

Algoritmos e Estrutura de Dados I

Prof. Gustavo Willam Pereira
Prof. Diego Rossi

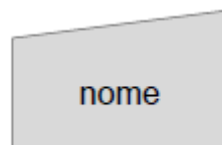


INSTITUTO FEDERAL
Sudeste de Minas Gerais

Python

- Comando input

No fluxograma quando precisamos de uma entrada de dados (que o usuário digita um dado), usamos a figura:



- Na linguagem Python, temos o comando input.
`nome = input("Digite o seu nome")`

Python

- Comando print

No fluxograma quando queremos exibir algo utilizamos a figura:



No Python usamos a função print
`print("Hello World")`

Variáveis no Python

- Assim como no fluxograma, na linguagem Python o uso de variáveis é fundamental.
- Python não é uma linguagem fortemente tipada, ou seja, você não precisa dizer o
- tipos das variáveis.
- As regras para dar nomes as variáveis são as mesmas (não usar número no início da variável e não usar caracteres especiais).

`a = 5`

Tipos de Dados no Python

Tipo	Exemplo
Inteiro	5
Real	1.5
Caractere	"Av. Brasil, 1500"
Lógico	True / False

Operadores Aritméticos no Python

Operador	Descrição
<code>^</code>	Potenciação
<code>*</code>	Multiplicação
<code>/</code>	Divisão (com resultado fracionário)
<code>//</code>	Divisão (com resultado inteiro)
<code>+</code>	Adição
<code>-</code>	Subtração
<code>%</code>	Resto da divisão
<code>**</code>	Exponenciação

Operadores Relacionais no Python

- Os operadores relacionais permitem realizar comparações entre valores, gerando como resultado um valor lógico (Verdadeiro ou Falso).

Operador	Descrição
==	Igual
!=	Diferente
<	Menor
<=	Menor ou igual
>	Maior
>=	Maior ou igual

Operadores Lógicos no Python

- Assim como os relacionais, estes operadores também geram valores lógicos.
- Porém, ao contrário dos operadores relacionais que podem comparar qualquer tipo de dado, os operadores lógicos trabalham apenas com valores lógicos.

Operador	Operação
not	Negação
and	Conjunção
or	Disjunção

Negação no Python

- Negação é uma operação lógica que gera como saída um valor inverso ao valor lógico de entrada.
- O operador NÃO também é usualmente representado por ! ou NOT.

A	.NÃO. A
V (True)	F (False)
F (False)	V (True)

Trabalhando com Strings no Python

- Uma string em Python é uma lista e também imutável;
- Qualquer posição pode ser acessado dentro da string;
- Obs.: Todas strings iniciam na posição 0;

```
S = "Alô mundo"  
print(S)
```

Variáveis String no Python

- As variáveis do tipo string é uma cadeia de caracteres.
- O tamanho da string pode ser obtido através da função len.
`len("a")`
- A junção de duas strings é realizada através da concatenação (+)
`nome = "Gustavo "`
`nome + "Willam"`
- É possível capturar apenas um caracter da string
`nome[0]`

Variáveis String no Python

- Podemos realizar a composição de uma string, ou seja, juntar textos fixos com valores de variáveis.

```
nome = input("Digite o seu nome")
```

```
saldo = 51.34
```

```
print("Olá {nome}, seu saldo é R${saldo}")
```

```
print("Olá {}, seu saldo é R${:5.2f}".format(nome, saldo))
```

- Podemos também fatiar uma string

```
nome = "ABCD"
```

```
nome[0:2]
```

- Exibe: AB

```
nome[1:2]
```

- Exibe: BC

Verificação parcial de strings no Python

- Para verificar se uma string começa (startswith) ou termina(endswith) com uma determinada palavra ou sequência de caracteres.

```
nome = "João da Silva"  
nome.startswith("João")  
nome.endswith("Silva")
```

Para verificar se uma palavra existe na palavra, independente da posição.

“da” in nome

Quando a palavra é encontrada o resultado é True.

Maiúsculas / Minúsculas

- A linguagem Python é case-sensitive (diferencia maiúscula de minúscula).
- É possível converter a string para maiúscula ou minúscula para comparar.

nome = "João da Silva"

nome.upper()

nome.lower()

- Podemos verificar se todas as letras de uma string é maiúscula/minúscula

nome.isupper()

nome.islower()

Contando Ocorrências

- É possível contar as ocorrências de uma palavra dentro de uma string.

t = “um tigre, dois tigres, três tigres”

t.count(“tigre”)

- Exibe 3 como resultado, pois aparecem três vezes a palavra.
- Se for preciso saber a posição da primeira ocorrência da palavra, se não encontrar a palavra, retorna -1;

t.find(“dois”)

- Retorna 10, pois é a primeira posição da ocorrência da palavra dois.
- Para saber a última posição da ocorrência, usamos o rfind(“dois”).

Manipulação de Strings

- Para separar/fatiar a string em uma lista.

frase= “um tigre, dois tigres, três tigres”

frase.split(“,”)

- O resultado é três strings em formato de lista [“um tigre”, “dois tigres”, “três tigres”]

frase.split(“ ”)

- O resultado é seis strings em formato de lista [“um”, “tigre,”, “dois”, “tigres,”, “três”, “tigres”]

Manipulação de Strings

- Para substituir parte de uma string.

```
frase= "um tigre, dois tigres, três tigres"
```

```
frase.replace("tigre", "gato")
```

- O resultado é "um gato, dois gatos, três gatos"
- Para efetuar a remoção do excesso de espaço em branco do início ou fim da string.

```
t = "  Olá  "
```

```
t.strip()
```

```
t.lstrip()
```

```
t.rstrip()
```

Manipulação de Strings

- Para saber o tamanho de uma string usamos a função len

frase= “um tigre, dois tigres, três tigres”

len(frase)

Conversão de dados no Python

- A função input retorna valores do tipo string, ou seja, não importa se digitarmos apenas números, o resultado é sempre é string.

```
anos = int(input("Digite a sua idade"))  
saldo = float(input("Digite o seu saldo"))
```



INSTITUTO FEDERAL
Sudeste de Minas Gerais