

Fundamentos de Programação

Prof. Márcio Miguel Gomes



JESUÍTAS BRASIL



Variáveis

- Durante a execução de um programa, é necessário manipular dados e armazená-los em memória para uso futuro
- Dados estáticos são chamados de constantes
- Dados dinâmicos são chamados de variáveis
- As variáveis possuem nome, tipo de dado e valor

Variáveis

- No Python não existe declaração de variáveis, elas passam a existir no momento em que um valor é atribuído a um nome, na sintaxe: *nome = valor*

```
nome = 'Maria'  
idade = 23  
nome = 'Mário'  
idade = 25
```

Variáveis

- Regras básicas para nomear variáveis:
 - Todo o nome só pode conter letras e dígitos
 - O caractere "_" é considerado como uma letra
 - Todo primeiro caractere deve ser sempre uma letra (ou "_")
 - Letras maiúsculas e minúsculas são consideradas caracteres diferentes (variavelUM é diferente de VariavelUM)
 - Palavras reservadas não podem ser usadas como nome de variáveis

Variáveis

- Boas práticas para nomear variáveis
- Utilizar nomes sugestivos
- Nomes como a, b, c, x, y, z, i, j devem ser utilizados em contextos restritos
- C, C++, C#, Java costumam utilizar a notação *camelCase*:
 - nome, idade, nomeDaMae, salarioPorHora, valorMaximo
- Python costuma utilizar a notação *snake_case*:
 - nome, idade, nome_da_mae, salario_por_hora, valor_maximo

Tipos de Dados

- Os tipos de dados definem um conjunto de padrões utilizados para manipular o conteúdo armazenado em uma variável
- Operações como leitura, escrita, lógica, matemática e uma infinidade de ações dependem diretamente dos tipos de dados envolvidos
- Os tipos de dado primitivos são:
 - Numérico (inteiro e real/fracionário)
 - Literal (letra ou texto)
 - Lógico (booleano)

Tipos de Dados

- **Inteiros:** números que não possuem parte fracionária e podem ser tanto positivos quanto negativos
- **Reais:** números que podem possuir parte fracionária e podem ser tanto positivos quanto negativos
- **Literal:** uma ou mais letras, dígitos ou símbolos especiais, delimitados por aspas simples ou aspas duplas. Também conhecidos por *char* ou *string*
- **Lógico:** possuem apenas dois valores possíveis, *verdadeiro* ou *falso*

Tipos de Dados

Inteiros

```
int_1 = 0
```

```
int_2 = 12
```

```
int_3 = -23
```

Reais ou Fracionários

```
real_1 = 0.0
```

```
real_2 = 12.3
```

```
real_3 = -23.4
```

Literais

```
literal_1 = 'A'
```

```
literal_2 = "Uma frase"
```

```
literal_3 = 'Abc 123 @$'
```

Lógicos

```
logico_1 = True
```

```
logico_2 = False
```


Tipos de Dados

- Mesmo que no Python os tipos de dados sejam dinâmicos, isso não significa que as variáveis não possuam um tipo de dados específico. Para consultar o tipo, use o comando “type()”

```
>>> int_1 = 0
>>> type(int_1)
<class 'int'>
>>> real_1 = 1.2
>>> type(real_1)
<class 'float'>
>>> literal_1 = 'A'
>>> type(literal_1)
<class 'str'>
```

Entrada de Dados

- Para interagir com o usuário e receber dados de entrada, no Python utilizamos o comando “input()”
- O comando “input()” sempre retorna um dado literal, mesmo que seja digitado um valor numérico ou lógico
- Para converter o valor no tipo desejado, usamos um “type cast”, que consiste em um comando com o próprio nome do tipo de dados, como “int()” ou “float()”
- Também é possível utilizar o comando “eval()”

Entrada de Dados

```
>>> nome = input('Informe seu nome: ')
Informe seu nome: Márcio
>>> idade = input('Informe sua idade: ')
Informe sua idade: 22
>>> type(nome)
<class 'str'>
>>> type(idade)
<class 'str'>
```

```
>>> nome = input('Informe seu nome: ')
Informe seu nome: Márcio
>>> idade = int(input('Informe sua idade: '))
Informe sua idade: 22
>>> type(nome)
<class 'str'>
>>> type(idade)
<class 'int'>
```

Saída de Dados

- Para interagir com o usuário e exibir dados na tela, no Python utilizamos o comando “print()”
- O comando “print()” deve receber no mínimo um valor a ser impresso na tela, mas pode receber uma lista de valores separados por vírgula
- Os valores passados ao comando “print()” podem ser de diversos tipos

Saída de Dados

```
nome = input('Informe seu nome: ')\nidade = input('Informe sua idade: ')\nprint(nome)\nprint(idade)\nprint('O nome informado foi', nome)\nprint('Olá', nome, 'sua idade é', idade, 'anos')
```

Saída de Dados

- Para uma melhor organização, é possível separar a estrutura do texto dos dados variáveis, utilizando o comando “format”
- Dessa forma, os dados assumem as posições indicadas entre “{ }” na mesma ordem em que foram declaradas

```
nome = input('Informe seu nome: ')\nidade = input('Informe sua idade: ')\nprint('Olá {} sua idade é {} anos'.format(nome, idade))
```

Saída de Dados

- Também é possível utilizar formatadores avançados, informando dentro de “{}” o padrão desejado

```
nome = input('Informe seu nome: ')\naltura = float(input('Informe sua altura: '))\nprint('Olá {} sua altura é {:.2f} metros'.format(nome, altura))
```

Saída de Dados

- Zeros à esquerda do número: `{:0nd}`, onde 'n' é a quantidade total de dígitos
- Limitando casas decimais: `{:.nf}`, onde 'n' é a quantidade total de casas decimais
- Hexadecimal: `{:X}`

```
print('Inteiro: {:d}'.format(12345))  
print('Zeros à esquerda: {:04d}'.format(25))  
print('Casas decimais: {:.2f}'.format(3.14159))  
print('Hexadecimal: {:X}'.format(255))  
print('Exponencial: {:E}'.format(234.567))
```


Atividades

1. Imprima o seu primeiro nome usando texto fixo e em seguida usando uma variável
2. Imprima a sua idade, utilizando uma variável numérica
3. Imprima a sua altura com 2 casas decimais, utilizando uma variável numérica
4. Imprima os números 1, 10, 100 e 1000 utilizando um espaço de 7 colunas, alinhando o conteúdo à esquerda, depois centralizando e em seguida alinhando à direita. Para essa atividade, consulte as referências na última página.

Atividades

5. Solicite ao usuário o dia do seu nascimento e imprima no terminal o dia informado
6. Solicite ao usuário o dia e mês do seu nascimento e imprima no terminal o dia e mês informados
7. Solicite ao usuário o dia, mês e ano do seu nascimento e imprima no terminal a data completa no formato dd/mm/yyyy
8. Solicite ao usuário que ele informe um número fracionário e o imprima como um número inteiro e depois como um número fracionário com três casas decimais
9. Solicite ao usuário que ele informe um caractere representando uma moeda e em seguida solicite um número fracionário representando um valor. Mostre na tela a moeda + \$ + valor com 2 casas decimais

Referências

- <https://docs.python.org/3/library/string.html>
- <https://www.w3schools.com/python/default.asp>
- https://www.w3schools.com/python/ref_string_format.asp
- <https://pyformat.info/>