Básico de Python

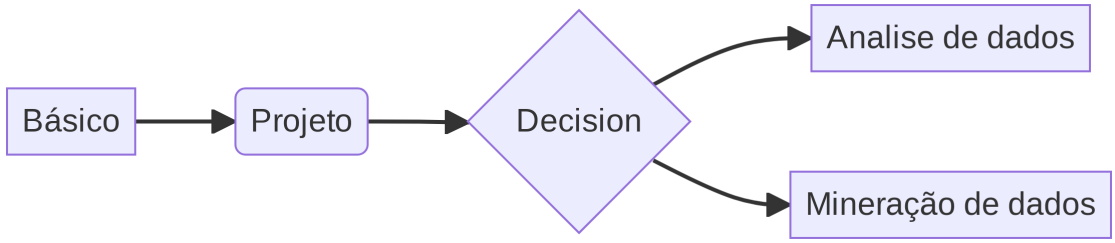
Sergio Pedro Rodrigues Oliveira

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

1 Diagrama de estudo



2 Variáveis e tipos de dados simples

2.1 `print()`

Print é uma função que exibe uma string na tela.

Exemplo: `print("string")`

2.2 `print()` com variáveis

Podemos usar a função `print()` para imprimir uma variável string.

Exemplo:

```
message = "Hello world!"
print(message)
```

2.3 Regras de nomes de variáveis

Regras ou diretrizes para usar variáveis em Python.

- Nomes de variáveis deve conter apenas letras, números e underscores. Podemos começar a variável com letra ou underscore, mas nunca com um número.
- Espaços não são permitidos em nomes de variáveis, mas underscores podem ser usados para separar palavras.
- Evite usar palavras reservadas e nome de funções em Python como nome de variáveis.
- Nomes de variáveis devem ser concisos, porém descritivos.
- Tome cuidado ao usar a letra l e a letra maiuscula O, pois podem ser confundidas com os números 1 e 0.

2.4 Concatenando strings

Podemos usar o simbolo de (+) para combinar strings (concatenar).

Exemplo:

```
>>> first_name = "ada"
>>> last_name = "lovelace"
>>> full_name = fist_name + " " + last_name
>>> print("Hello, " + full_name.title() + "!")
Hello, Ada Lovelace!
```

Os espaços em branco entre aspas servem para criar espaços na string.

2.5 Métodos auxiliares da função `print()`

1. `.title()`

Coloca apenas as primeiras letras em maiúsculas de cada palavra e o resto em minúscula.

Exemplo:

```
>>> print(full_name.title())`
Ada Lovelace
```

2. `.upper()`

Coloca todas as letras em maiúsculas.

Exemplo:

```
>>> print(full_name.upper())`
ADA LOVELACE
```

3. `.lower()`

Coloca todas as letras em minusculas. O método `.lower()` é particularmente útil para armazenar dados. Converter os dados em minúscula antes de armazenar.

Exemplo:

```
>>> print(full_name.lower())`
ada lovelace
```

2.6 Caracteres de escape

Podemos inserir alguns caracteres de escape no texto para executar alguma ação, como pular linha, gerar tabulação e etc. Alguns caracteres podem ser vistos na Table 1.

Todos os caracteres de escape começam com barra(\) + complemento.

Table 1: Caracteres de escape

Caracteres de escape	Descrição
\t	Gera tabulação (tab).
\n	Gera quebra de linha.

Exemplo:

```
print("Language:\nPython\nJava\nC\nJavaScript")
```

```
Language:
Python
Java
C
JavaScript
```

2.7 Removendo espaços em branco `print()`

1. `.rstrip()`

Remove espaço em branco do lado direito.

Exemplo:

```
favorite_linguage = 'python '
favorite_linguage.rstrip()
```

```
'python'
```

2. `.lstrip()`

Remove espaço em branco do lado esquerdo.

Exemplo:

```
favorite_linguage = ' python'
favorite_linguage.lstrip()
```

```
'python'
```

3. `.strip()`

Remove os espaços em branco dos dois lados ao mesmo tempo.

Exemplo:

```
favorite_linguage = ' python '
favorite_linguage.strip()
```

```
'python'
```

- Os metodos usados não removem os espaços em branco em definitivo, para remover em definitivo é necessario armazenar o valor novo na variável.

```
favorite_linguage = ' python '
favorite_linguage = favorite_linguage.strip()
favorite_linguage
```

```
'python'
```

2.8 Números

A linguagem Python faz tipagem automática (dinâmica), tipa a variável de acordo com o uso. E o Python contém uma tipagem forte, não faz converção automática do tipo de uma variável para executar uma ação (operação).

Em resumo, python tem é uma linguagem de tipagem dinâmica e forte.

Os principais tipos de dados no Python são estão presentes na Table 2.

Table 2: Principais tipos de dados

Nome	Abreviação	Descrição
Inteiro	int	Números inteiros
Ponto flutuante	float	Números com ponto decimal

2.9 Funções de conversão de tipo

Table 3: Funções de conversão de tipo

Tipo para converter	Função	Descrição
int	`int()`	Converte variável para tipo inteiro(int)
float	`float()`	Converte variável para tipo float

2.10 Operações básicas

A Table 4 apresenta as principais operações básicas do python.

Table 4: Operações básicas

Operação	Símbolo	Exemplo
Soma	+	2+2=4
Subtração	-	3-2=1
Multiplicação	*	2*3=6
Divisão	/	5/4=1.25
Divisão inteira	//	5//4=1
Resto da divisão (módulo)	%	10%8=2
Potência	**	3**2=9
Raiz	**	4**0.5=2

Podemos usar o pacote `math` para ampliar as funções matemáticas do Python. A Table 5 apresenta as principais funções básicas da biblioteca `math`.

Modificar a tabela Table 5.

Table 5: Algumas operações da biblioteca `math`

Operação	Símbolo	Exemplo
Soma	+	2+2=4
Subtração	-	3-2=1
Multiplicação	*	2*3=6
Divisão	/	3/2=1
Resto da divisão	%	10%8=2
Potência	**	3**2=9
Raiz	**	4**0.5=2

2.11 Operações lógicas básicas

A Table 6 apresenta as principais operações lógica básica do python. As operações lógicas retornam `True` ou `False`.

Table 6: Operações Lógicas

Operação	Nome	Função	Exemplo
==	Igual a	Varifica se um valor é igual ao outro.	1==1 = True
!=	Diferente de	Varifica se um valor é diferente ao outro.	1!=2 = True
>	Maior que	Varifica se um valor é maior que outro.	5>1 = True
>=	Maior ou igual	Varifica se um valor é maior ou igual a outro.	5>=5 = True
<	Menor que	Varifica se um valor é menor que outro.	1<5 = True
<=	Menor ou igual	Varifica se um valor é menor ou igual a outro.	1<=4 = True
and	E	Retorna True se ambas as afirmações forem verdadeiras.	(1==1) and (4<5)
or	Ou	Retorna True se uma das afirmações for verdadeiras.	(1==1) or (2<1) =
not	Negação	Retorna Falso se o resultado for verdadeiro, ou o contrario.	not (1==1) = Fals

2.12 Operadores de identidade

Os operadores de identidade, Table 7, são utilizados para comparar objetos, se os objetos testados referenciam o mesmo objeto.

Table 7: Operadores identidade

Operador	Definição
is	Retorna `True` se ambas as variáveis são o mesmo objeto.
is not	Retorna `True` se ambas as variáveis não são o mesmo objeto.

Exemplo de operações de identidade:

```
lista = [1,2,3]
outra_lista = [1,2,3]
recebe_lista = lista

print(f"São o mesmo objeto: {lista is outra_lista}")
```

```
São o mesmo objeto: False
```

```
lista = [1,2,3]
outra_lista = [1,2,3]
recebe_lista = lista

print(f"São o mesmo objeto: {lista is recebe_lista}")
```

```
São o mesmo objeto: True
```

2.13 Operações de associação

Os operadores de associação, Table 8, servem para verificar se determinado objeto esta **associado** ou **pertence** a determinada estrutura de dados.

Table 8: Operadores de associação

Operação	Função
in	Retorna `True` caso valor seja encontrado na sequência.
not in	Retorna `True` caso valor não seja encontrado na sequência.

Exemplos de operações de associação:

```
lista = ["Python", 'Academy', "Operadores", 'Condições']
print('Python' in lista)
```

```
True
```

```
lista = ["Python", 'Academy', "Operadores", 'Condições']
print('SQL' not in lista)
```

```
True
```

2.14 Evitando erros de tipo com a função `str()`