

CC1612

Fundamentos de Algoritmos

Prof. Danilo H. Perico

Variáveis

Variáveis

Uma variável é um nome que se refere a um dado ou valor

Variáveis

Uma variável é um recurso utilizado nos programas para escrever e ler dados da memória do computador!

Variáveis

São simplesmente espaços na memória o qual reservamos e damos nomes.

Variáveis - Tipos de Dados

- **Dados** são as entidades mais fundamentais que um programa manipula!
- Os dados podem ser de diferentes **tipos**

Variáveis - Tipos de Dados

- Números:
 - Inteiro (*int*): 1 ; 2 ; -3 ; 0 ; 10
 - Real (*float*): 1.3 ; -3.63 ; 7.2 ; 16.42
 - Complexo (*complex*): $6 + 3j$; $-2 + 4j$
- Texto (*string*): “Olá” ; “Isto é uma string”
- Tipo Lógico (*bool*): True ; False

Variáveis - Tipos de Dados

- Para descobrir qual é o tipo de determinado dado, podemos utilizar a função `type()` no Python.

```
In [1]: type(3.6)
```

```
Out[1]: float
```

```
In [2]: type(2)
```

```
Out[2]: int
```

```
In [3]: type("olá")
```

```
Out[3]: str
```

```
In [4]: type("2.8")
```

```
Out[4]: str
```

```
In [5]: type(2+6j)
```

```
Out[5]: complex
```


Variáveis - Tipos de Dados - Aritmética com *float*

- Como já sabemos, o computador trabalha com números binários
- Porém, a maioria das frações decimais não pode ser representada exatamente como frações binárias.
- Uma consequência é que, em geral, os números decimais de ponto flutuante que você digita acabam sendo armazenados de forma aproximada.

Variáveis - Tipos de Dados - Aritmética com *float*

- Assim, podemos ter diferenças na aritmética realizada com float
- Exemplo:

```
1 a = 0.1
2 b = 0.2
3
4 c = a+b
5
6 print(c)
```

0.30000000000000004

- Quase todas as máquinas atuais (julho de 2010) usam aritmética de ponto flutuante conforme a norma IEEE-754, que limita a quantidade de casas decimais

Variáveis - Nomes

- No Python os nomes das variáveis começam, obrigatoriamente, com uma letra
- Porém, o nome completo da variável pode conter números e o símbolo sublinhado (_)
- Em programação, normalmente não se utiliza acentos nos nomes das variáveis, porém o Python 3 permite esta utilização
- O Python diferencia letras maiúsculas de letras minúsculas

Variáveis - Nomes - Exemplos

nome	válido	Comentários
a1	sim	Começa com letra
velocidade	sim	letras
velocidade90	sim	Letras e números
salário_médio	sim	Letras e _
salário médio	não	Não pode conter espaços
_b	sim	_ é aceito no início do nome
1a	não	Não pode iniciar com números

Variáveis - Nomes - Palavras Reservadas

- Os programadores geralmente escolhem os nomes das suas variáveis com base no que a variável vai armazenar
- No entanto, algumas palavras são reservadas em Python e não podem ser nomes de variáveis. São elas:

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

Variáveis - Atribuição de Valores

- Para armazenar valores nas variáveis, utilizamos o símbolo de igual (=).
- Esta operação é chamada de **atribuição**, pois um valor é atribuído a uma variável
- Exemplo:

```
a = 5
b = 6
print(a + b)

11
```

- **a** recebe o valor 5
- **b** recebe o valor 6
- Imprimi o resultado da soma de **a** com **b**

Variáveis - Conversão de Valores

- A conversão de valores é feita por meio das funções:

- `int()`
- `float()`
- `str()`

- Exemplos:

```
In [9]: a = "3"  
        b = int(a)  
        type(b)
```

```
Out[9]: int
```

```
In [10]: a = 3  
         b = str(a)  
         type(b)
```

```
Out[10]: str
```

Saída de dados

Função *print()*

- Conforme já vimos, podemos imprimir *strings* ou o valor de variáveis com os seguintes comandos:

```
print("Olá mundo!")
```

```
Olá mundo!
```

```
a = 4  
print(a)
```

```
4
```

Função *print()*

- Para a impressão de dados na tela, podemos também combinar texto e o valor de alguma variável, utilizando vírgula (,) entre um dado e outro:

```
a = 3  
print("a vale", a)  
a vale 3
```

Função *print()* - composição

- Combinar várias *strings* com o valor de variáveis sem sempre é prático com o uso da vírgula.
- Por isso podemos usar composição de *strings* para combinar texto com o valor de variáveis.
- Exemplo:

```
anos = 30  
print("João tem %d anos" % anos)
```

```
João tem 30 anos
```

Função *print()* - composição

```
anos = 30
print("João tem %d anos" % anos)

João tem 30 anos
```

- O símbolo *%d* é chamado de **marcador**.
- O marcador indica que naquela posição estaremos colocando o valor da variável *anos*, que deve ser um número inteiro neste exemplo.

Função *print()* - composição - marcadores

marcador	tipo
%d	Número inteiro (int)
%f	Números decimais (float)
%s	Strings

Função *print()* - composição - marcadores

- Exemplos:

```
a = "ola"  
dia = 16  
mes = 8  
print("%s hoje é dia %d do mês %d" % (a,dia,mes))
```

ola hoje é dia 16 do mês 8

```
pi = 3.141592  
print("0 pi vale %f" % pi)
```

0 pi vale 3.141592

Função *print()* - composição - marcadores

- Limitando a quantidade de casas decimais na impressão:
 - Basta incluir *.x* entre o *%* e o *f*, onde x é o número desejado de casas decimais
 - Exemplo:

```
pi = 3.141592
print("0 pi vale %.2f" % pi)

0 pi vale 3.14
```

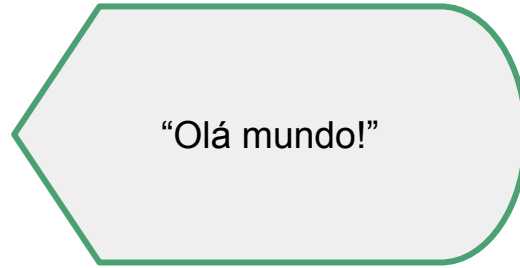
Função *print()* - *format()*

- O método *format()* formata a *string* dada de acordo com o desejado para a saída de dados
 - Exemplo:

```
1 name = "Fulano"
2
3 peso = 78.51
4
5 print("0 {0} pesa {1:2.1f} Kg".format(name,peso))
```

0 Fulano pesa 78.5 Kg

Símbolo no fluxograma - saída de dados



Pseudocódigo

ESCREVA “Olá mundo!”

ou

WRITE “Olá mundo!”

Entrada de datos

Função *input()*

- A entrada de dados é feita pela função *input()*
- *input()* aceita como parâmetro uma mensagem a ser exibida
- O valor recebido pela entrada de dados deve ser atribuído a uma variável
- Todo valor recebido pela função *input()* tem sempre o tipo *string*
- Se a ideia é utilizar o valor recebido em alguma conta ou cálculo, ele deve ser convertido para algum tipo numérico

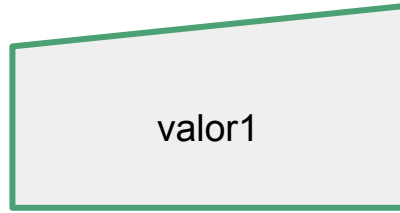
Função *input()*

- Exemplo:
 - O valor digitado para pi será recebido como string; antes de ser atribuído a variável pi, ele é convertido em float:

```
pi = float(input("Digite o valor de pi: "))  
print("O valor digitado é %.1f" % pi)
```

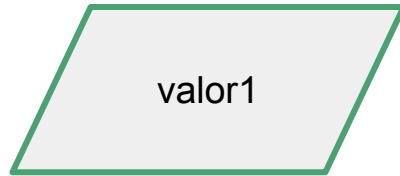
```
Digite o valor de pi: 3.14159  
O valor digitado é 3.1
```

Símbolo no fluxograma - entrada de dados



Entrada via teclado

Entrada genérica



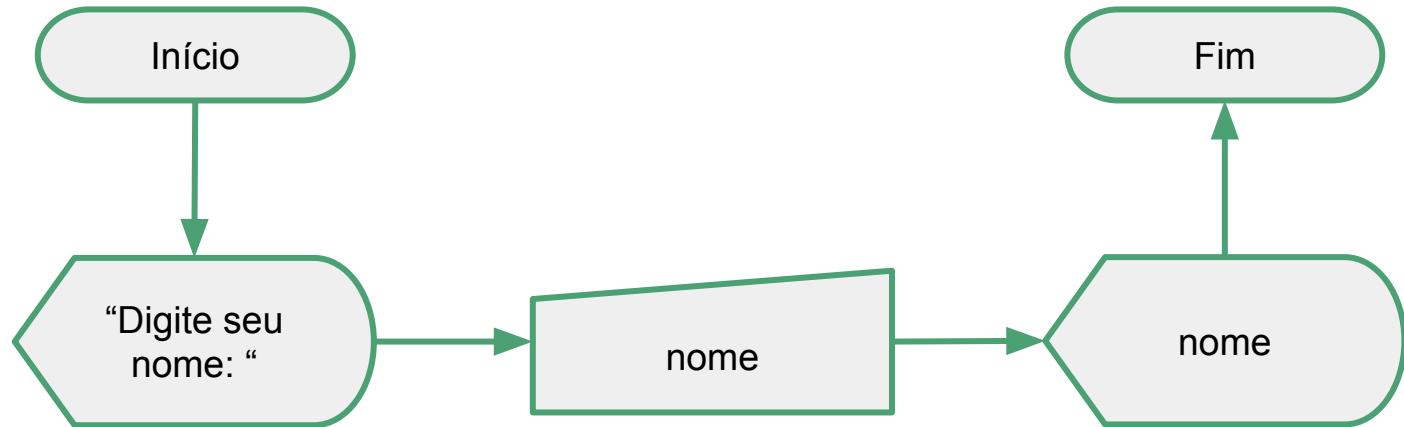
Pseudocódigo

LEIA valor1

ou

READ valor1

Exemplo Fluxograma



Exemplo Pseudocódigo

INÍCIO

ESCREVA “Digite seu nome: “

LEIA nome

ESCREVA nome

FIM

Exercícios

1. Faça um programa que solicita do usuário uma quantidade de dias, horas, minutos e segundos. Calcule e imprima o total convertido em somente segundos.
2. Escreva um programa que pergunte a quantidade de km percorridos por um carro alugado, assim como a quantidade de dias pelos quais o carro foi alugado. Calcule o preço a pagar, sabendo que o carro custa R\$ 60,00 por dia e R\$ 0,15 por km rodado.

Exercícios

3. Tendo como dados de entrada a altura de uma pessoa, construa um algoritmo que calcule seu peso ideal, usando a seguinte fórmula: $(72.7 * \text{altura}) - 58$

Exercícios

4. Faça um Programa que pergunte quanto você ganha por hora e o número de horas trabalhadas no mês. Calcule e mostre o total do seu salário no referido mês, sabendo-se que são descontados 11% para o Imposto de Renda, 8% para o INSS e 5% para o sindicato, faça um programa que nos dê:
- salário bruto.
 - quanto pagou ao INSS
 - quanto pagou ao sindicato
 - o salário líquido

Calcule os descontos e o salário líquido e exiba conforme a tabela abaixo:

+ Salário Bruto : R\$
- IR (11%) : R\$
- INSS (8%) : R\$
- Sindicato (5%) : R\$
= Salário Líquido : R\$

Obs.: Salário Bruto - Descontos = Salário Líquido.

Operadores Aritméticos

Operadores Aritméticos

+	Soma
-	Subtração
*	Multiplicação
/	Divisão
//	Divisão com resultado inteiro
%	Módulo: retorna o resto da divisão
**	Potência: $x ** y$ (x elevado a y)

Operadores Aritméticos - Exemplos

```
a = 4
b = 5
c = a+b
print("Soma: ")
print(c)
c = a-b
print("Subtração: ")
print(c)
c = a*b
print("Multiplicação: ")
print(c)
c = b/a
print("Divisão: ")
print(c)
```

Soma:
9
Subtração:
-1
Multiplicação:
20
Divisão:
1.25

```
a = 4
b = 5
c = b//a
print("Divisão (resultado parte inteira): ")
print(c)
c = a**b
print("Potência: ")
print(c)
c = b%a
print("Módulo: ")
print(c)
```

Divisão (resultado parte inteira):
1
Potência:
1024
Módulo:
1

Função Matemáticas

Algumas das funções matemáticas mais usadas

- É necessário importar o módulo de matemática
- *from math import **

abs(x)	Valor absoluto de x
sqrt(x)	Raiz quadrada de x
log(x)	Retorna o logaritmo natural de x
log10(x)	Retorna o logaritmo base-10 de x
sin(x)	Retorna o seno de x radianos
cos(x)	Retorna o cosseno de x radianos
exp(x)	Retorna e^{**x}
round(x, n)	Número x arredondado para n dígitos

Exemplo

```
from math import *  
  
print(sqrt(9))  
print(log(3))
```

3.0

1.0986122886681098

Exercícios

Represente as seguintes equações no padrão do Python:

5. $y = x^3 - |\ln(x)|$

5. *Teste: $x = 3$ $y = 25.901387$*

6. $y = x^2 - w^k$

6. *Teste: $x = 3, w = 3, k = 3$ $y = -18$*

7. $k = \frac{e^2 + \sqrt{x+y}}{3}$

7. *Teste: $x = 3, y = 3$ $k = 3.279512$*

8. $y = \frac{x^2 - \log(x) + \text{sen}(x)}{x - \cos^3(x)}$

8. *Teste: $x = 3$ $y = 2.182215$*

Operadores Relacionais

Operadores Relacionais

- Operadores relacionais são utilizados para se realizar comparações entre valores;
- Estes valores podem ou não estar armazenados em variáveis.
- O resultado de toda comparação é um tipo lógico: **True** ou **False**

Operadores Relacionais

operador	operação	Símbolo matemático
==	Igualdade	=
>	Maior que	>
<	Menor que	<
!=	diferente	≠
>=	Maior ou igual	≥
<=	Menor ou igual	≤

Exemplos

```
In [26]: 5 > 9
```

```
Out[26]: False
```

```
In [27]: 9 == 9
```

```
Out[27]: True
```

```
In [28]: 7 > 3
```

```
Out[28]: True
```

```
In [29]: 4 != 6
```

```
Out[29]: True
```

```
In [30]: 2 >= 2
```

```
Out[30]: True
```

```
In [32]: a = 10  
b = -5
```

```
b >= a
```

```
Out[32]: False
```

```
In [33]: b <= a
```

```
Out[33]: True
```

```
In [34]: b != a
```

```
Out[34]: True
```

```
In [35]: b == a
```

```
Out[35]: False
```

Exercício

9. Teste cada comparação no Python e coloque True ou False na tabela ao lado dados os valores das variáveis

- `a = 4`
- `b = 10`
- `c = 5.0`
- `d = 1`
- `f = 5`

<code>a == c</code>	
<code>a < b</code>	
<code>d < b</code>	
<code>c != f</code>	
<code>a == b</code>	
<code>c < d</code>	
<code>b > a</code>	
<code>c >= f</code>	
<code>f >= c</code>	
<code>c <= c</code>	
<code>c <= f</code>	

Estrutura Condicional

Estrutura Condicional

- Nem sempre todas as linhas do código devem ser executadas!
- Normalmente, o programa deve decidir quais partes serão executadas com base em uma ou mais condições
- As condições são construídas com operadores relacionais: são feitas com base no resultado de comparações!

Comando **if**

- Em Python, e em várias outras linguagens de programação, o comando principal para a realização de decisões é o **if**
- Sintaxe do **if** no Python:

```
if <condição>:  
    bloco verdadeiro
```

- **If** nada mais é do que nosso **se**
- Em português, podemos entender o comando if da seguinte forma:
 - **Se a condição for verdadeira, faça alguma coisa**

Comando **if** - Exemplo

Ler dois valores e apresentar o maior deles:

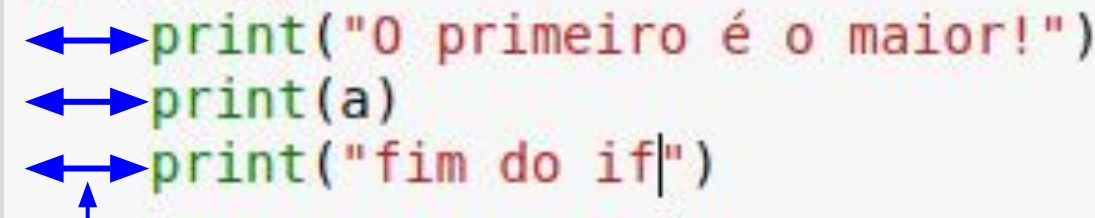
```
a = int(input("Primeiro Valor: "))  
b = int(input("Segundo Valor: "))  
  
if a > b:  
    print("O primeiro é o maior!")  
  
if b > a:  
    print("O segundo é o maior!")
```

```
Primeiro Valor: 87  
Segundo Valor: 54  
O primeiro é o maior!
```

Comando **if** - indentação

- O bloco que será executado se a condição do if for verdadeira fica **indentado** com relação ao comando if
- **Indentação** é o **recuo** (deslocamento do texto à direita)
- **Indentação**: neologismo derivado da palavra em inglês *indentation*

```
if a > b:  
    print("O primeiro é o maior!")  
    print(a)  
    print("fim do if")
```




indentação

Comando **if** - indentação - Exemplo

```
a = int(input("Primeiro Valor: "))
b = int(input("Segundo Valor: "))

if a > b:
    print("O primeiro é o maior!")
    print(a)
    print("fim do if")

print("Este print() executa de forma independente com relação à condição a > b")
```



Estes são os comandos que pertencem ao bloco do if

Comando **if** - Exemplo

Voltando ao exemplo do maior entre dois números, o que acontece se os valores digitados para os dois números forem iguais???

```
a = int(input("Primeiro Valor: "))
b = int(input("Segundo Valor: "))

if a > b:
    print("O primeiro é o maior!")

if b > a:
    print("O segundo é o maior!")
```

Comando **if** - Exemplo

Voltando ao exemplo do maior entre dois números, o que acontece se os valores digitados para os dois números forem iguais???

```
a = int(input("Primeiro Valor: "))
b = int(input("Segundo Valor: "))

if a > b:
    print("O primeiro é o maior!")

if b > a:
    print("O segundo é o maior!")
```

Primeiro Valor: 45
Segundo Valor: 45

Nenhuma das duas condições será verdadeira!

Nenhum print() será chamado!

Exercícios

10. Faça um Programa que peça dois números e, então, imprima o maior deles.
11. Faça um Programa que peça um valor numérico e escreva na tela se o valor é “positivo” ou “negativo”.
12. Faça um Programa que leia um número e exiba o dia correspondente da semana: 1 - Domingo, 2 - Segunda etc.
13. Escreva um programa que pergunte o salário de um funcionário e calcule o valor do aumento. Para salários superiores a R\$ 1250,00, calcule um aumento de 10%. Para inferiores ou iguais, de 15%. Imprima o novo salário.
14. Escreva um programa que deve dizer se um carro é novo (menos que 3 anos) ou velho (mais ou igual a 3 anos). A idade do carro deve ser informada pelo usuário.

Exercícios

15. Escreva um programa que pergunte a distância que um passageiro deseja percorrer em km. Calcule o preço da passagem, cobrando R\$ 0,50 por km para viagens até 200 km e R\$ 0,45 para viagens mais longas.
16. Faça um Programa para leitura de três notas parciais de um aluno. O programa deve calcular a média ($M = (N1 + N2*2 + N3*3)/6$) alcançada pelo aluno e apresentar:
 - a. A mensagem "Aprovado", se a média for maior ou igual a 5, com a respectiva média alcançada;
 - b. A mensagem "Reprovado", se a média for menor do que 5, com a respectiva média alcançada;

Comando **else**

- O comando **else** (senão) é utilizado nos casos em que a segunda condição é simplesmente o **contrário** da primeira.
- Sempre utilizado como uma sequência de um **if**
- Sintaxe:

```
if <condição>:  
    bloco verdadeiro  
else:  
    bloco contrário
```