



Universidade Federal de Viçosa
Campus Rio Paranaíba
Instituto de Ciências Exatas e Tecnológicas

SIN 141

Computação

Orientada a Objetos

Sistemas de Informação
Alan Diêgo Aurélio Carneiro
alan.carneiro@ufv.br



Universidade Federal de Viçosa
Campus Rio Paranaíba
Instituto de Ciências Exatas e Tecnológicas

Aula de Hoje

Introdução à linguagem Python

Introdução ao Python

Python é uma linguagem dinamicamente tipada. Isso quer dizer que não é necessário tipar as variáveis para usa-las. Seu tipo é modificado dependendo do tipo de seu valor.

Para se atribuir o valor 2 para a variável "var" apenas precisamos fazer:

```
>>> var = 2
```

O valor 2 é inteiro, então a variável "var" é uma variável inteira.



Introdução ao Python

Por via das duvidas, para se saber de que tipo é a variável utilizamos a função "type ([variável])".

```
>>> var = 2
```

```
>>> type(var)
```

```
<type 'int'>
```

```
>>>
```



Introdução ao Python

Os principais tipos de variáveis em Python são:

- inteiros (int)
- floats (float)
- strings (str)
- listas (list)
- tuplas (tuple)
- dicionários (dic)

Veremos mais detalhadamente a seguir.



A função print()

A função para imprimir dados em Python é a função print(). Ela é responsável por mostrar valores em seu terminal:

```
1 print('Olá Mundo')
```

Por padrão, quando utilizamos virgula para separar os itens, a função print utiliza espaços para separar cada saída.

Porém, podemos utilizar o parâmetro sep= para definir um caractere de separação.

Entenda no exemplo a seguir:

```
1 print('Dia', 'Mês', 'Ano', sep='/')
2 print('ontem', 'Hoje', 'Amanhã', sep='-')
3 print("B", "n", "n", ".", sep='a')
```

A função print()

Por padrão, a função print utiliza a quebra de linha (\n) como último caracter.

O parâmetro end= é responsável por alterar esse comportamento. Vamos entender melhor no exemplo a seguir:

```
1  # Exemplo com fim de linha sem nenhum caracter
2  print('Vamos estudar Na ', end='')
3  print('Python Academy')
4
5  # Exemplo com fim de linha igual à ->
6  print('As rosas são', end=' -> ')
7  print('Vermelhas')
8
9  # Exemplo com fim de linha igual à :
10 print("Quantidade", end=': ')
11 print(40)
```

A função input()

A função `input()` recebe como parâmetro uma string que será mostrada como auxílio ao usuário, geralmente o informando que tipo de dado o programa está aguardando receber.

Vejo um exemplo:

```
1 nome = input("Escreva seu nome: ")
2
3 print('Seu nome é:', nome)
```


Operadores Aritméticos

Operador	Nome	Função
+	Adição	Realiza a soma de ambos operandos.
-	Subtração	Realiza a subtração de ambos operandos.
*	Multiplicação	Realiza a multiplicação de ambos operandos.
/	Divisão	Realiza a Divisão de ambos operandos.
//	Divisão inteira	Realiza a divisão entre operandos e a parte decimal de ambos operandos.
%	Módulo	Retorna o resto da divisão de ambos operandos.
**	Exponenciação	Retorna o resultado da elevação da potência pelo outro.



Operadores Aritméticos

```
1  quatro = 4
2  dois = 2
3
4  soma = quatro + dois
5  print(soma)  # Resultado: 6
6
7  subtracao = quatro - dois
8  print(subtracao)  # Resultado: 2
9
10 multiplicacao = quatro * dois
11 print(multiplicacao)  # Resultado: 8
12
13 divisao = quatro / dois
14 print(divisao)  # Resultado: 2.0
15
16 divisao_interna = quatro // dois
17 print(divisao_interna)  # Resultado: 2
18
19 modulo = quatro % dois
20 print(modulo)  # Resultado: 0
21
22 exponenciacao = quatro ** dois
23 print(exponenciacao)  # Resultado: 16
```

01

Operadores de Comparação

Operador	Nome	Função
==	Igual a	Verifica se um valor é igual ao outro
!=	Diferente de	Verifica se um valor é diferente ao outro
>	Maior que	Verifica se um valor é maior que outro
>=	Maior ou igual	Verifica se um valor é maior ou igual ao outro
<	Menor que	Verifica se um valor é menor que outro
<=	Menor ou igual	Verifica se um valor é menor ou igual ao outro



Operadores de Comparação

```
1  var = 5
2
3  if var == 5:
4      print('Os valores são iguais')
5
6  if var != 7:
7      print('O valor não é igual a 7')
8
9  if var > 2:
10     print('O valor da variável é maior de 2')
11
12 if var >= 5:
13     print('O valor da variável é maior ou igual a 5')
14
15 if var < 7:
16     print('O valor da variável é menor que 7')
17
18 if var <= 5:
19     print('O valor da variável é menor ou igual a 5')
```

Operadores

Atribuição

Operador	Equivalente a
=	$x = 1$
+=	$x = x + 1$
-=	$x = x - 1$
*=	$x = x * 1$
/=	$x = x / 1$
%=	$x = x \% 1$

Lógico

Operador	Definição
and	Retorna True se ambas as afirmações forem verdadeiras
or	Retorna True se uma das afirmações for verdadeira
not	retorna Falso se o resultado for verdadeiro



Operadores Lógicos

03

```
1  num1 = 7
2  num2 = 4
3
4  # Exemplo and
5  if num1 > 3 and num2 < 8:
6      print("As Duas condições são verdadeiras")
7
8  # Exemplo or
9  if num1 > 4 or num2 <= 8:
10     print("Uma ou duas das condições são verdadeiras")
11
12 # Exemplo not
13 if not (num1 < 30 and num2 < 8):
14     print('Inverte o resultado da condição entre os parênteses')
```

Operadores de Identidade

Operador	Definição
is	Retorna True se ambas as variáveis são o mesmo objeto
is not	Retorna True se ambas as variáveis não forem o mesmo objeto

```
1  lista = [1, 2, 3]
2  outra_lista = [1, 2, 3]
3  recebe_lista = lista
4
5  # Recebe True, pois são o mesmo objeto
6  print(f"São o mesmo objeto? {lista is recebe_lista}")
7
8  # Retorna False, pois são objetos diferentes
9  print(f"São o mesmo objeto? {lista is outra_lista}")
```

04

Muitas vezes programadores Python ficam na dúvida em quando utilizar o operador de igualdade == ou o operador de identidade is.

Mas agora que você já conhece os dois sabe que o operador == verifica os valores testados, enquanto o operador is testa a referência dos valores testados!

Operadores de Associação

Operador	Função
in	Retorna True caso o valor seja encontrado na sequência
not in	Retorna True caso o valor não seja encontrado na sequência

```
1  lista = ["Python", 'Academy', "Operadores", 'Condições']
2
3  # Verifica se existe a string dentro da lista
4  print('Python' in lista)  # Saída: True
5
6  # Verifica se não existe a string dentro da lista
7  print('SQL' not in lista) # Saída: True
```


Estrutura Condicional if

Agora que já está craque com os Operadores do Python, vamos aprender sobre nossa primeira estrutura condicional.

Sua sintaxe é bem simples, bastando utilizarmos `if` seguido pela condição seguido por dois pontos:

Veja os exemplos a seguir:

```
1 valor = 10
2
3 if valor > 5:
4     print('O valor é maior que 5.')
```

```
1 if True:
2     print("Este bloco sempre irá ser executado.")
```

Estrutura Condicional if/else

O else é utilizado quando ao testar a condição do if ela é falsa, um fluxo alternativo deve ser executado dispensando testes adicionais.

Exemplo:

```
1  idade = 20
2
3  if idade < 17:
4      print('A idade é MENOR que 17')
5  else:
6      print('A idade é MAIOR que 17')
```

Estrutura Condicional if-elif-else

O elif é utilizado quando mais de uma condição if precisa ser testada. Exemplo:

```
1  linguagem = "Python"
2
3  if linguagem == "C++":
4      print('C++ é uma linguagem de programação compilada.')
5  elif linguagem == "Python":
6      print("Python é uma linguagem de programação de alto nível")
7  elif linguagem == "Java":
8      print("Java é uma linguagem de programação amplamente utilizada no mercado")
9  else:
10     print('Não é nenhuma das duas opções')
```

Estrutura Condicional match e case

Da versão 3.10 em diante, o Python implementou um recurso de switch case chamado “correspondência de padrões estruturais”. Você pode implementar esse recurso com as palavras-chave match e case.

```
1  lang = input("What's the programming language you want to learn? ")
2
3  match lang:
4      case "JavaScript":
5          print("You can become a web developer.")
6
7      case "Python":
8          print("You can become a Data Scientist")
9
10     case "PHP":
11         print("You can become a backend developer")
12
13     case "Solidity":
14         print("You can become a Blockchain developer")
15
16     case "Java":
17         print("You can become a mobile app developer")
18     case _:
19         print("The language doesn't matter, what matters is solving problems.")
```

05

Funções em Python

Como definir uma função em Python:

Def nome_funcao(lista_parametros):

bloco de instruções

return retorno_opcional

```
1 | def hello(meu_nome):  
2 |     print('Olá', meu_nome)
```

```
1 | def calcular_pagamento(qtd_horas, valor_hora):  
2 |     horas = float(qtd_horas)  
3 |     taxa = float(valor_hora)  
4 |     if horas <= 40:  
5 |         salario=horas*taxa  
6 |     else:  
7 |         h_excd = horas - 40  
8 |         salario = 40*taxa+(h_excd*(1.5*taxa))  
9 |     return salario
```

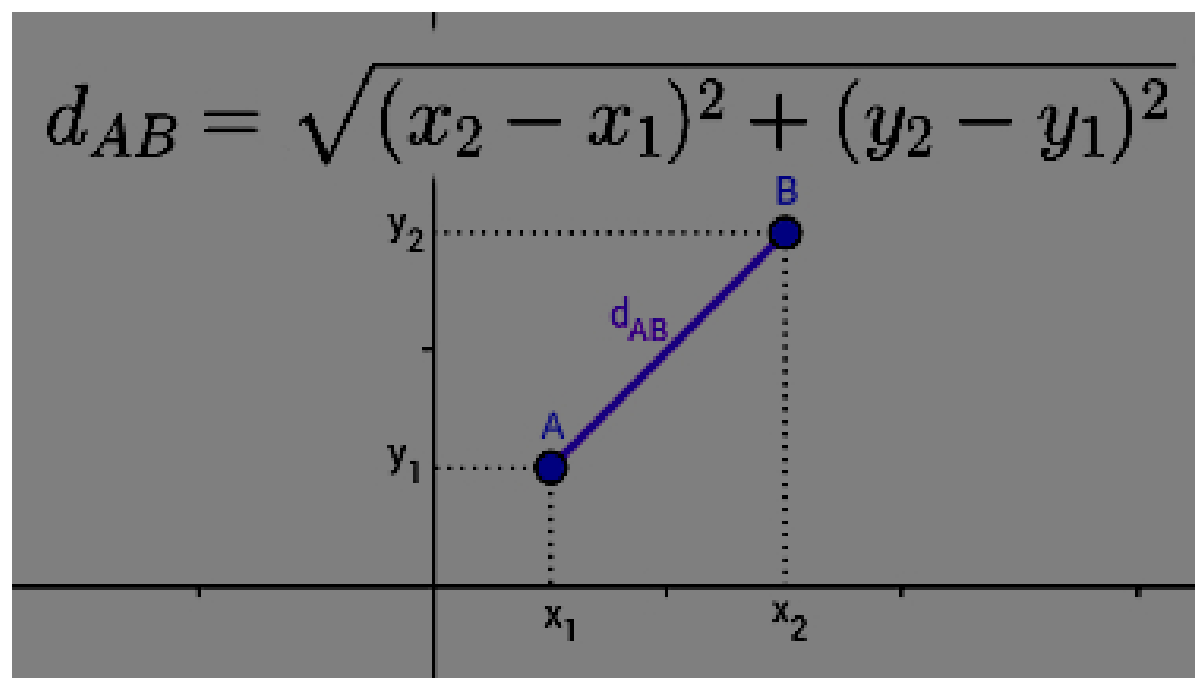
Exercício

Escreva funções que dados dois números x e y , retorna:

1. soma \rightarrow soma(2,3) = $2 + 3 = 5$
2. subtração \rightarrow subt(7,4) = $7 - 4 = 3$
3. divisão \rightarrow divis(8,2) = $8 / 2 = 4$
4. produto \rightarrow prod(3,5) = $3 * 5 = 15$

Exercício

Seja a relação de **distância euclidiana** entre dois pontos **cartesianos**, representada por uma reta entre eles:



Introdução ao Python

Exercício:

Implemente uma função que de como retorno a distância entre dois pontos.

Represente cinco pontos quaisquer de um plano cartesiano, imprima-os e por fim, mostre a distância entre cada um deles.



Referência Bibliográfica

- HARVEY, M. D.; DEITEL, P.J.
Python: como programar. 5. ed.
São Paulo: Pearson Prentice Hall, 2006.
[Exemplares disponíveis: 30]



UFV - Campus Rio Paranaíba
Sistemas de Informação