

Básico de Python

Sergio Pedro Rodrigues Oliveira

SUMÁRIO

| | | |
|----------|---|----------|
| 1 | Diagrama de estudo | 1 |
| 2 | Variáveis e tipos de dados simples | 1 |
| 2.1 | <code>print()</code> | 1 |
| 2.2 | <code>print()</code> com variáveis | 1 |
| 2.3 | Regras de nomes de variáveis | 1 |
| 2.4 | Concatenando strings | 2 |
| 2.5 | Métodos auxiliares da função <code>print()</code> | 2 |
| 2.6 | Caracteres especiais de texto | 3 |

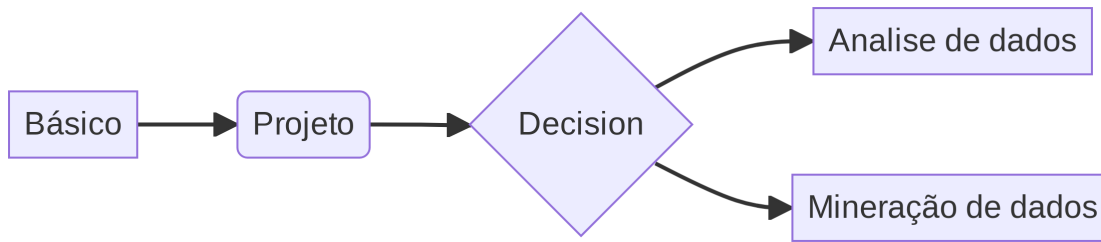
LISTA DE FIGURAS

LISTA DE TABELAS

| | | |
|---|--------------------------------|---|
| 1 | Caracteres especiais | 3 |
|---|--------------------------------|---|

```
from IPython.display import Markdown
from tabulate import tabulate
```

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 maiúscula 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 = first_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 especiais de texto

Podemos inserir alguns caracteres especiais no texto para executar alguma ação, como pular alguma linha e etc. Alguns caracteres podem ser vistos na ?@tbl-CarcEsp. todos os caracteres especiais começam com barra(\) + complemento.

Table 1: Caracteres especiais

| Caracteres especiais | Descrição |
|----------------------|-----------------------|
| t | Gera tabulação. |
| n | Gera quebra de linha. |