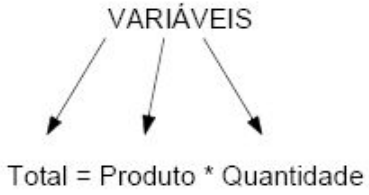
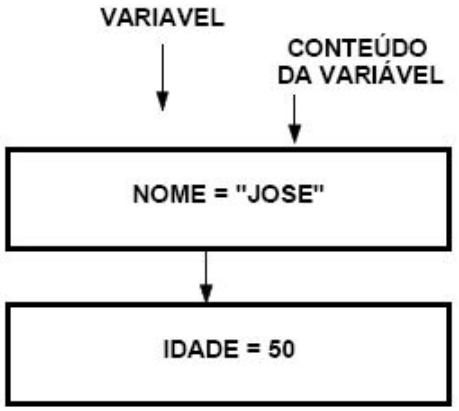
Aula -3 - Variáveis, expressões e instruções

Prof. Rodrigo Maciel

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 na maioria das linguagens, o Python é **Case Sensitive**, isto é, diferencia caracteres minúsculos de maiúsculos.

Nome ≠ nome

Idade ≠ idade

Nomenclatura de variáveis em Python

- Nomes de variáveis podem ser tão longos quanto você queira;
- Podem conter tanto letras quanto 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

Tipos básicos de variáveis em Python

Em Python, os tipos de dados básicos são:

- Tipo **inteiro** (armazena números inteiros). **Ex**: -3,-2,-1,0,1,2,3
- Tipo **float** (armazena números em formato decimal). **Ex**: -2.2, 1.7
- Tipo **string** (armazena um conjunto de caracteres). Ex: “olá”
- Tipo **bool** (armazena booleanos). Ex: **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
```

A variável **a** se torna uma variável do tipo **inteiro**.

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

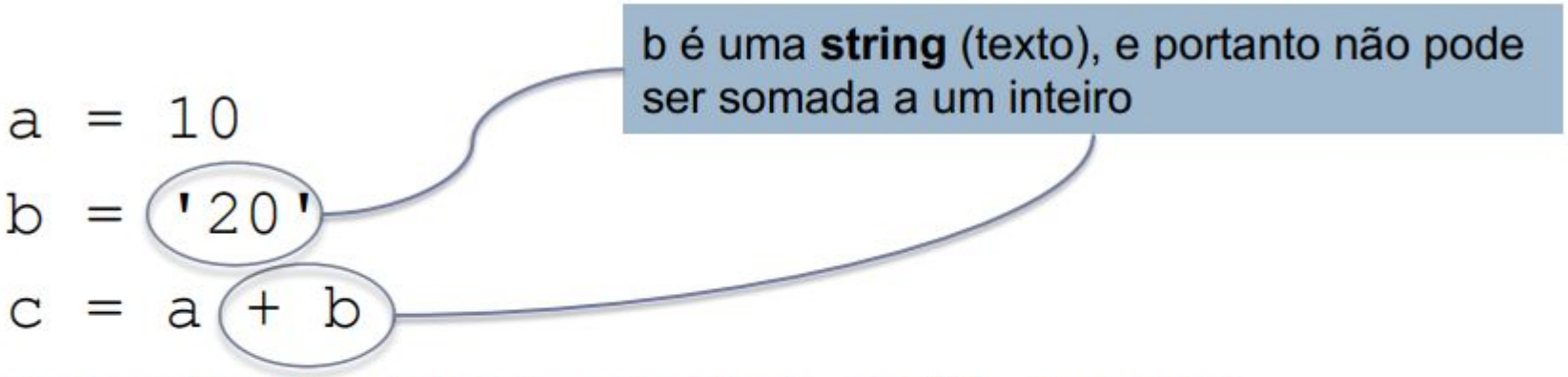
A variável **c** se torna uma variável do tipo **string**.

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

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

Tipagem forte e dinâmica

- 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
```

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 a ela no terminal ou na tela.

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

- 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;

Funções print() – Exemplos

```
[1]  nome = 'Rodrigo Maciel'
     frase = 'Meu nome é:'
     print(frase,nome)
```

```
Meu nome é: Rodrigo Maciel
```

```
[2]  dia = '24'
     mes = 'fev'
     ano = '2022'
     print(dia,mes,ano,sep='/')
```

```
24/fev/2022
```

```
[3]  print('Meu nome é: ',nome)
```

```
Meu nome é:  Rodrigo Maciel
```

```
[5]  print(dia + '/' + mes + '/' + ano)
```

```
24/fev/2022
```

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 do tipo **inteiro**
- **s** é usado para **strings**

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

Função: print()

```
tipo = 'equiátero'
altura = 1.55
base = 2.27
area = 3.3333
print('A área do triângulo %s de altura %.2f e base %.2f é: %.4f m²' % (tipo, altura, base, area))
```

```
A área do triângulo equiátero de altura 1.55 e base 2.27 é: 3.3333 m²
```

Funçãoinput()

- O comando input() é utilizado para obtermos uma entrada de dados, ou seja, é para que o usuário escreva algo no 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
```

Tipos de Erros

- Erros de sintaxe (escrita):
  - Falha na tradução do algoritmo
  - O compilador vai detectar e dar dicas
  - Mais fáceis de corrigir
- Erros de lógica
  - Resultados diferentes do esperado
  - Erro de projeto do algoritmo
  - Mais difíceis de corrigir

BUG?

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

```
# Filtra o dataframe pela sample
dataframeFilter = dataframe[dataframe['sample']==sample]
# Lista de espectrograma de cada sensor para aquela sample
listSensorSpectrogram = []
```

Praticando Python ...

- Escreva um programa que receba 2 valores do tipo inteiro x e y, calcule e imprima o valor de z:

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

- Faça um Programa que leia 4 notas, mostre as notas e a média na tela.