Seletores Python

Dominando a Força do Código

Dayane Teodoro

Introdução

Domine os Seletores Python: Um Guia Simples e Prático

Python é uma linguagem poderosa e versátil, amplamente utilizada por desenvolvedores em todo o mundo. Um dos aspectos fundamentais para escrever código eficiente e claro em Python é o uso correto dos seletores. Vamos explorar os principais seletores Python, com explicações simples e exemplos de código em contextos reais.

SELEÇÃO DE ELEMENTOS EM LISTAS

01

1. **Seleção de Elementos em Listas**

As listas são uma estrutura de dados essencial em Python. Podemos acessar seus elementos usando índices.

Python

```python
# Criando uma lista de frutas
frutas = ["maçã", "banana", "cereja", "damasco"]

# Acessando o primeiro elemento
primeira_fruta = frutas[0]
print(primeira_fruta)  # Saída: maçã

# Acessando o último elemento
ultima_fruta = frutas[-1]
print(ultima_fruta)  # Saída: damasco
```

FATIAMENTO DE LISTAS

02

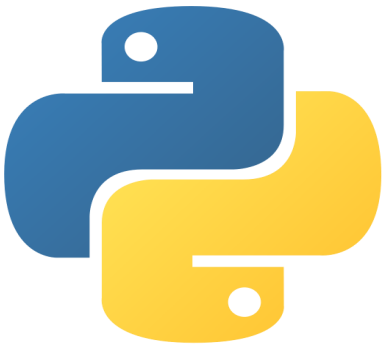
2. Fatiamento de Listas

O fatiamento permite acessar sublistas de uma lista maior.

Python

```python
# Fatiando a lista de frutas
primeiras_duas_frutas = frutas[:2]
print(primeiras_duas_frutas)  # Saída: ['maçã', 'banana']

# Fatiando do segundo elemento até o final
frutas_restantes = frutas[1:]
print(frutas_restantes)  # Saída: ['banana', 'cereja', 'damasco']
```



SELEÇÃO DE ELEMENTOS EM DICIONÁRIOS

03

3. Seleção de Elementos em Dicionários

Dicionários são estruturas de dados que armazenam pares chave-valor.

Python

```python
# Criando um dicionário de informações de um estudante
estudante = {"nome": "Ana", "idade": 22, "curso": "Engenharia"}

# Acessando o valor associado à chave 'nome'
nome_estudante = estudante["nome"]
print(nome_estudante)  # Saída: Ana

# Acessando o valor associado à chave 'idade'
idade_estudante = estudante["idade"]
print(idade_estudante)  # Saída: 22
```

SELEÇÃO DE ELEMENTOS EM CONJUNTOS

04

4. Seleção de Elementos em Conjuntos

Conjuntos são coleções não ordenadas de elementos únicos.

Python

```python
# Criando um conjunto de números
numeros = {1, 2, 3, 4, 5}

# Verificando se um elemento está no conjunto
existe = 3 in numeros
print(existe)  # Saída: True

# Verificando se um elemento não está no conjunto
nao_existe = 6 not in numeros
print(nao_existe)  # Saída: True
```

SELEÇÃO DE CARACTERES EM STRINGS

05

5. Seleção de Caracteres em Strings

Strings são sequências de caracteres, e podemos acessar seus elementos por índices.

Python

```python
# Criando uma string
mensagem = "Olá, Mundo!"

# Acessando o primeiro caractere
primeiro_caractere = mensagem[0]
print(primeiro_caractere)  # Saída: O

# Acessando os últimos três caracteres
ultimos_tres_caracteres = mensagem[-3:]
print(ultimos_tres_caracteres)  # Saída: do!


```

SELETORES AVANÇADOS EM PYTHON

06

Outros tipos de Seletores

Seletores Avançados em Python

Além dos seletores básicos como índices e fatiamento em listas e dicionários, Python oferece uma variedade de seletores avançados que podem tornar a manipulação de dados mais eficiente e poderosa. Vamos explorar alguns desses seletores com exemplos práticos.

SELEÇÃO POR CONDIÇÃO EM ARRAYS COM NUMPY

07

7 - Seleção por Condição em Arrays com NumPy

NumPy permite selecionar elementos de arrays com base em condições, o que é útil para análise de dados.

Python

```python
import numpy as np

# Array de dados de temperatura
temperaturas = np.array([22, 25, 19, 30, 28, 21, 23])

# Selecionando temperaturas acima de 25 graus
altas_temperaturas = temperaturas[temperaturas > 25]

print(altas_temperaturas)  # Saída: [30 28]
```

SELEÇÃO DE COLUNAS E LINHAS EM DATAFRAMES COM PANDAS

08

8. Seleção de Colunas e Linhas em DataFrames com Pandas

Pandas oferece seletores avançados para acessar colunas e linhas específicas em DataFrames.

Python

```python
import pandas as pd

# Criando um DataFrame de exemplo
dados = pd.DataFrame({
    'nome': ['Ana', 'Bruno', 'Carlos', 'Daniela'],
    'idade': [23, 35, 22, 29],
    'salario': [5000, 7000, 5500, 6000]
})

# Selecionando a coluna 'nome'
coluna_nome = dados['nome']

# Selecionando a primeira linha
primeira_linha = dados.iloc[0]

print(coluna_nome)
print(primeira_linha)

```

SELEÇÃO DE SUBSTRINGS EM STRINGS

09

9. **Seleção de Substrings em Strings**

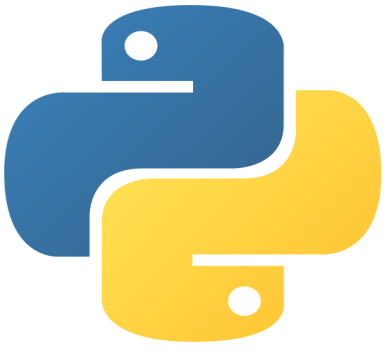
Strings em Python permitem a seleção de substrings usando fatiamento.

Python

```python
# String de exemplo
texto = "Big Data com Python"

# Selecionando a palavra "Data"
palavra_data = texto[4:8]

print(palavra_data)  # Saída: Data
```



SELEÇÃO DE ELEMENTOS EM CONJUNTOS COM COMPREENSÃO DE CONJUNTOS

10

10. Seleção de Elementos em Conjuntos com Compreensão de Conjuntos

Compreensão de conjuntos permite selecionar elementos de conjuntos com base em condições.

Python

```python
# Conjunto de números
numeros = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

# Selecionando números pares
numeros_pares = {x for x in numeros if x % 2 == 0}

print(numeros_pares)  # Saída: {2, 4, 6, 8, 10}

```

SELEÇÃO EM DICIONÁRIOS COM COMPREENSÃO DE DICIONÁRIOS

11

11. Seleção em Dicionários com Compreensão de Dicionários

Compreensão de dicionários permite criar novos dicionários filtrando elementos.

Python

```python
# Dicionário de preços de produtos
precos = {
    'banana': 3.50,
    'maçã': 2.00,
    'laranja': 4.00,
    'uva': 5.00
}

# Selecionando produtos com preço maior que 3.00
precos_altos = {k: v for k, v in precos.items() if v > 3.00}

print(precos_altos)  # Saída: {'banana': 3.5, 'laranja': 4.0, 'uva': 5.0}


```

Conclusão

Entender e utilizar corretamente os seletores em Python é fundamental para escrever código mais eficiente e legível. Este guia prático abordou os principais seletores em listas, dicionários, conjuntos e strings, com exemplos simples e contextos reais. Com essa base, você está pronto para explorar ainda mais as possibilidades que o Python oferece.

Também exploramos seletores avançados em Python, incluindo seleção por condição em arrays NumPy, seleção de colunas e linhas em DataFrames Pandas, seleção de substrings em strings, e seleção de elementos em conjuntos e dicionários com compreensão. Esses seletores podem tornar seu código mais eficiente e expressivo, facilitando a manipulação e análise de dados.

Obrigado por ler até aqui

Esse ebook foi gerado pela IA, e diagramado por humano.
O passo a passo se encontra no meu Github.

GitHub

Link: https://github.com/Dayanebiaerafa

.

Obrigado por ler meu ebook! Espero que você tenha encontrado as informações úteis e que elas possam ajudá-lo a aprimorar suas habilidades em Seletores Python. Se você tiver qualquer dúvida ou feedback, não hesite em entrar em contato. Sucesso em sua jornada de aprendizado!