

INSTITUTO FEDERAL
BAIANO

Lógica e Linguagem de Programação

Python

Prof. José Honorato Ferreira Nunes

honoratonunes@softwarelivre.org

<http://softwarelivre.org/zenorato>



Resumo da aula

- Por que estudar Python?
- Usando o Interpretador de Python
- Variáveis e Tipos de dados em Python
- Operadores e Expressões em Python
- Comandos de Entrada e Saída em Python

Por que estudar Python?

- Como existe muitas linguagens diferentes, por que aprender *Python*?
- Os conceitos fundamentais da linguagem são simples de entender;
- A sintaxe do *Python* é clara e fácil de aprender;
- Os tipos pré-definidos em *Python* são poderosos e simples de usar;
- O interpretador *Python* permite aprender e testar rapidamente trechos de código
- *Python* é expressivo, com abstrações de alto nível => código curto e rápido desenvolvimento;

Por que estudar Python?

- É fácil escrever extensões para Python em C e C++, quando for necessário desempenho máximo, ou quando necessitar de interfacear alguma ferramenta nestas linguagens;
- Python permite que o programa execute em múltiplas plataformas, sem alterações;
- Possui tratamento de exceções (moderno mecanismo de tratamento de erros);
- Python é orientado a objetos (incluindo herança Múltiplas);
- Python é livre.

Usando o Interpretador de Python

Classificação das linguagens:

- compilada
- interpretada

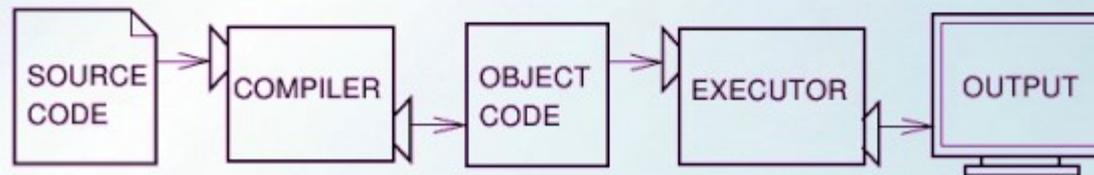
Compiladas:

Fortran, C, C++, Visual Basic, ...

```
#include <iostream>
using namespace std;

int main ()
{
    cout << "Hello World! \n";
    return 0;
}
```

```
$ gcc hello.c -o hello
$ ./hello
Hello World
```



Usando o Interpretador de Python

Interpretadas:

Python, Perl, Basic tradicional, Shell Script, ...

```
$ python hello.py  
Hello World
```



Usando o Interpretador de Python

Ao contrário das linguagens compiladas, que transformam o código escrito dos programas para uma plataforma específica, por exemplo, Windows ou Linux, Python transforma o código do programa em bytecodes e estes são executados por um interpretador, o que possibilita o aplicativo ser executado em várias plataformas com poucas ou mesmo nenhuma alteração.

Usando o Interpretador de Python

A maioria das distribuições Linux já vem com o interpretador do Python instalado. Para a plataforma Windows há um instalador, que inclui o interpretador, a documentação e um ambiente de desenvolvimento integrado (IDE) – o IDLE.

O instalador do Python no Windows inclui o IDLE, cujo nome é tanto uma homenagem a Eric Idle, quanto abreviação de Integrated DeveLopment Editor, e possui recursos como coloração de sintaxe (syntax highlighting), autocompletar (autocomplete) e um debugger.

Variáveis e Tipos de dados em Python

Identificadores são nomes utilizados para identificar objetos - variáveis, funções e classes, por exemplo. Os identificadores devem começar com uma letra sem acentuação ou com um sublinhado (`_`), e podem conter números, letras sem acentuação e sublinhados.

Python é case-sensitive, ou seja, o identificador `python` é diferente dos identificadores `PYTHON` e `Python`.

Variáveis e Tipos de dados em Python

Variáveis são espaços reservados na memória utilizados para armazenar valores, como por exemplo, textos, resultados de cálculos, entrada de usuário, resultados de consultas a uma base de dados, etc. Variáveis devem seguir as regras de identificadores, vistos anteriormente.

Variáveis e Tipos de dados em Python

Tipos de dados restringem os valores a serem armazenados nas variáveis. Os tipos pré-definidos no Python podem ser simples, e também os tipos que funcionam como coleções, como as listas e dicionários.

Python possui tipagem dinâmica, o que associa o valor da variável a um tipo no momento da atribuição de valor e aloca o espaço necessário para armazenar seus valores.

Variáveis e Tipos de dados em Python

O que é Python: Tipagem dinâmica

Python possui o que se chama de **tipagem dinâmica**, ou seja, a tipagem pode mudar a cada nova entrada de dados em uma variável.

```
>>> a = 1
>>> type(a)
<type 'int'>
>>> a = 'abacaxi'
>>> type(a)
<type 'str'>
>>> a = 1.0
>>> type(a)
<type 'float'>
```

A tipagem dinâmica reduz a quantidade de tempo de planejamento prévio e é um mecanismo importante para garantir flexibilidade e simplicidade das funções *Python*.

Variáveis e Tipos de dados em Python

EXEMPLOS DE NOMES VÁLIDOS

nome_candidato

endereco

RG

mes_ferias

dataNasc

fone1

EXEMPLOS DE NOMES INVÁLIDOS

nome candidato

endereço

R.G.

mês_férias

data-Nasc

1fone

Operadores e Expressões em Python

Python Básico: Operadores - aritméticos

```
>>> 7 + 3          # adição
10
>>> 7 - 3          # subtração
4
>>> 8 % 3          # resto da divisão
2
>>> 8 / 3          # divisão inteira
2
>>> 8 / 3.         # divisão em ponto flutuante
2.6666666666666665
>>> 8 * 3          # produto
24
>>> 8 ** 2         # exponenciação
64
```

Operadores e Expressões em Python

O interpretador interativo do Python também pode ser utilizado como uma calculadora, digitando os cálculos e recebendo os valores. Exemplos:

```
1 >>> 2 + 2           # Imprime 4
2 >>> 5 / 2           # Imprime 2
3 >>> 10 % 5          # Imprime 0
4 >>> 12.356 / 2      # Imprime 6.1779999999999999
5 >>> 5 * 5           # Imprime 25
6 >>> 2 ** 3          # Imprime 8
7 >>> 5 - 2           # Imprime 3
8 >>> 12.356 // 2     # Imprime 6.0
```

Comando de Saída

Comando de Impressão ***print*** /saída (Escreva).

```
print 'ola mundo'
```

```
print ("Informe o maior valor: ")
```

```
print ("maior valor = ", maior)
```

Comandos de Entrada

Ele é o complemento de `print()` e nos permite ler dados formatados da entrada padrão (teclado). Sua sintaxe:

```
Valor = input('informe um valor positivo : ')
```

ou

```
Aluno = raw_input('Digite o nome: ')
```

Comandos de Entrada e Saída

```
Idade = raw_input("Digite a sua idade : ")  
print("A sua idade é ", Idade)
```

Atividades

Python Básico: O interpretador Python

Mantendo a tradição vamos fazer o *“Hello World”*

```
>>> a = "Hello"  
>>> b = "World"  
>>> print a, b  
Hello World
```

Observe que o comando *print* insere um espaço automaticamente entre as duas variáveis. O mesmo poderia ter sido feito com os comandos abaixo:

```
>>> a = "Hello World"  
>>> print a  
Hello World  
>>> print "Hello World"  
Hello World  
>>> "Hello World"  
'Hello World'
```

Atividades

Python Básico: Criando um programa Python

O mesmo pode ser feito através da criação de um módulo (como são chamados os programas em *Python*). Em um editor de sua escolha escreva:

```
# Primeiro programa Python: hello.py
a = "Hello"
b = "World"
print a, b
```

Salve o programa com o nome *hello.py* e execute-o chamando o interpretador:

```
$ python hello.py
Hello World
$ _
```

Atividades

- Crie um algoritmo para calcular a média de consumo de combustível de um veículo qualquer. O usuário deverá informar: quilometragem inicial, quilometragem final e a quantidade de litros consumida durante a viagem.

Atividades

- Crie um algoritmo para armazenar a matrícula, idade e o sexo (1 - Masculino e 2 - Feminino) de um aluno. O programa deve solicitar que o usuário informe os dados acima e depois mostrar essas informações na tela.

Atividades

- 1. Construa um algoritmo que, tendo como dados de entrada dois pontos quaisquer no plano, $P(x_1, y_1)$ e $P(x_2, y_2)$, escreva a distância entre eles. A fórmula que efetua tal cálculo é:**

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

- 2. Faça um algoritmo que leia o tempo de duração de um evento em uma fábrica expressa em segundos e mostre-o expresso em horas, minutos e segundos.**

Atividades

3. Escreva um algoritmo que leia três números inteiros e positivos (A, B, C) e calcule a seguinte expressão:

$$D = \frac{R + S}{2}, \text{ onde}$$

$$R = (A + B)^2$$

$$S = (B + C)^2$$

4. O custo ao consumidor de um carro novo é a soma do custo de fábrica com a percentagem do distribuidor e dos impostos (aplicados ao custo de fábrica). Supondo que a percentagem do distribuidor seja de 28% e os impostos de 45%, escrever um algoritmo que leia o custo de fábrica de um carro e escreva o custo ao consumidor.

Atividades

5. Um sistema de equações lineares do tipo:

$$\begin{array}{l} ax + by = c \\ dx + ey = f \end{array} \text{ , pode ser resolvido segundo} \\ \text{mostrado abaixo :}$$

$$x = \frac{ce - bf}{ae - bd} \qquad y = \frac{af - cd}{ae - bd}$$

Escreva um algoritmo que lê os coeficientes a , b , c , d , e e f e calcula e mostra os valores de x e y .

Bibliografia

- ❑ MANZANO, Wilson Y. Yamaturni-São Paulo-SP. **Lógica estruturada para programação de computadores**, Ed. Érica 1997 e 2001.
- ❑ MORAES, Celso Roberto. **Estruturas de Dados e Algoritmos**. Ed. Érica, São Paulo
- ❑ LOPES, Anita. **Introdução à programação**. Rio de Janeiro: Campus, 2002.
- ❑ BENEDUZZI, Humberto M. e METZ, João A. **Lógica e Linguagem de Programação – Introdução ao Desenvolvimento de Software (1ª edição)**. Editora do Livro Técnico, 2010.
- ❑ SEBESTA, Robert W. **Conceitos de linguagens de programação**. 9. ed. Porto Alegre: Bookman, 2003.
- ❑ CORMEN, Thomas H. **Algoritmos: teoria e prática**. Rio de Janeiro: Campus, 2002.
- ❑ ASCENCIO, Ana Fernanda Gomes; CAMPOS, Edilene Aparecida Veneruchi de. **Fundamentos da programação de computadores: algoritmos, Pascal e C/C++ e Java**. 2. ed. São Paulo: Pearson Prentice Hall, 2008.
- ❑ ZIVIANI, Nivio. **Projeto de algoritmos com implementações em Pascal e C**. São Paulo: Pioneira Thomson Learning.