desenvolvidas e mantidas por programadores experientes e especializados em determinadas áreas. Ao utilizarmos essas bibliotecas, podemos aproveitar o conhecimento e experiência desses desenvolvedores, melhorando a qualidade e eficiência dos nossos programas.

INTRODUÇÃO



- Ao utilizar bibliotecas no desenvolvimento de projetos em Python, podemos:
 - **Expandir as capacidades da linguagem:** O Python é uma linguagem de programação extremamente versátil, mas não possui todas as funcionalidades embutidas. As bibliotecas nos permitem expandir as capacidades da linguagem, adicionando recursos que não estão disponíveis nativamente.

INTRODUÇÃO



UNIVERSIDADE
FEDERAL DO PIAUÍ

- Ao utilizar bibliotecas no desenvolvimento de projetos em Python, podemos:
 - **Resolver problemas complexos:** Muitas vezes, ao desenvolver um projeto, nos deparamos com problemas complexos que exigem algoritmos avançados ou processamento de dados especializado. As bibliotecas nos fornecem essas soluções prontas para uso, permitindo-nos resolver problemas complexos de forma mais eficiente e eficaz.

COMANDO IMPORT



UNIVERSIDADE
FEDERAL DO PIAUÍ

- Comando **import**:
 - Permite importar uma biblioteca.

Sintaxe:

```
import <nome_biblioteca>
```

Exemplo:

```
import math
```


MÓDULO MATH



UNIVERSIDADE
FEDERAL DO PIAUÍ

- O módulo math é uma biblioteca disponível em Python que possui diferentes funções matemáticas.
- O math é utilizado/importado por meio da instrução import.

MÓDULO MATH



UNIVERSIDADE
FEDERAL DO PIAUÍ

- Exemplos de algumas funções matemáticas do módulo math:
 - `math.sqrt(x)` - retorna a raiz quadrada de x .
 - `math.exp(x)` - retorna a exponencial de x (e^x).
 - `math.log(x)` - retorna o logaritmo natural de x .
 - `math.sin(x)` - retorna o seno de x (em radianos).
 - `math.cos(x)` - retorna o cosseno de x (em radianos).
 - `math.tan(x)` - retorna a tangente de x (em radianos).
 - `math.radians(x)` - converte x de graus para radianos.
 - `math.degrees(x)` - converte x de radianos para graus.

MÓDULO MATH



UNIVERSIDADE
FEDERAL DO PIAUÍ

- Exemplos de algumas funções matemáticas do módulo math:
 - No link a seguir, há outras funções matemáticas.

<https://docs.python.org/pt-br/3/library/math.html>

MÓDULO RANDOM



UNIVERSIDADE
FEDERAL DO PIAUÍ

- O módulo random permite gerar números aleatórios e realizar operações relacionadas.
- Principais métodos da classe Random são:
 - **random()**: retorna um número real aleatório entre 0 e 1.
 - **randint(a, b)**: retorna um número inteiro aleatório entre a e b (inclusive).
 - **choice(seq)**: retorna um elemento aleatório de uma sequência.

MÓDULO RANDOM



UNIVERSIDADE
FEDERAL DO PIAUÍ

- O módulo random permite gerar números aleatórios e realizar operações relacionadas.
- Principais métodos da classe Random são:
 - **shuffle(seq):** embaralha os elementos de uma sequência.
 - **sample(seq, k):** retorna uma lista com k elementos aleatórios de uma sequência, sem repetições.

- Exemplos:

#Exemplo 1: Gerando um número aleatório entre 1 e 10

```
import random
```

```
numero = random.randint(1, 10)
```

```
print(numero)
```

- Exemplos:

#Exemplo 2: Embaralhando uma lista de cartas

```
import random
```

```
cartas = ['A', '2', '3', '4', '5', '6', '7', '8', '9', '10', 'J', 'Q', 'K']
```

```
random.shuffle(cartas)
```

```
print(cartas)
```

- Exemplos:

#Exemplo 3: Selecionando amostras aleatórias de uma população

```
import random
```

```
populacao = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
amostra = random.sample(populacao, 3)
```

```
print(amostra)
```




UNIVERSIDADE
FEDERAL DO PIAUÍ





CTT/UFPI



UNIVERSIDADE
FEDERAL DO PIAUÍ

Até a Próxima Aula!