
GAG009: Alg e Prog de Computadores

Aula 01 – Variáveis

Prof. Anilton

Faculdade de Computação

Universidade Federal de Uberlândia



Linguagem Python



- ✓ **Linguagem de alto nível**
 - Sintaxe simples: fácil de aprender;
 - Suporta programação procedimental e orientada objetos
 - Diversas bibliotecas disponíveis;
- ✓ **Pode ser usada em diferentes Sistemas Operacionais**
 - Windows, Linux, Mac OS, etc.
- ✓ **Usada no “mundo real”**
 - Google, Microsoft, Yahoo!, NASA, etc.
- ✓ **Site oficial:** <http://www.python.org>

Ambientes de Desenvolvimento

IDLE

- `sudo apt-get install build-essential`
- `sudo apt-get install zlib1g-dev`
- `cd /opt`
- `sudo wget`

<https://www.python.org/ftp/python/3.6.3/Python-3.6.3.tgz>

- `sudo tar -xvf Python-3.6.3.tgz`
- `cd Python-3.6.3`
- `sudo ./configure`
- `sudo make`
- `sudo make install`

PyCharm

<https://www.jetbrains.com/pycharm/download/>

Uso Interativo



✓ Comandos são executados no terminal do Python

Two terminal windows are shown side-by-side. The left window, titled '*Python 2.7.13 Shell*', displays the Python 2.7.13 startup information, including the version, date, and compiler details. The right window, titled 'jrsouza — Python · idle — 80x24', shows the user logging in and running the 'idle' command in a terminal.

- Podemos escrever os comandos em Python e observamos os resultados imediatamente.

A terminal window titled 'Python 2.7.13 Shell' showing the execution of several Python commands. The output shows the result of an addition, the output of a print statement, and the value of a variable.

Arquivo de script



- Podemos criar um arquivo de texto (File -> New File), onde iremos escrever um programa completo (Script) e executar de uma só vez pressionando **F5**;
✓ **Arquivos de programas Python têm extensão .py**

```
a = int(input("Digite o valor de A: "))
b = int(input("Digite o valor de B: "))
while a <= b:
    print (a)
    a = a + 1

print ("Fim do programa")
```

Variáveis



✓ **Computação**

- É uma posição de memória que armazena um valor;
- Pode ser utilizada e modificada pelo programa;
- Deve ser definida antes de ser usada.

✓ **Vejam algumas Propriedades**

• Nome

- Pode ter um ou mais caracteres

• Tipo

- Conjunto de valores aceitos (inteiro, reais, string);

• Escopo

- global ou local

Variáveis



- Nome

- ✓ É recomendável iniciar com letras;
- ✓ Podem conter letras, n.º ou underscores (_);
- ✓ Letras maiúsculas e minúsculas são diferentes;
- ✓ **Palavras reservadas** não podem ser usadas como nomes.

and	def	nonLocal	if	not	return
assert	del	finally	import	or	try
break	elif	for	in	pass	while
class	else	from	is	as	yield
continue	except	global	lambda	raise	with
True	False	None			

Variáveis



- ✓ Quais nomes de variáveis estão corretos?
- Contador
 - contador1
 - comp!
 - .var
 - Teste_123
 - _teste
 - int1
 - 1contador
 - -x
 - Teste-123
- Contador, contador1, Teste_123, _teste, int1
 - **comp!, 1contador, -x, Teste-123** **.var,**

Variáveis



- Tipo

- O tipo define os valores que a variável pode assumir e as operações que podem ser realizadas com ela;
- ✓ Diferente de outras linguagens de programação, não precisamos definir o tipo de uma variável:
 - Comando **type(x)** permite saber o tipo da variável em x.

```
Python 2.7.13 (default, Dec 18 2016, 07:03:39)
[GCC 4.2.1 Compatible Apple LLVM 8.0.0 (clang-800.0.42.1)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> WARNING: The version of Tcl/Tk (8.5.9) in use may be unstable.
Visit http://www.python.org/download/mac/tcltk/ for current information.
a = 2
>>> type(a)
<type 'int'>
>>> a = 2.0
>>> type(a)
<type 'float'>
>>> a = "ola"
>>> type(a)
<type 'str'>
>>> |
```

- int - Tipo inteiro
- float - n.º fracionários (ponto flutuante)
- Str - Cadeia de caracteres, Escrevemos o texto entre aspas **'simples'** ou **"duplas"**



Escopo de Variáveis

- ✓ Escopo - Define onde e quando a variável pode ser usada;
 - ✓ Global - Variável definida fora de qualquer função;
 - ✓ Local - Variável definida dentro de uma função ou na sua lista de parâmetros existentes;
- **x** e **y** são variáveis locais
 - Definidas em um bloco indentado
 - **x** e **z** são variáveis globais
 - Definidas em um bloco não-indentado
 - **x** existe nos dois escopos
 - A variável definida no escopo local ofusca **completamente** o escopo global

Exemplo

```
def func():  
    x = 20  
    y = 10  
    print("Função = ", x, y, z)
```

```
x = 5;  
z = 15  
print("Antes = ", x, z)  
func()  
print("Depois = ", x, z)
```

Saída

```
>>>  
Antes = 5 15  
Função = 20 10 15  
Depois = 5 15  
>>>
```

Comando Saída/Entrada Dados



✓ Função **print()**

- Realiza impressão dos dados do programa no terminal
- Forma: **print(expressão1, expressão2, ..., expressãoN)**

Exemplo

```
x = 1
print(x)
print("Teste")
y = 2.5
print("Y = ",y)
```

Saída

```
>>>
1
Teste
Y = 2.5
>>>
```

✓ Função **input()**

- Realiza a leitura dos caracteres do teclado no terminal
- Forma: **variável = input(texto)**

Exemplo

```
str = input("Digite um texto: ")
print("Texto digitado:", str)
```

Saída

```
>>>
Digite um texto: Python
Texto digitado: Python
>>>
```

Comandos Entrada de Dados



- O que fazer se for preciso ler valor numérico?
 - ✓ Podemos forçar a **conversão de tipos**

- Permite converter um tipo em outro, como?
 - ✓ **int(x)**: converte x para inteiro via truncagem (parte inteira é considerada);
 - ✓ **round(x)**: converte x para inteiro via arredondamento;
 - ✓ **float(x)**: converte x para ponto-flutuante;
 - ✓ **str(x)**: converte um valor x para texto.

```
>>>
>>> str(5.25)
'5.25'
>>>
>>>
>>> int(2.5)
2
>>> int("2")
2
>>> round(3.5)
4
>>> float("3.5")
3.5
>>> str(5.25)
'5.25'
>>>
```

Comentários



- ✓ Adicionam uma descrição sobre o programa. Ajuda a lembrar como funciona o programa;
- ✓ # São ignorados pelo interpretador;
- ✓ Começam com o símbolo # e se estendem até ao fim da linha do código em questão.

```
def fatorial(N):  
# Função que calcula o fatorial de um número inteiro N  
# Exemplo: x = fatorial(5)  
    if (n <= 1):  
        return 1  
    return N * fatorial(N-1)
```



Alguma dúvida?

Prof. Anilton

Bloco B – Sala 133

anilton.ufu@outlook.com

anilton@ufu.br