Introdução ao Python

Do Zero até o primeiro projeto!

FAALLAA MAKERS!

Hérika Ströngreen

Engenheira da Computação, Pós Graduanda em Internet das Coisas (IoT) e Ethical Hacking & Cyber Security. Entusiasta de Hardware Hacking e Embarcados | Security Delivery Analyst na Accenture

Criadora de Conteúdo no Youtube, Instagram, Lolzeira e GymRat

Porque Python?

- Programação de alto nível, interpretada e orientada a objetos;
- Criada por 1991 por Guido van Rossum;
- Usada: Backend, Ciência de dados, Inteligência artificial, Automação de processos e segurança;
- Fácil de aprender;
- Bibliotecas, Módulos para diversas tarefas, como manipulação de arquivos, comunicação em rede, manipulação de dados;
- Comunidade ativa;
- Framewoks e ferramentas: Django e Flask (desenvolvimento de aplicações), NumPy e Pandas (ciência de dados), TensorFlow e Pytorch (modelos de aprendizado de máquina e deep learning);

Configuração do ambiente

- Instalar o python:
  - https://www.python.org/downloads/
- Instalar um editor ou IDE
  - https://code.visualstudio.com/download
- Instalar pacotes Python
  - pip install nome-do-pacote
- Configurar o ambiente virtual (virtualenv)
  - `python3 -m venv nome_do_ambiente_virtual`
  - `source nome_do_ambiente_virtual/bin/activate`
  - `nome_do_ambiente_virtual\Scripts\Activate`

Sintaxe básica de Python

- Comentários

```
# Este é um comentário
print("Olá mundo!")
```

- Variáveis

```
nome = "João"
idade = 25
salario = 1500.50
```

- Operadores aritméticos

```
x = 10
y = 3

soma = x + y
subtracao = x - y
multiplicacao = x * y
divisao = x / y
resto = x % y
```

Sintaxe básica de Python

- Estruturas condicionais

```
idade = 18

if idade < 18:
    print("Você é menor de idade.")
elif idade >= 18 and idade < 60:
    print("Você é adulto.")
else:
    print("Você é idoso.")
```

- Estruturas de repetição

```
# Repetição com while
i = 1
while i <= 10:
    print(i)
    i = i + 1

# Repetição com for
for i in range(1, 11):
    print(i)
```

Tipos de dados

- Números

```
x = 10          # inteiro
y = 3.14        # ponto flutuante
z = 2 + 3j      # número complexo

soma = x + y    # soma de um inteiro e um ponto flutuante
```

- Strings

```
nome = "João"
sobrenome = 'Silva'
frase = "O rato roeu a roupa do rei de Roma."
```

Tipos de dados

- Booleanos
  - >
  - <
  - and
  - or
  - Not
- None
- Listas, tuplas, conjuntos, dicionários e objetos.

```
x = 10
y = 3

# Testa se x é maior que y
resultado = x > y


x = 10
y = 3

# Testa se x é maior que y e menor que 20
resultado = x > y and x < 20



x = None
```

Operadores aritméticos e lógicos

1. Operadores aritméticos:
   - `+` : Adição
   - `-` : Subtração
   - `*` : Multiplicação
   - `/` : Divisão (retorna um ponto flutuante)
   - `//` : Divisão inteira (retorna um inteiro)
   - `%` : Módulo (retorna o resto da divisão)
   - `**` : Exponenciação

```
x = 10
y = 3

# Adição
soma = x + y

# Subtração
diferenca = x - y

# Multiplicação
produto = x * y

# Divisão
divisao = x / y

# Divisão inteira
divisao_inteira = x // y

# Módulo
resto = x % y

# Exponenciação
potencia = x ** y
```

Operadores aritméticos e lógicos

2. Operadores de comparação:
   - `==` : Igual a
   - `!=` : Diferente de
   - `>` : Maior que
   - `<` : Menor que
   - `>=` : Maior ou igual a
   - `<=` : Menor ou igual a

```
x = 10
y = 3

# Testa se x é igual a y
igual = x == y

# Testa se x é diferente de y
diferente = x != y

# Testa se x é maior que y
maior = x > y

# Testa se x é menor que y
menor = x < y

# Testa se x é maior ou igual a y
maior_igual = x >= y

# Testa se x é menor ou igual a y
menor_igual = x <= y
```

Operadores aritméticos e lógicos

3. Operadores lógicos:
   - `and` : Retorna `True` se ambas as expressões são verdadeiras
   - `or` : Retorna `True` se pelo menos uma das expressões é verdadeira
   - `not` : Inverte o valor booleano da expressão

```
x = 10
y = 3
z = 5

# Testa se x é maior que y e z
resultado1 = x > y and x > z

# Testa se x é maior que y ou z
resultado2 = x > y or x > z

# Testa se y não é igual a z
resultado3 = not y == z
```

Funções

- Blocos de código que realizam uma tarefa específica;
- Função de organizar e reutilizar o código;

```
# solicita ao usuário a operação desejada
operacao = input("Qual operação você deseja realizar (+, -, *, /)?  ")
```

```
# Este é um comentário
print("Olá mundo!")
```

Projeto Calculadora

Criar uma calculadora que receba o operador aritmético, o número 1, o número 2 e retorne o resultado.

- Utilize:
  - Variáveis
  - Estruturas condicionais
  - print
  - input

O que é uma API?

- Conhecendo o pip
- Instalar Biblioteca (request)

```
\Curso de Python> py -m pip install --upgrade pip
```

```
\Curso de Python> py -m pip install requests
```

- Código de status HTTP
  - 200
  - 400
  - 500
- Fazer requisições

```
\Curso de Python> python3 -m http.server
```

- Receber informações de uma API

Projeto API - POKEMON

- Url: pokeapi.co
- Como ler documentação
- Endpoint
  - https://pokeapi.co/api/v2/ability/{id ou nome}/
  - https://pokeapi.co/api/v2/berry/{id ou nome}/
- Postman

Projeto API - FLASK

- O que é um Framework? Como usar?
- Criando Endpoint
- Subindo a API
- Usando o Postman