A SAGA PYTHON

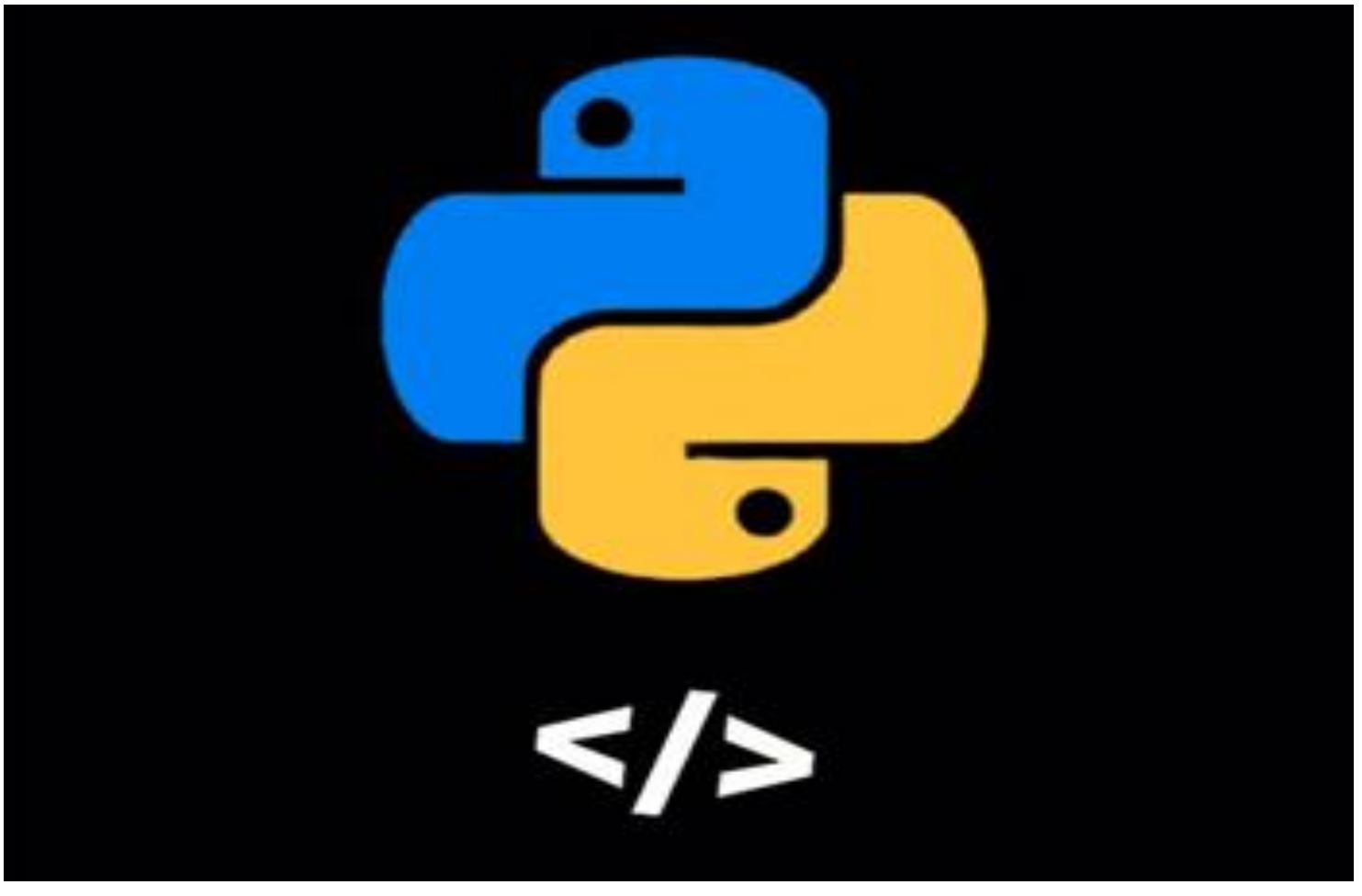
UMA JORNADA PELA GALAXIA DO DESENVOLVIMENTO

MARCOS S. RODRIGUES

Seletores em Python:

Aprendendo a Escolher e Filtrar Dados com Facilidade.

Os seletores em Python são formas de acessar, filtrar ou escolher partes específicas de dados. Eles aparecem o tempo todo no código: quando se busca um item de uma lista, uma chave de um dicionário, ou até quando se escolhe qual ação o programa deve seguir. Este capítulo mostra como esses seletores funcionam de forma clara, com exemplos de uso no dia a dia de quem programa.



01

SELECIONANDO ITENS EM LISTAS

Listas são coleções ordenadas de dados. Cada item tem uma posição (índice) que começa em 0.
Selecionar elementos em listas é uma das tarefas mais comuns em Python.

Esse tipo de seleção é muito usado em sistemas de estoque ou carrinhos de compra, para acessar produtos de forma direta.

```
// produtos = ["mouse", "teclado", "monitor", "webcam"]

primeiro = produtos[0]
ultimo = produtos[-1]

print("Primeiro produto:", primeiro)
print("Último produto:", ultimo)
put your code here
```

snappify.com

02

SELECIONANDO FAIXAS (SLICING)

O slicing permite "fatiar" listas ou textos, acessando várias posições de uma só vez.
A sintaxe básica é [início:fim], sendo o início o índice inicial e fim o índice onde o corte para.

O slicing é útil ao montar relatórios, gerar resumos ou trabalhar com grandes volumes de dados.

```python
vendas = [120, 135, 150, 160, 145, 170, 180]

ultimos_dias = vendas[-3:]  # Últimos 3 dias de vendas
print("Últimos dias:", ultimos_dias)
```


03

SELECIONANDO VALORES EM DICIONÁRIOS

Dicionários guardam informações no formato chave → valor. Para selecionar um valor, basta usar sua chave.

Esse tipo de seleção é comum em APIs, arquivos JSON e sistemas que armazenam dados de usuários.

```python
cliente = {"nome": "Lucas", "idade": 29, "cidade": "São Paulo"}

print("Nome do cliente:", cliente["nome"])

# Usando .get() evita erro se a chave não existir
email = cliente.get("email", "não informado")
print("Email:", email)
```


SELECIONANDO CAMINHOS COM CONDIÇÕES

- Python permite selecionar qual caminho o programa seguirá através de if, elif e else.

Essa estrutura é essencial em regras de negócio, como descontos, cálculo de frete ou aprovação de crédito.

```python
valor_compra = 320

if valor_compra ⩾ 500:
    desconto = 0.15
elif valor_compra ⩾ 300:
    desconto = 0.10
else:
    desconto = 0.05

total = valor_compra * (1 - desconto)
print(f"Total com desconto: R$ {total:.2f}")
```

snappify.com

SELECIONANDO COM OPERADOR TERNÁRIO

Quando a escolha é simples, dá para escrever tudo em uma linha, usando o operador ternário.

Esse formato deixa o código mais limpo e é útil para decisões rápidas em funções ou relatórios.

```python
nota = 8.2
status = "Aprovado" if nota ⩾ 7 else "Reprovado"
print("Status do aluno:", status)
```


06

FILTRANDO DADOS COM LIST COMPREHENSION

List comprehensions permitem criar listas novas a partir de outras listas, filtrando ou transformando os dados de forma compacta.

Essa seleção é muito útil em análise de dados, relatórios e filtragens automáticas em sistemas.

```python
produtos = [
    {"nome": "Mouse", "preco": 80},
    {"nome": "Teclado", "preco": 150},
    {"nome": "Cabo HDMI", "preco": 40},
]

baratos = [p for p in produtos if p["preco"] < 100]

for p in baratos:
    print(f"Produto em oferta: {p['nome']} - R${p['preco']}")
```


AGRADECIMENTOS

OBRIGADO POR LER ATE AQUI

Este ebook foi gerado por IA e Diagramado por um humano.

Este conteúdo foi gerado com fins didáticos de construção, não foi realizado uma validação cuidadosa no conteúdo e pode conter erros gerados por uma IA