📌Fundamentos de Programação em Python

Pontifícia Universidade Católica de Campinas

Prof. Dr. Denis Mayr Lima Martins

📝Aula 1: Introdução à linguagem Python e ao ambiente Jupyter Notebook

Bem-vindo à nossa primeira aula prática de **Fundamentos de Programação em Python**! 🚀

Nesta aula, vamos dar os primeiros passos no mundo da programação, explorando os conceitos básicos da linguagem **Python** e nos familiarizando com o ambiente **Jupyter Notebook**.

🔷 **O que veremos hoje?**

- O que é programação e como Python se encaixa nesse contexto.
- Introdução ao ambiente **Jupyter Notebook**.
- Escrevendo e executando os primeiros comandos em Python.

🎯**Objetivos de Aprendizagem**

- Familiarizar-se com o ambiente do Jupyter Notebook
- Aprender a **criar, editar e executar** células de código e Markdown.
- Explorar **atalhos e funcionalidades** para otimizar a escrita de código e a organização do material.
- Escrever e executar os primeiros comandos em Python

🛠️Estrutura do Jupyter Notebook

- **Células de Código**: Para escrever e executar comandos Python.
- **Células de Markdown**: Para adicionar explicações, títulos e formatações.
- **Atalhos úteis**:
    - `Shift + Enter`: Executa a célula e avança para a próxima.
    - `Esc + A`: Cria uma nova célula acima.
    - `Esc + B`: Cria uma nova célula abaixo.
    - `Esc + M`: Converte a célula em **Markdown**.

📝Introdução ao Markdown

O **Markdown** é uma linguagem de marcação leve que facilita a formatação de texto usando uma sintaxe simples e intuitiva. É amplamente utilizado em ambientes como o **Jupyter Notebook** para criar documentos que combinam código, texto formatado, imagens e muito mais.

✨Vantagens do Markdown

- **Simplicidade**: Comandos fáceis de aprender e usar.
- **Leitura**: Arquivos em Markdown são legíveis mesmo sem formatação.
- **Flexibilidade**: Permite a inclusão de elementos como listas, links, imagens e trechos de código.

Guia: https://github.com/mende1/guia-definitivo-de-markdown

⌨️Escrever Texto

Vamos experimentar um pouco com a linguagem Markdown.

Este algoritmo computa 2 + 2

🐍Introdução à Linguagem Python

Agora, vamos explorar e executar nossos primeiros comandos em Python! 🚀

Operações Básicas

```
10*5
```

```
1+1+2 # Para colocar um comentário, use #
```

```
1+1#+2
```

```
1-1 # Subtração


```

```
6*5 # Multiplicação

```

```
3/2 # Divisão
```

```
3//2 # Divisão que mostra apenas o número inteiro
```

```
10**3 # Número Elevado a outro
```

```
2**4
```

Precedência do operador

Em Python, a precedência dos operadores é:

1: () o que estiver entre ()

2: **

3: * ou / (o que vier primeiro)

4: + ou - (o que vier primeiro)

```
12 / (2 - 4) #Faz a operacao que estiver em () primeiro
```

```
4 * (5 + 2) * 8 + 1
```

Exercício: Qual é o resultado do código a seguir?

`print(2**2 + (3 - 1) * (1 + 2) - (2 / 2))`

```
2**2 + (3 - 1) * (1 + 2) - (2 / 2)
```

🔎Verificação de mensagens de erro

O Python gera erros para problemas no código. Os erros vêm acompanhados de uma mensagem que (espera-se) explica o que aconteceu e onde está o problema.

Vamos tentar executar uma divisão por zero:

```
5/0
```

💾Salvando alterações

Pressione o atalho de teclado `Ctrl + S` para salvar todas as alterações e garantir que você

não perderá nenhuma informação nova.

📝Adição de comentários no código

Os comentários de código devem ser lidos pelos desenvolvedores. Eles geralmente descrevem o que, por que e/ou como algo foi feito naquela parte do programa. Os comentários devem ser frases completas. Eles devem facilitar a compreensão de códigos complexos escritos por outras pessoas (ou por você mesmo).

Os comentários em Python são marcados com um sinal `#`. Tudo o que vem depois do sinal `#` até o final da linha é ignorado pelo Python:

```
# Um comentário de uma única linha para uma operação simples
1 + 1
```

```
1 + 1 # Os comentários também podem ser colocados logo após o código
```

```
# Mas não à esquerda do código 1 + 1
```

```
1 + # No meio, o código também é problemático 1
```

```
# Para várias linhas,
# use vários '#'
# assim.
1 + 1
```

Imprimir uma mensagem 🖨️

Vamos imprimir uma mensagem.

```
print("Bom Dia")
```

```
print(1)
```

```
input("Digite sua idade")
```

Deixe abaixo sua própria mensagem para o Mundo Python 🐍🌎

```
Start coding or generate with AI.
```


```
Start coding or generate with AI.
```

```
print("Welcome!")
```

```
print("Print 1")
print("Print 2")
```

```
print(1+2)
print(2**3)
```

```
# Ele consegue entender as 2 contas, mas mostra apenas a última.
# Se quiser mostrar os dois resultados, tem que usar 'print'
1+1
2+2
```

```
print("Resultado 1:", 1+1)
print("Resultado 2:", 2+2)
```

Exercício para fazer em sala: Verifique os valores produzidos pelas seguintes operações:

a) `5*2`

b) `2**3`

c) `8 / 3`

d) `2 + 5 + 6`

d) Imprima a mensagem `Hello World`

✅Conclusão

Parabéns por concluir esta aula! 🎉

Hoje, exploramos os conceitos básicos de **Python** e nos familiarizamos com o ambiente do **Jupyter Notebook**. Com esse conhecimento, você já pode começar a escrever e executar seus próprios códigos.

➡️SOONPróximos Passos:

- Praticar os conceitos aprendidos resolvendo pequenos exercícios.
- Explorar mais comandos básicos e entender como o Python pode ser usado para resolver problemas computacionais.
- Continuar aprimorando o uso do **Jupyter Notebook** para organização e documentação do código.

Caso tenha dúvidas, revise os exemplos e experimente modificá-los. A programação se aprende na prática! 🚀

aprende na prática! 🚀

📌**Até a próxima aula!**