

# Variáveis, Expressões e Instruções

Disciplina: Programação para Engenharia

Professores:

Cristiane Pavei Martinello Fernandes

Douglas de Medeiros Deolindo

Giovani Martins Cascaes

Joel Barbosa Panchyniak

Marcelo Marcos Amoroso

Marcos Antônio Jeremias Coelho

Ramon de Souza Coan

# Exercícios

1) A fórmula dos juros compostos é a seguinte:

$$\mathbf{FV = PV.(1+i)^n}$$

Onde:

FV - Valor final do investimento, ao término do tempo

PV - Valor inicial que o cliente irá investir

i - Rentabilidade mensal (em porcentagem)

n - Quantidade de meses que o dinheiro do cliente vai ficar aplicado

Elabore um Script em Python que solicite o valor do investimento (PV), o número de meses (n) que irá permanecer aplicado e a rentabilidade (i). Ao final, o script deve mostrar o valor do montante total (FV).

# Exercícios

2) Ainda sobre juros compostos: Um cliente pediu que o sistema do banco tivesse a seguinte função:

Informar o valor inicial que ele deve investir, para ao final de 'n' meses ele tenha um valor 'fv', supondo que este dinheiro esteja com uma rentabilidade 'i' mensal, em porcentagem.

Desenvolva um script em Python que solicite o valor final, o número de meses que retende deixar seu dinheiro aplicado, bem como a rentabilidade. O script deve exibir o valor inicial que ele deve investir para atingir tal objetivo.

$$PV = \frac{FV}{(1 + i)^n}$$

# Exercícios

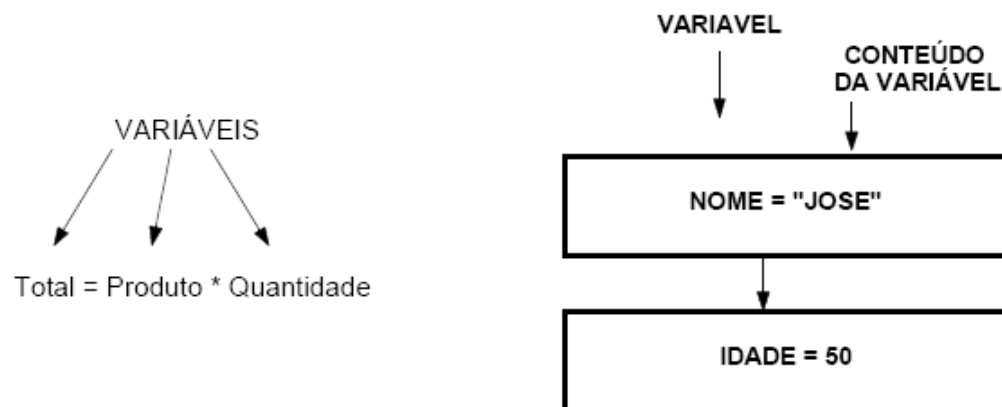
3) Suponha que o preço de capa de um livro seja R\$ 24,95, mas as livrarias recebem um desconto de 35%. O transporte custa R\$ 3,00 para o primeiro exemplar e 75 centavos para cada exemplar adicional. Qual é o custo total de atacado para X cópias? Solicite o valor de X. Crie um Script em linguagem Python para solicitar os dados necessários e exibir o custo total da compra.

# O que são variáveis?

Em algoritmos de computação é necessário armazenar informações diversas;

Cada bloco de memória que armazena um dado recebe uma etiqueta, um nome e essas estruturas são chamadas de **variáveis do programa**;

Isto é, nomes de controle que existirão no código para armazenar dados.



```
>>> message = 'And now for something completely different'
>>> n = 17
>>> pi = 3.141592653589793
```

# Nomenclatura de variáveis em Python

Os programadores geralmente escolhem nomes significativos para as suas variáveis – eles documentam o uso da variável.

Como a maioria, a linguagem Python é **Case Sensitive**, isto é, diferencia caracteres minúsculos de maiúsculos.

Nome  $\neq$  nome

Idade  $\neq$  idade

# Nomenclatura de variáveis em Python

- Nomes de variáveis podem ser tão longos quanto você queira;
- Podem conter tanto letras como números, porém não pode se iniciar com números;
- A convenção é usar apenas letras minúsculas para nomes de variáveis;
- O caractere de sublinhar ( `_` ) pode aparecer em um nome.
- Não pode conter caracteres especiais, ex: `@,/,#`

# Palavras reservadas em Python

- O interpretador usa palavras-chave para reconhecer a estrutura do programa e elas NÃO podem ser usadas como nomes de variável.

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

Você não precisa memorizar essa lista. Na maior parte dos ambientes de desenvolvimento, as palavras chave são exibidas em uma cor diferente; se você tentar usar uma como nome de variável, vai perceber.



# Palavras reservadas em Python

- Se você der um nome ilegal a uma variável, recebe um erro de sintaxe:

```
>>> 76trombones = 'big parade'  
SyntaxError: invalid syntax
```

É ilegal porque começa com um número.

```
>>> more@ = 1000000  
SyntaxError: invalid syntax
```

É ilegal porque contém um caractere ilegal, o @.

```
>>> class = 'Advanced Theoretical Zymurgy'  
SyntaxError: invalid syntax
```

É uma das palavras reservadas

# Padrões de escrita ...

- camelCase

nomeUsuario = 'Guido'



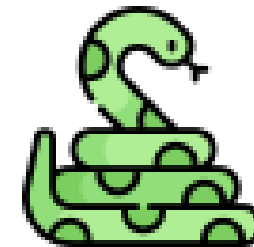
- PascalCase

NomeUsuario = 'Guido'



- snake\_case

nome\_usuario = 'Guido'



# Tipos básicos de variáveis em Python

Em Python, os tipos de dados básicos são:

- Tipo **int** (armazena números inteiros);
- Tipo **float** (armazena números em formato decimal);
- Tipo **string** (armazena um conjunto de caracteres);
- Tipo **bool** (armazena booleanos – **True** ou **False**).

Cada variável pode armazenar apenas um tipo de dado a cada instante.

# Tipos básicos de variáveis em Python

Em Python, diferentemente de outras linguagens de programação, não é preciso declarar de que tipo será cada variável no início do programa. Quando se faz uma atribuição de valor, automaticamente a variável se torna do tipo do valor armazenado.

```
>>> a = 10
>>> a
10
```

```
>>> c = "Olá Mundo"
>>> c
'Olá Mundo'
```

```
>>> b = 1.2
>>> b
1.2
```

A variável **a** se torna uma variável do tipo **int**

A variável **c** se torna uma variável do tipo **string**

A variável **b** se torna uma variável do tipo **float**

# Tipagem forte

Uma vez que uma variável tenha um valor de um tipo, ele não pode ser usado como se fosse de outro tipo.

```
a = 10
```

```
b = '20'
```

```
c = a + b
```

b é uma **string** (texto), e portanto não pode ser somada a um inteiro

```
Traceback (most recent call last):
```

```
File "<stdin>", line 1, in <module>
```

```
TypeError: unsupported operand type(s) for  
+: 'int' and 'str'
```

# Função: `print()`

A função `print()` serve para imprimir os argumentos passados a ela no terminal ou na tela.

Por exemplo, o script

```
print(1)
x = 2
print(x)
```

produz a saída

```
1
2
```

# Função: `print()`

A função `print()` serve para imprimir os argumentos passados à ela no terminal ou na tela.

```
print(object(s), sep = separator, end = end)
```

- **object(s)** = Qualquer objeto e quantos você quiser. Será convertido em string antes de ser impresso;
- **sep** = 'separator' [Opcional]. Especifique como separar os objetos, se houver mais de um;
- **end** = 'end' [Opcional]. Especifique o que imprimir no final. O padrão é '\n' (avanço de linha)

# Função: `print()` – Exemplos

```
nome = "Guido Van Rossum"  
msg = "Meu nome é: "
```

```
print(msg, nome)  
Meu nome é: Guido Van Rossum
```

```
print("Meu nome é:", nome)  
Meu nome é: Guido Van Rossum
```

```
>>> print(dia + '/' + mes + '/' + ano)  
06/nov/2001
```

```
>>> dia = '06'  
>>> mes = 'nov'  
>>> ano = '2001'  
>>> print(dia, mes, ano, sep='/')  
06/nov/2001
```



# Função: `print()`

- Formatação para a função `print()`. É possível especificar uma máscara no comando `print` para imprimir números com um determinado formato.

`print("%.2f" % variável)`

- **f** é usado para números do tipo **float**
- **d** é usado para números **int**
- **s** é usado para **string**

```
print('A area do triangulo %s de altura %.0f e base %.0f  
    eh: %.2f' % (tipo, altura, base, area))
```

# Praticando com o Python ...

1) Elabore um script em linguagem Python que exiba na tela seu nome completo, sua cidade, estado e data de nascimento.

Para cada dado, crie uma variável apropriada. Use nomes que façam sentido ('cidade', 'estado', etc. Evite criar variáveis com nomes 'a', 'b', 'x' ou 'y' - isso é um péssimo hábito entre programadores).

2) Crie um script em linguagem Python que exiba na tela o texto 'A melhor banda do mundo é [nome da banda] e a melhor música é [nome da música]'.

O nome da banda e o nome da música devem estar declarados em duas variáveis diferentes.

# Tipos de Erros

- Erro de sintaxe
  - Falha na tradução do algoritmo para Python;
  - O compilador vai detectar e dar dicas;
  - Mais fáceis de corrigir.
- Erro de lógica
  - Resultados diferentes do esperado;
  - Erro de projeto do algoritmo;
  - Mais difíceis de corrigir.



**BUG?**

# Variáveis numéricas – Tipo `int` (Inteiro)

Nesse tipo os dados são interpretados a fim de determinar o sinal e o valor, assumindo valores negativos, positivos e o zero, exatamente como no conjunto dos números inteiros.

Exemplo de valores válidos: `1, 100, -50, 36, 0`

Exemplo de variáveis:

```
idade = 20  
altura = 176  
num = -3
```

# Variáveis numéricas – Tipo float (Flutuante)

O tipo ponto flutuante é responsável por representar o conjunto dos números reais.

Pode-se também representar inteiros, mas não tantos quanto o tipo inteiro, pois a capacidade de armazenamento deste tipo está direcionada para poder suportar valores decimais.

Exemplo de valores válidos: 0.5, 1.0, -0.723, 0.0

Exemplo de variáveis:

```
peso = 72.7  
saldo_conta = 105.79  
temperatura = -1.7
```

# Função: input()

- O comando **input()** é utilizado para obtermos uma entrada de dados, ou seja, para que o usuário forneça algo ao programa pelo terminal.

```
nome = input("Entre com o seu nome: ")
Entre com o seu nome: Fulano da Silva
>>> nome
'Fulano da Silva'
```

```
num = int(input("Entre com um numero? :"))
Entre com um numero? :100
>>> num
100
```

```
altura = float(input("Entre com a sua altura? :"))
Entre com a sua altura? :1.80
>>> altura
1.8
```

# Função: `input()`

É importante destacar que a função **`input()`** sempre retornará o valor recebido no formato de uma **string**, mesmo que ele seja de um tipo de dado diferente como, por exemplo, um número real. Vamos testar?

```
aluno = input("Digite seu nome: ")
idade = input("Qual a sua idade: ")
nota = input("Qual sua expectativa de nota para a prova: ")
regular = input("Você é aluno regular (True) ou não (False): ")

print(type(aluno))
print(type(idade))
print(type(nota))
print(type(regular))
```

# Função: **type()**

Para simplificar o que foi mencionado, vamos apresentar alguns exemplos utilizando a função **type()**, conforme programa anterior. **type()** é uma função que recebe como entrada um valor, uma variável ou um objeto (por enquanto, vamos deixar o conceito de objeto para outro momento) e retorna a qual tipo o valor (ou a variável) pertence.



# Função: `input()`

Embora a intenção do usuário tenha sido digitar um número inteiro, um número real e um valor lógico, respectivamente. Como contornar esse problema? Fácil! Basta usar as funções de conversão **`int()`**, **`float()`** e **`bool()`**, respectivamente.

```
aluno = input("Digite seu nome: ")
idade = int(input("Qual a sua idade: "))
nota = float(input("Sua expectativa de nota para a prova: "))
regular = bool(input("Você é aluno regular(True) ou (False):"))

print(type(aluno))
print(type(idade))
print(type(nota))
print(type(regular))
```

# Comentários em Python

- Comentários são trechos do programa voltados para a leitura por humanos, e ignorados pelo interpretador;
- Documentar seu código e fazer com que ele seja fácil de entender por outras pessoas;
- Começam com o símbolo #;
- Tudo na linha após # é ignorado pelo interpretador.

# Praticando com o Python ...

1) Escreva um programa que receba 2 valores de x e y. Em seguida calcule e imprima o valor de z:

$$z = \frac{(x^2 + y^2)}{x - y}$$

2) Construa um programa que receba do usuário a variação do deslocamento de um objeto (em metros) e a variação do tempo percorrido (em segundo). Ao fim, o programa deve calcular a velocidade média, em m/s, do objeto. Mostrar os dados fornecidos e o valor calculado.

# Praticando com o Python ...

3) Escreva um programa que pede o raio de um círculo, e em seguida exiba o perímetro e área do círculo. Para saber o valor do pi, faça:

```
import math  
print(math.pi)
```

4) Você está no Brasil, e para temperatura usamos o grau Celsius. Porém, quando você for contrato para trabalhar como programador Python no exterior, deverá usar graus Fahrenheit.

Ou seja, você fornece a temperatura em graus Celsius, e seu script faz a conversão para graus Fahrenheit.

# Praticando com o Python ...

5) Agora faça o contrário. Você fornece a temperatura em graus Fahrenheit, seu programa converte para Celsius e exibe na tela.

6) Um novo modelo de carro, super econômico foi lançado. Ele faz 20 km com 1 litro de combustível. Cada litro de combustível custa R\$ 4,95.

Faça um programa que pergunte ao usuário quanto de dinheiro ele pretende usar e em seguida o programa informa quantos litros de combustível ele pode comprar e quantos quilômetros o carro consegue rodar com esta quantidade de combustível.

Seu script será usado no computador de bordo do carro.

# Praticando com o Python ...

7) Sua tarefa é criar um programa em Python que pede o preço original de um produto e dá 20% de desconto. Você deve mostrar:

Preço original do produto

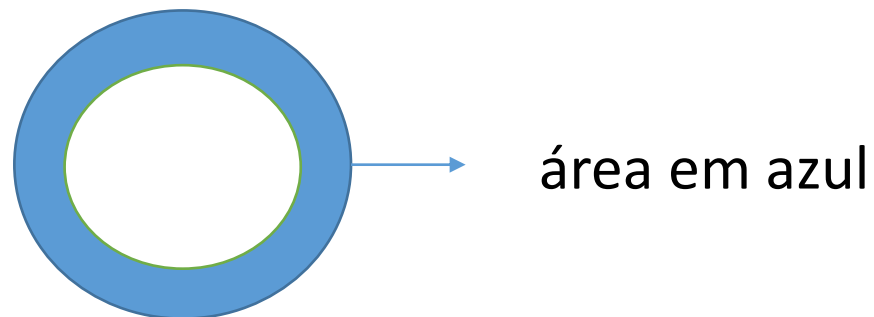
Valor do desconto em R\$ (tipo 'Você ganhou R\$ xx,xx de desconto')

Valor do produto com o desconto

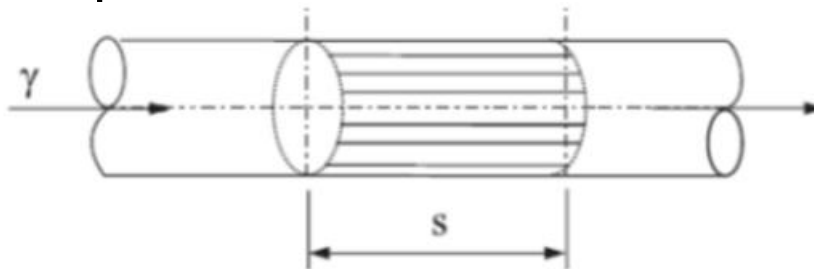
8) A loja percebeu que não quer dar 20% em tudo. Quer dar 20% em algumas coisas, 10% em outras, nada em outros produtos e até 30% em alguns outros produtos. Crie um script em Python que pergunte o preço original e o desconto que deve ser concedido. Ele deve mostrar a saída igual ao exercício anterior.

# Praticando com o Python ...

9) Você foi contratado para desenvolver um programa que calcule área de uma coroa circular com base em duas medidas de raio fornecido pelo usuário.



10) Você está desenvolvendo um programa para calcular a vazão de um fluido em um tubo com base no diâmetro interno do tubo e na velocidade do fluxo. A fórmula para calcular a vazão deve ser pesquisada. Os dados de entrada devem ser alimentados em metros e m/s.

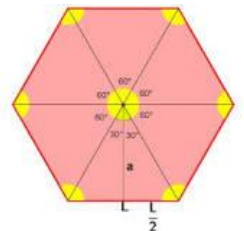


# Praticando com o Python ...

11) Você foi contratado para desenvolver um programa que calcule o Índice de Massa Corporal (IMC) com base nos dados de altura e peso fornecidos pelo usuário. O IMC é uma medida que relaciona o peso e a altura de uma pessoa para avaliar se ela está abaixo do peso, com peso normal, com sobrepeso ou obesa.

A fórmula para calcular o IMC é:  $IMC = \text{peso} / (\text{altura}^2)$ , onde o peso é em quilogramas e a altura é em metros.

12) Você está desenvolvendo um programa para calcular a área de um hexágono regular com base no raio fornecido pelo usuário. Um hexágono regular tem seis lados de igual comprimento e seis ângulos internos de 120 graus. Assim, para determinar a área desse hexágono, basta determinar a área de um dos triângulos e, em seguida, multiplicar o resultado por 6.





Obrigado!

