

Paradigmas de Programação

Prof. Maicon R. Zatelli

Python - Uma Linguagem Multiparadigma

Universidade Federal de Santa Catarina

Florianópolis - Brasil

2018/1

Python

Desenvolvida por Guido van Rossum (1991)

Multiparadigma:

- Orientado a Objetos
- Imperativo
- Funcional
- Procedural (estruturado)
- Reflectivo (capacidade de um programa inspecionar e modificar a si próprio durante a execução)

Interpretada

Multiplataforma

Software livre e código aberto

<https://www.python.org/>

Python - Variáveis

Tipagem dinâmica

- Interpretador infere o tipo da variável

```
n, m = 2, 5  
print("Soma: ", n + m)
```

* Tipagem estática: C, Java

Tipagem forte

- Interpretador avalia as expressões e não faz conversões automáticas entre tipos não compatíveis

```
x, y = 1, "Floripa"  
print("Soma: ", x + y)
```

* Tipagem fraca: PHP e JavaScript

Tente executar os códigos Python em:

http://www.compileonline.com/execute_python_online.php

Python - Sintaxe

```
if x >= 100 or x > 15 and x < 25:  
    print "a"  
elif not (x < 10):  
    print "b"  
else:  
    print "c"
```

Python - Sintaxe

```
i = 0
while i >= 10:
    print i
    i += 2
```

Python - Sintaxe

```
for i in xrange(10):  
    print i
```

Python - Sintaxe

```
vetor = [None] * n #cria um vetor preenchido com ‘nada’ de n posições
for i in xrange(n): #inicia em 0 e vai até n - 1 (igual em C/Java)
    vetor[i] = [] #cria uma lista para cada posição do vetor

i = 0
while vetor[i] != []: #testa se lista da posição i do vetor não é vazia
    ...

for k in vetor[i]: #para cada elemento k da lista da posição i do vetor
    ...
```

Python

Vamos praticar um pouco ... (ver lista de problemas)

Entrada de Dados (Python - Unicamp)

```
# -*- coding: utf-8 -*-
```

```
10
20
*
A
```

```
a = int(input())
b = int(input())
c = input()
d = input()
```

```
10 * 34
```

```
a, b, c = raw_input().split()
a = int(a)
c = int(c)
```

```
10 20 30 40 50 0
```

```
lista = raw_input().split()
for i in xrange(len(lista)):
    lista[i] = int(lista[i])
```

Entrada de Datos (Python - SPOJ)

```
# -*- coding: utf-8 -*-
```

```
10
20
*
A
```

```
a = int(input())
b = int(input())
c = input()
d = input()
```

```
10 * 34
```

```
import sys
a, b, c = sys.stdin.readline().split()
a = int(a)
c = int(c)
```

```
10 20 30 40 50 0
```

```
import sys
lista = sys.stdin.readline().split()
for i in xrange(len(lista)):
    lista[i] = int(lista[i])
```

Exemplo 1

10 30 40 50 10 2 4

```
import sys
lista = sys.stdin.readline().split()
x = 0
for i in xrange(len(lista)):
    x += int(lista[i])
print x
```

Exemplo 2

10 30 40 50 10 2 4

```
import sys
lista = sys.stdin.readline().split()
x = 0
for i in lista:
    x += int(i)
print x
```

Exemplo 3

10,30,40,50,10,2,4

```
import sys
lista = sys.stdin.readline().split(",")
x = 0
for i in lista:
    x += int(i)
print x
```

Exemplo 4

4 / 2

5 * 2

```
import sys
a,b,c = sys.stdin.readline().split()
a = int(a)
c = int(c)
if b == "/":
    print a / c
else:
    print a * c
```

Exemplo 5

4 / 2 = 2

50 * 2 = 8

```
import sys
a,b,c,d,e = sys.stdin.readline().split()
a = int(a)
c = int(c)
e = int(e)
if b == "/":
    resposta = a / c
else:
    resposta = a * c
if resposta == e:
    print "OK"
else:
    print "Erro"
```

Atividade

Pesquise sobre programação orientada a objetos em Python e construa um pequeno tutorial, com exemplos, que deve constar, ao menos, os itens abaixo:

- Declaração de classes (métodos, atributos, construtor);
- Instanciação de classes;
- Herança (simples e múltipla) e polimorfismo;
- Métodos abstratos e estáticos;