

PARADIGMAS DE LINGUAGENS DE PROGRAMAÇÃO EM PYTHON

Profº: Sebastião Rogério

Introdução ao Python

Python é amplamente utilizado em áreas como desenvolvimento web, ciência de dados, inteligência artificial e automação de tarefas. Sua simplicidade e legibilidade fazem dele uma linguagem favorita tanto para iniciantes quanto para desenvolvedores experientes.

Manipulação de Variáveis e Strings

Variáveis: Armazenam dados e não precisam ter o tipo declarado.

- Devem começar com letras e podem incluir números e sublinhado (_).
Entrada de dados: Use `input()` para receber dados do usuário, que por padrão é uma string.
- Conversão de tipos: Embora o `input()` sempre retorne uma string, você pode converter facilmente para outros tipos como `int()`, `float()` e `bool()`.

Exemplo:

```
idade = int(input("Digite sua idade: "))
```

- **Fatiamento de strings:** Python permite acessar partes específicas de uma string usando índices.

Exemplo:

```
texto = "Python"  
print(texto[0:3]) # Saída: Pyt
```

Mais métodos de manipulação de strings

MÉTODO	DESCRIÇÃO	EXEMPLO
UPPER()	Converte para maiúsculas	"python".upper() ➔ 'PYTHON'
LOWER()	Converte para minúsculas	"PYTHON".lower() ➔ 'python'
REPLACE()	Substitui partes da string	"banana".replace("a", "o") ➔ 'bonono'
STRIP()	Remove espaços em branco no início e fim	" teste ".strip() ➔ 'teste'

Curiosidade: Strings são imutáveis, ou seja, não podem ser alteradas diretamente.



Dicionários em Python

- Dicionários armazenam dados como pares de chave: valor, sendo acessados através de suas chaves.

Exemplo de iteração sobre dicionários:

```
dados = {"Nome": "Ana", "Idade": 25}
for chave, valor in dados.items():
    print(f"{chave}: {valor}")
```

Dica: Dicionários são ideais para representar objetos do mundo real e estruturas de dados complexas.

Estruturas de Decisão

- **Operadores Lógicos:** Python possui operadores lógicos que são frequentemente utilizados em estruturas condicionais: and, or, not.

Exemplo prático:

```
idade = 20
nome = "Ana"
if idade >= 18 and nome == "Ana":
    print("Maior de idade chamada Ana")
```

Estruturas de Repetição

- Além do while, temos a estrutura for, que é muito utilizada
- para iterar sobre sequências.

Exemplo de for loop: *nomes = ["Ana", "Carlos", "Beatriz"]*

```
for nome in nomes:
    print(nome)
```

Range é muito útil em loops:

```
for i in range(5):
    print(i) # Imprime de 0 a 4
```

Trabalhando com Funções

As funções permitem reutilizar código e organizá-lo de forma lógica.

Definição de uma função:

```
def saudacao(nome):
```

```
    print(f"Olá, {nome}!")
```

Importando Módulos e Bibliotecas

- Python possui uma vasta biblioteca padrão e também permite a importação de pacotes externos para expandir suas funcionalidades.

Exemplo de importação:

```
import math
```

```
print(math.sqrt(16)) # Saída: 4.0
```

Manipulação de Arquivos

Python facilita a leitura e escrita de arquivos.

Exemplo de leitura de arquivos:

with open('dados.txt', 'r') as arquivo:

```
    conteudo = arquivo.read()
```

```
    print(conteudo)
```



**Bons
estudos!**

