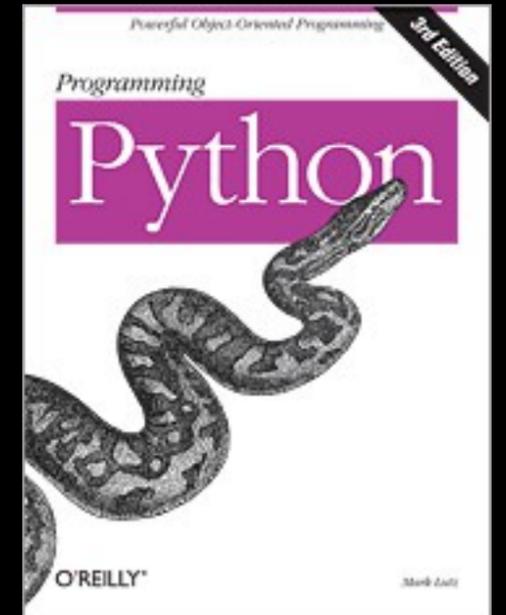


A Linguagem Python: Uma visão geral

Prof. Alberto Costa Neto
Programação em Python

Python é a linguagem do interpretador Python e daqueles que são capazes de conversar com ele. Um indivíduo que “fala” Python é denominado Pythonista. Quase todos os Pythonista conhecidos usam software desenvolvido por Guido van Rossum.



Iniciantes: Syntax Errors

- Precisamos aprender a **linguagem Python** para comunicar nossas instruções ao Python. No início cometeremos muitos erros, como crianças pequenas aprendendo a falar.
- Quando você comete um destes erros, o computador não vai lhe achar “lindo”. Ele irá dizer **“syntax error”** - dado que ele **sabe** a linguagem e você está aprendendo. Parece que o Python é cruel e sem sentimentos.
- Você deve lembrar que **vocês** são inteligentes e **podem** aprender – o computador é simples e rápido – mas não pode aprender – então é **mais fácil para você aprender Python do que para o computador aprender Português.**

Comunicando-se com o Python

```
MacBook-Pro-de-Alberto:~ albertocostaneto$ python
Python 2.7.6 (default, Sep 9 2014, 15:04:36)
[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.39)] on
darwin
Type "help", "copyright", "credits" or "license" for more
information.
```

```
>>>
```

Próximo
Comando?



```
MacBook-Pro-de-Alberto:~ albertocostaneto$ python
Python 2.7.6 (default, Sep  9 2014, 15:04:36)
[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.39)] on
darwin
```

```
Type "help", "copyright", "credits" or "license" for more
information.
```

```
>>> x = 1
>>> print x
1
>>> x = x + 1
>>> print x
2
>>> exit()
```

Este é um bom teste para certificar-se de que você tem o Python instalado corretamente. Note que `quit()` também encerra a seção de interação.

Vamos falar com o Python...

```
Default
dr-chuck2:~ csev$ python
Python 2.6.1 (r261:67515, Jun 24 2010, 21:47:49)
[GCC 4.2.1 (Apple Inc. build 5646)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> print "hello world"
hello world
>>> █
```

```
Administrator: C:\Windows\system32\cmd.exe - C:\Python27\python.exe
Microsoft Windows [Version 6.0.6001]
Copyright (c) 2006 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>C:\Python27\python.exe
Python 2.7.2 (default, Jun 12 2011, 15:08:59) [MSC v.1500 32 bit (Intel)] on win
32
Type "help", "copyright", "credits" or "license" for more information.
>>> print "hello world"
hello world
>>> █
```

O quê diremos?

Elementos de Python

- **Vocabulário / Palavras** – Variáveis e Palavras Reservadas
- **Estrutura de Sentenças** – Padrões de sintaxe válidos
- **Estrutura de Estória** – construindo um programa para um propósito

Palavras Reservadas

- Você não pode usar **palavras reservadas** como nomes de variáveis / identificadores

**and del for is raise assert elif
from lambda return break else
global not try class except if or
while continue exec import
pass yield def finally in print
as with**

Uma “estória” curta sobre como contar palavras em um arquivo com Python

```
nome = raw_input('Nome do arquivo:')
arquivo = open(nome, 'r')
texto = arquivo.read()
palavras = texto.split()

contadores = dict()
for palavra in palavras:
    contadores[palavra] = contadores.get(palavra, 0) + 1
maior_contador = None
palavra_mais_frequente = None

for palavra, contador in contadores.items():
    if maior_contador is None or contador > maior_contador:
        palavra_mais_frequente = palavra
        maior_contador = contador
print palavra_mais_frequente, maior_contador
```

```
python palavras.py
Nome do arquivo: words.txt
to 16
```

Sentenças ou Linhas

x = 2



Comando de Atribuição

x = x + 2



Atribuição com expressão

print x



Comando de impressão

Variável

Operador

Constante

Palavra
Reservada

Programando Parágrafos

Scripts Python

- **Python Interativo** é bom para experimentos e programas de 3 a 4 linhas
- Como a maioria dos programas é bem maior, nós os digitamos em um arquivo e pedimos Python rodar os comandos contidos no arquivo
- De certa forma estamos “**dando um script para o Python**”
- Como uma convenção, damos uma **extensão “.py”** a estes arquivos para indicar que contêm scripts Python

Escrevendo um Programa Simples

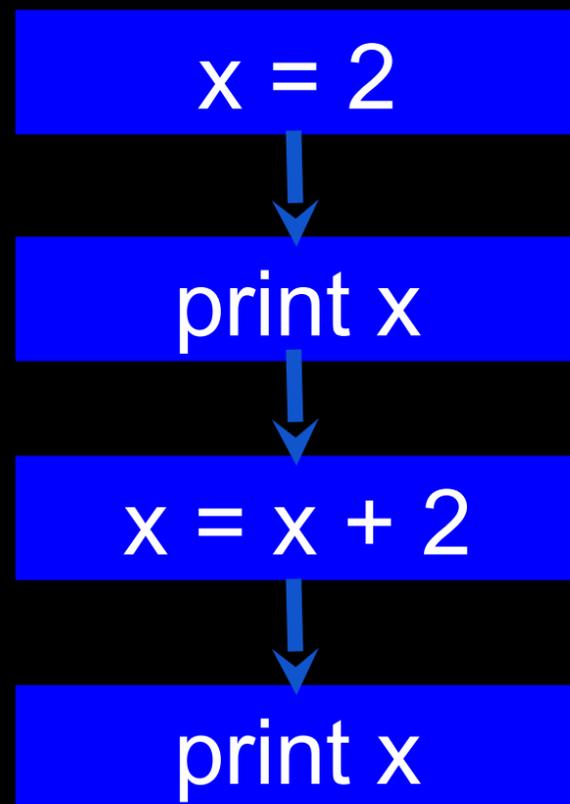
Interativo versus Script

- **Interativo**
 - Você pode digitar diretamente uma linha por vez para o Python que ele responde
- **Script**
 - Você entra com uma seqüência de comandos (linhas) em um arquivo usando um editor de texto e diz ao Python que execute os comandos deste arquivo

Passos de um Programa ou Fluxo de um Programa

- Como uma receita ou instruções de instalação, um programa é uma seqüência de passos a serem executados em ordem
- Alguns passos são condicionais – podem ser pulados
- Às vezes um passo ou grupo de passos precisam ser repetidos
- Algumas vezes armazenamos um conjunto de passos a serem executados várias vezes em muitos pontos de um programa

Passos Seqüenciais



Programa:

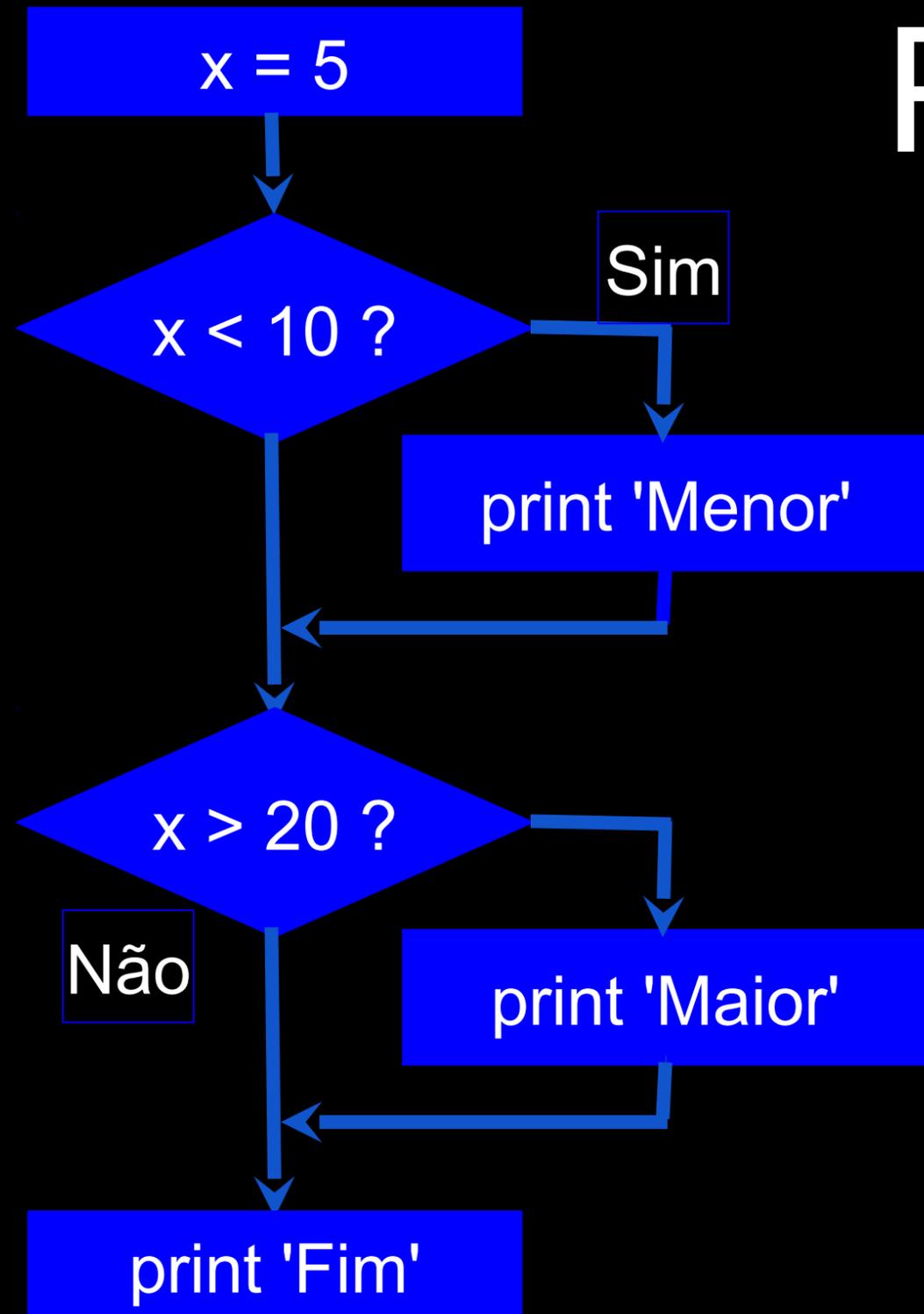
```
x = 2  
print x  
x = x + 2  
print x
```

Saída:

2
4

Quando um programa está executando, ele segue o fluxo de um passo para o próximo. Como programadores, nós ajustamos os passos para o programa seguir

Passos Condicionais



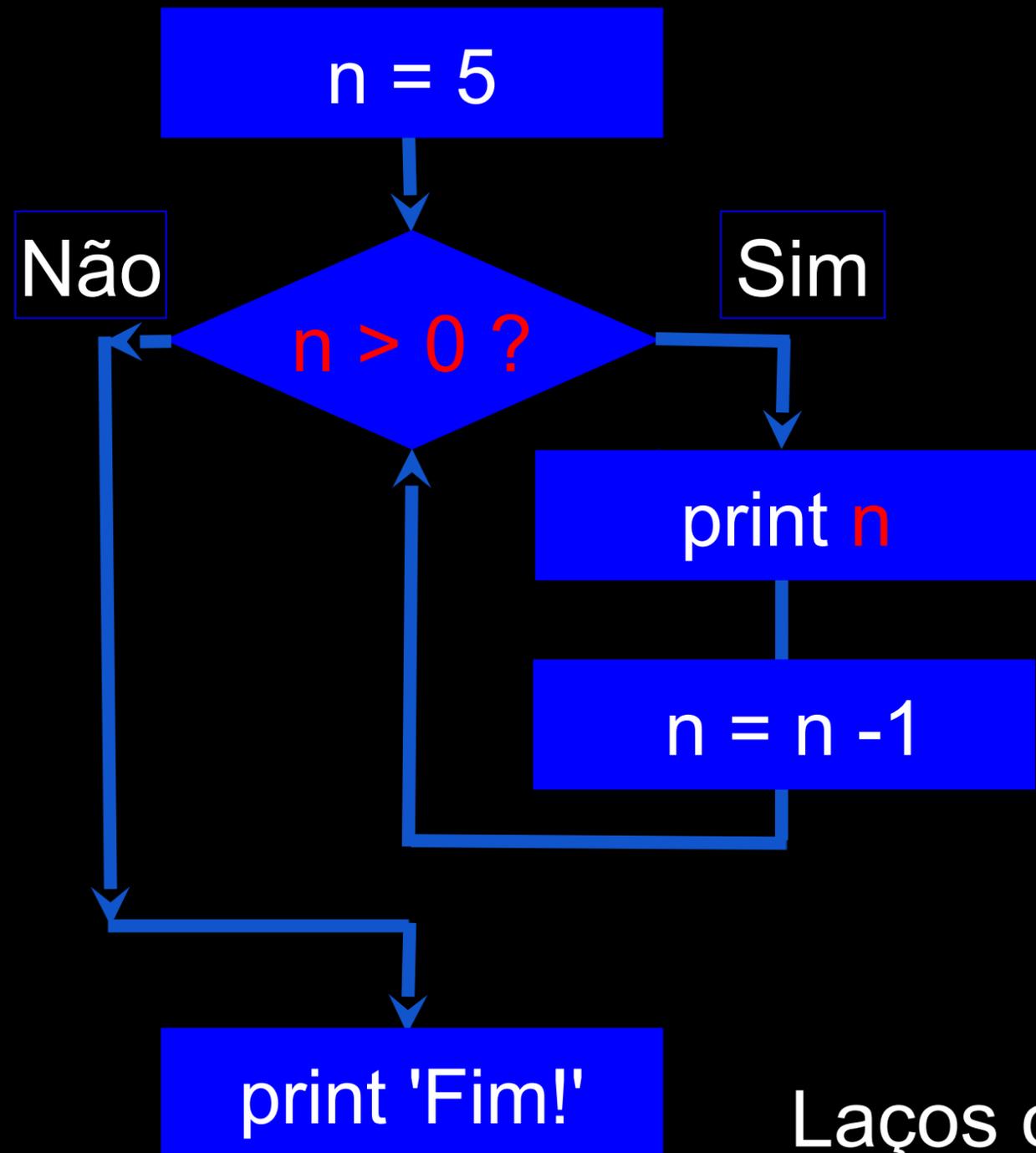
Programa:

```
x = 5
if x < 10:
    print 'Menor'
if x > 20:
    print 'Maior'
print 'Fim'
```

Saída:

Menor
Fim

Passos Repetidos



Programa:

```
n = 5
while n > 0:
    print n
    n = n - 1
print 'Fim!'
```

Saída:

```
5
4
3
2
1
Fim!
```

Laços ou Loops (passos repetidos) têm variáveis de **variáveis de iteração** que mudam a cada passada do laço. Frequentemente estas **variáveis de iteração** recebem uma seqüência de números.

```
nome = raw_input('Nome do arquivo:')
arquivo = open(nome, 'r')
texto = arquivo.read()
palavras = texto.split()
```

Sequencial

Repetido

Condicional

```
contadores = dict()
for palavra in palavras:
    contadores[palavra] = contadores.get(palavra,0) + 1
maior_contador = None
palavra_mais_frequente = None

for palavra,contador in contadores.items():
    if maior_contador is None or contador > maior_contador:
        palavra_mais_frequente = palavra
        maior_contador = contador

print palavra_mais_frequente, maior_contador
```

Uma curta “estória” em Python sobre como contar palavras em um arquivo

Uma palavra usada para ler um dado do usuário

Uma sentença para atualizar um dos muitos contadores

Um parágrafo sobre como encontrar o maior item de uma lista

```
nome = raw_input('Nome do arquivo:')
arquivo = open(nome, 'r')
texto = arquivo.read()
palavras = texto.split()

contadores = dict()
for palavra in palavras:
    contadores[palavra] = contadores.get(palavra,0) + 1
maior_contador = None
palavra_mais_frequente = None

for palavra,contador in contadores.items():
    if maior_contador is None or contador > maior_contador:
        palavra_mais_frequente = palavra
        maior_contador = contador

print palavra_mais_frequente, maior_contador
```



Acknowledgements / Contributions

Agradecimentos / Contribuições



These slides are Copyright 2010- Charles R. Severance (www.dr-chuck.com) of the University of Michigan School of Information and open.umich.edu and made available under a Creative Commons Attribution 4.0 License. Please maintain this last slide in all copies of the document to comply with the attribution requirements of the license. If you make a change, feel free to add your name and organization to the list of contributors on this page as you republish the materials.

Initial Development: Charles Severance, University of Michigan School of Information



These slides were translated and adapted by Alberto Costa Neto (albertocn.sytes.net) of the Federal University of Sergipe