

Introdução à Linguagem de Programação



XVI-Infoeste



Instrutor:
Diogo Branquinho Ramos

O que é Python?

- Criada pelo holandês Guido Van Rossum
- Grupo de comédia Monty Python's Flying Circus.
- Linguagem de mais alto nível
- Interpretada e orientada a objetos
- Não há declaração de variáveis
- O controle de bloco é feito por indentação

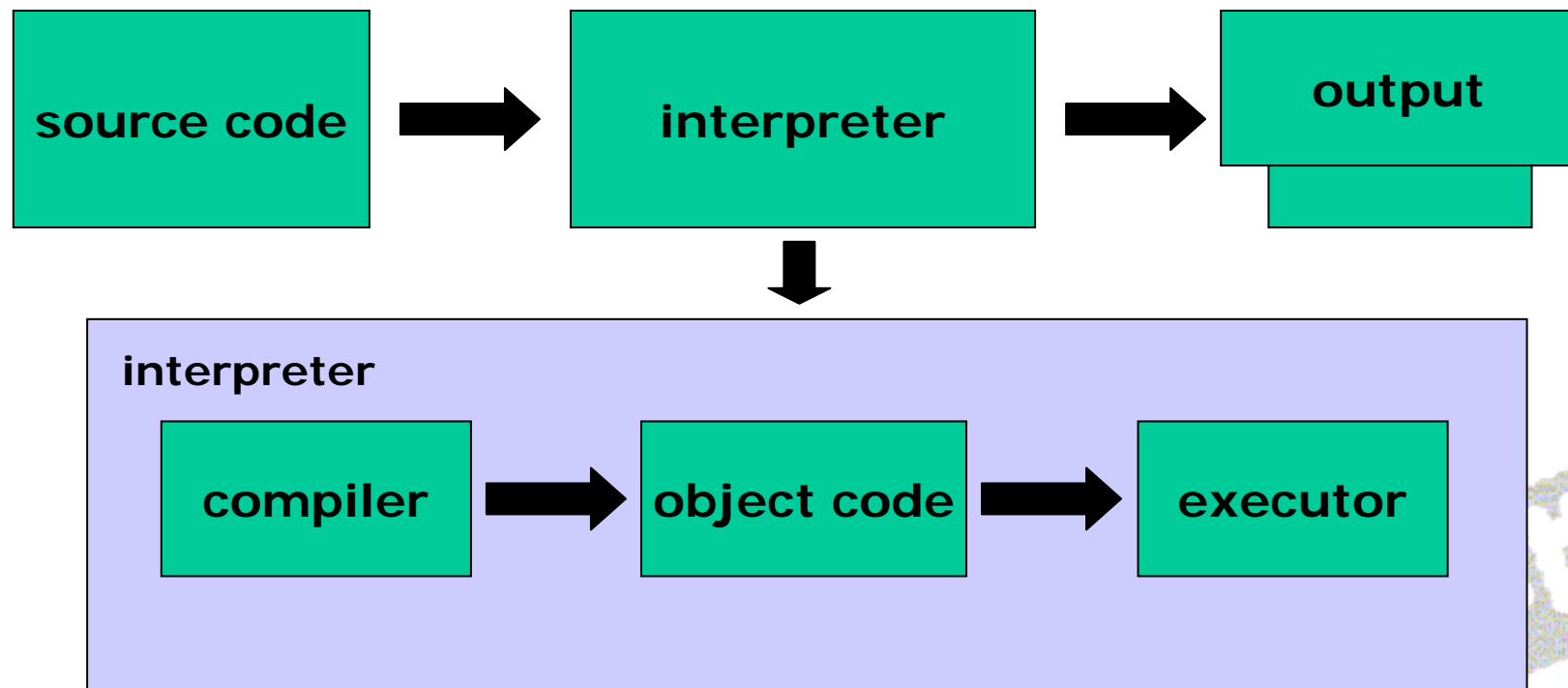


O que é Python?

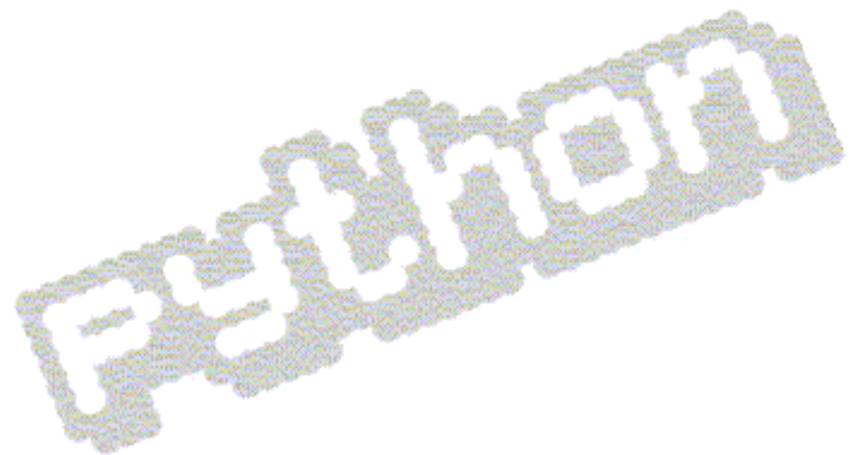
- Possui tipos de variáveis de alto nível
- Sintaxe clara e sólida
- RAD – Rapid Application Development
- Suporta bancos de dados Relacionais e Orientados a Objeto
- É Free (GPL) e não depende de fornecedores exclusivos



Projeto do Compilador



Programando 1ª Parte



Variáveis

- Linha de comando Python
- Variáveis Dinâmicas
 - `a = 2` # Integer Variable
 - `a = 3.14` # Float Variable
 - `a = "Infoeste"` # String Variable
- Nomes de Variáveis
 - não começam por números
 - não possuem caracteres especiais
 - não podem ser palavras reservadas

Palavras Reservadas

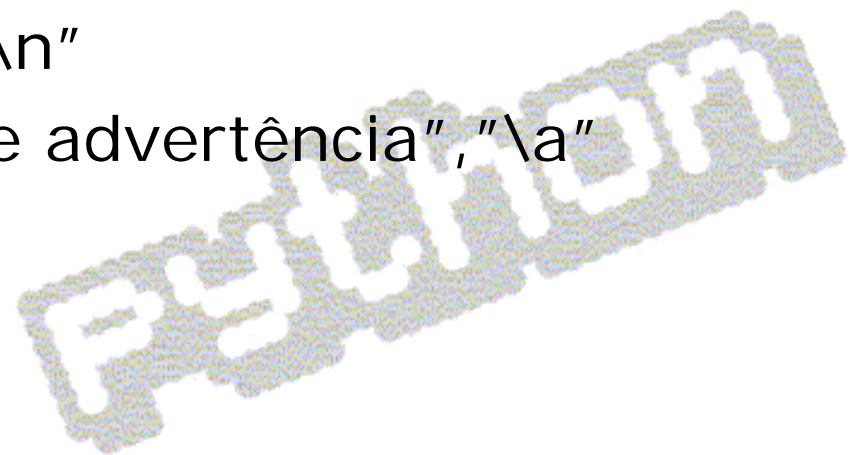
- Python possui 28 palavras reservadas

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

- Qualquer linha precedida de # não é considerada pelo compilador, portanto, comentário.

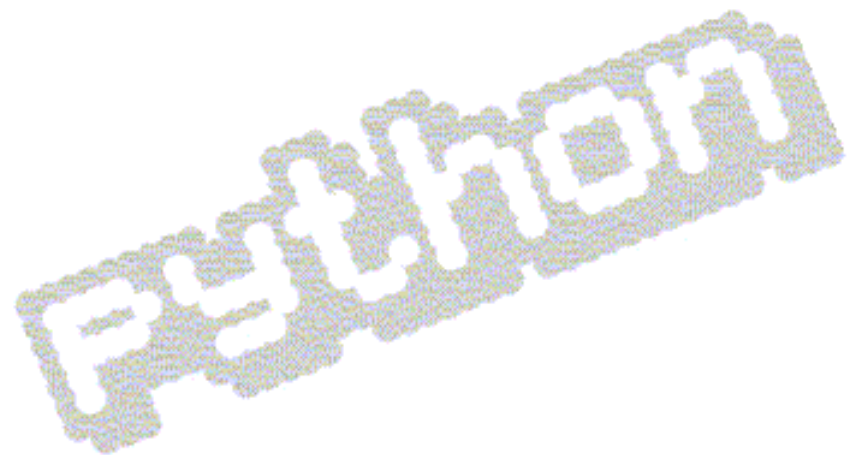
Entrada e Saída de dados

- Imprimindo com o comando print
 - `a = 3`
`print "O número é ",a`
`a="FIPP-Faculdade de Informática de P.P. "`
`print a`
 - `print "Pulando a linha \n"`
`print "Emitindo sinal de advertência","\a"`
 - `print 17 + 3`
`print 2 * 45`



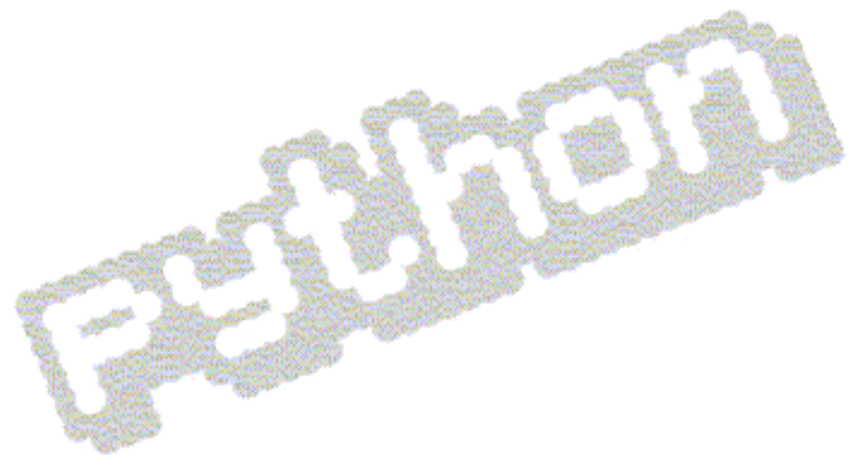
Entrada e Saída de dados

- Lendo dados com o comando `raw_input`
 - `a = raw_input("Lendo uma string: ")`
 - `a = int(raw_input("Lendo um inteiro: "))`
 - `a = float(raw_input("Lendo um float: "))`



Operadores Aritméticos

- Operadores aritméticos em Python seguem a mesma linha das Linguagens C e C++
 - + adição
 - - subtração
 - * multiplicação
 - / divisão (int)
 - % resto



Operadores Lógicos

- Equivalentes a C e C++
 - `x==y` "x é igual a y"
 - `x!=y` "x é diferente de y"
 - `x>(=)y` "x é maior que y" "x é maior ou igual a y"
 - `x<(=)y` "x é menor que y" "x é menor ou igual a y"



Condicional

- Condicional if
 - a=5
 - if a == 5:
 - print "A é igual a 5, portanto"
 - print "5 é igual a A"
 - a = 5 + 5
 - print a
 - print "Fim"

Python

Condicional

- Condicional if com else

- a=2

- b=3

- if a<b:

- a=a+1

- print a

- else:

- b=b+1

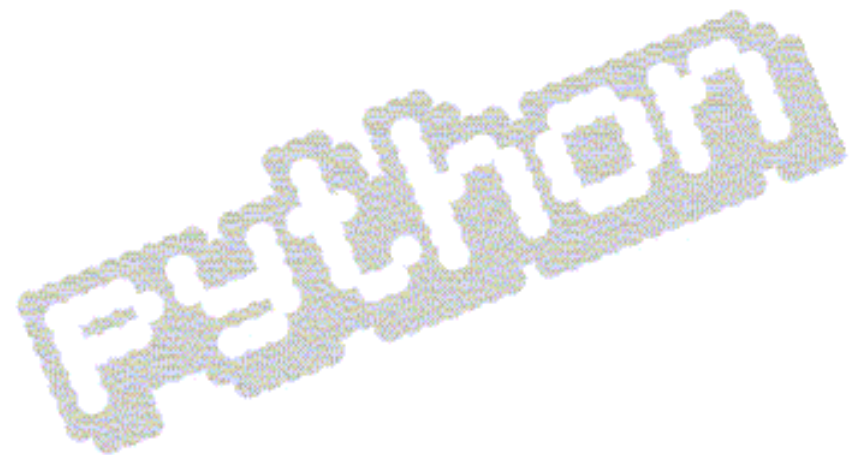
- print b

- print "Fim"

Python

Condicional

- Condicional if com o elif
 - `op=raw_input("Entre com a opção: ")`
 - `if op == 1:`
 - `print "Opção 1"`
 - `elif op == 2:`
 - `print "Opção 2"`
 - `else:`
 - `print "Opção 3"`



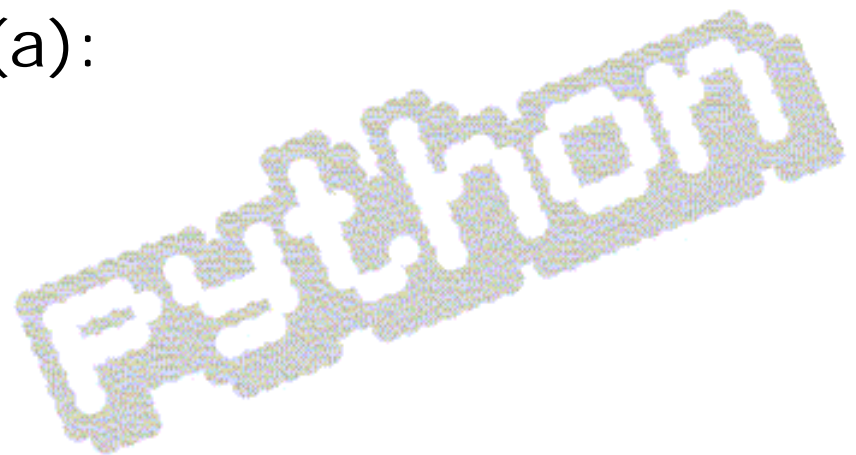
Funções

- Criando funções em Python
 - `def newLine():`
 `print`
 - `def twoLines():`
 `newLine()`
 `newLine()`
 - `print "Função pula 2 linhas"`
 - `twoLines()`

Python

Funções com parâmetros

- Criando funções com parâmetros em Python
 - `def soma_um(a):`
 `a=a+1`
 `print a`
 - `def soma_um_retorna(a):`
 `a=a+1`
 `return(a)`



Funções Recursivas

- Criando funções recursivas em Python
 - def fatorial(n):
 - if n == 1:
 - return n
 - else:
 - return fatorial(n-1) * n

Python

Iterações

- Construindo laços de repetição

- import math

- x=1.0

- while x<10.0:

- print x, '\t', math.log(x)

- x=x+1

'\t' representa uma tabulação 'Tab'

Python

Strings

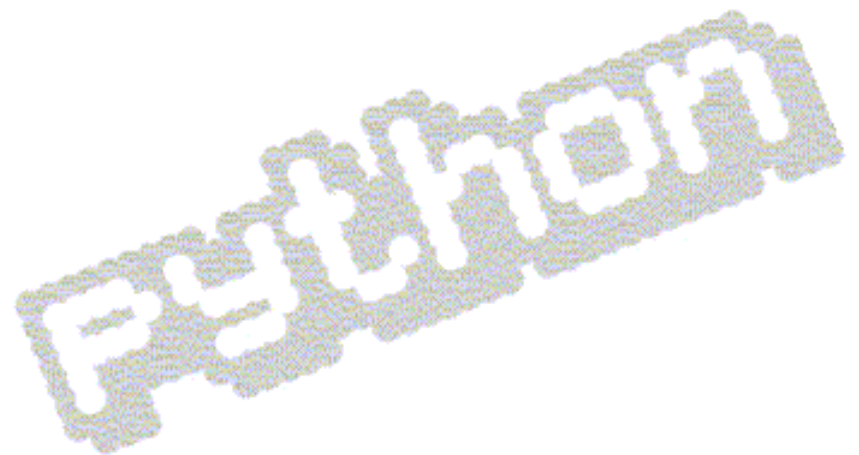
- Utilizando strings em Python
 - `pal = 'infoeste'`
 - `pal[0]`? 'i' -strings se iniciam pelo índice 0
 - `len(pal)`? 8 -len retorna o tamanho da string
 - `ultimo = pal[len(pal)]` #ERROR!
 - `pal[0:2]`? 'in' Imprime a partir da posição 0 e vai até a posição 3 que não é inclusa nesse subconjunto
 - `pal[:3]`? 'inf' Imprime os 3 primeiros caracteres. Ao contrário `pal[3:]`

Listas

- Como um vetor, porém com tamanho indefinido e capaz de armazenar dados de tipos diferentes.
 - `lista=[1,2,3,4,5,6]`
 - `lista=[1,'a',2,'b',3,'c']`
 - `lista[0]`? 1
 - `lista.append(4)` – insere o elemento 4
 - `lista.pop` – remove o último elemento
 - `lista.insert(i,x)` – insere um elemento x na pos i

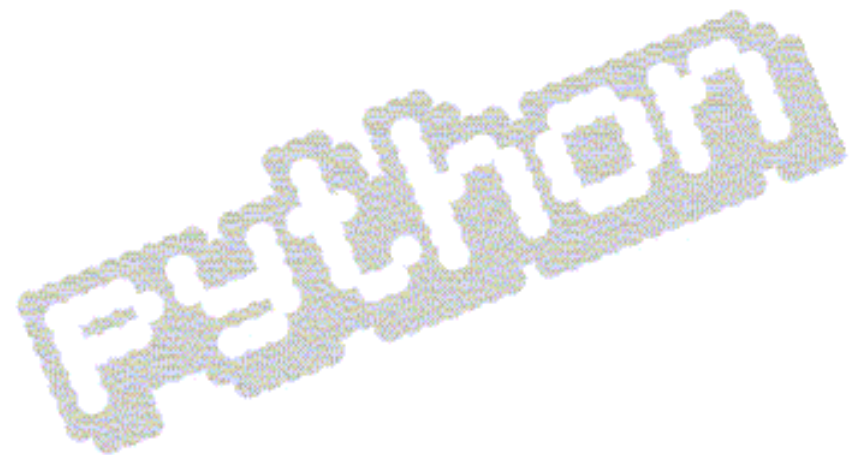
Iteração (For)

- Mágica
 - `numeros=[1,2,3,4,5,6]`
 `for i in numeros:`
 `print i`
 - `palavra='Infoeste'`
 `for car in palavra:`
 `print car`



Tuplas

- Semelhante às listas, porém não podem ser alteradas
 - `tupla=(1,2,3,4,5,6)`
 - `tupla[0]`? 1
 - `tupla[:3]`? 1,2,3



Exercícios

- 1) Desenvolva um programa que apresente as tabuadas do 1 ao 10, imprimindo os resultados no seguinte padrão: $6 \times 7 = 42$.
- 2) Dizemos que um número i inteiro positivo é perfeito, se este número for igual à soma dos seus divisores próprios. Exemplo: 6 é perfeito, pois $1 + 2 + 3 = 6$. Dado um número natural, verificar se ele é perfeito ou não.
- 3) Faça uma função que receba um número inteiro e retorne cálculo do fatorial desse número.

Faltam

Dicionários

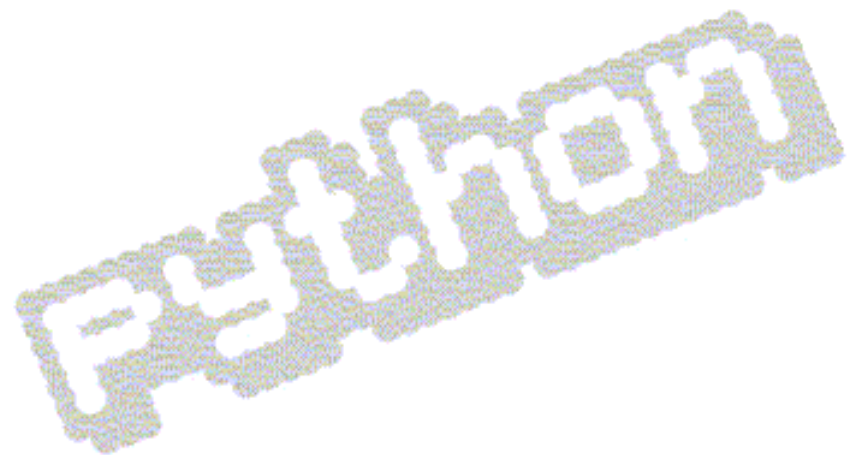
- Os dicionários associam um objeto (chamado de chave) a outro objeto (chamado de valor). As chaves podem ser de qualquer tipo, salvo de algumas exceções.
 - `dic={'livro':'book', 'gato':'cat'}`
 - `dic['mesa']='table'`
 - Existe a chave: `dic.has_key('mesa')`

Python

Exercício

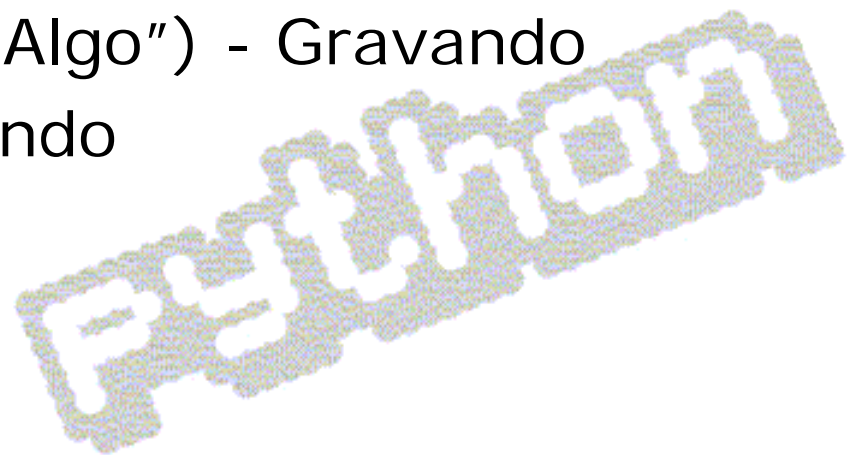
- Dado as seguintes matrizes de strings:
-ingles={"cat","wall","mouse","book","table"}
-portugues={"gato","muro","rato","livro","mesa"}
Escreva um programa em C, que ao ler uma palavra em português, faça a sua tradução para o inglês (caso encontre). O programa deverá ser executado repetidamente até que a palavra "fim" seja digitada. Sendo desenvolvido em C, era necessário o uso de matrizes, agora em Python, qual seria a melhor solução?

Programando 2ª Parte



Arquivo

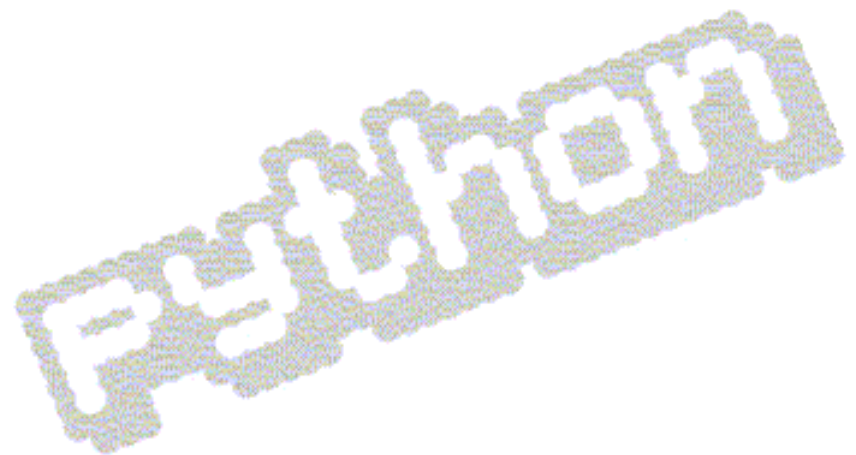
- Python trabalha de forma simples e eficiente com arquivos do tipo binário e texto
 - `arq = open("teste.dat", "w"/"r")` - Abrir
 - `arq.close` - Fechar
 - `arq.write("Escrevendo Algo")` - Gravando
 - `dado = arq.read()` - Lendo



Classes

- Você pode definir as suas próprias classes em Python com métodos e atributos privados
 - class quadrado:

```
def __init__(self):  
    self.x=0  
    self.y=0  
def setX(self,vx):  
    self.x=vx
```



Classes

- Desenvolva um programa para controlar uma conta bancaria, utilizando classes e as demais ferramentas da linguagem.
 - class conta:

...

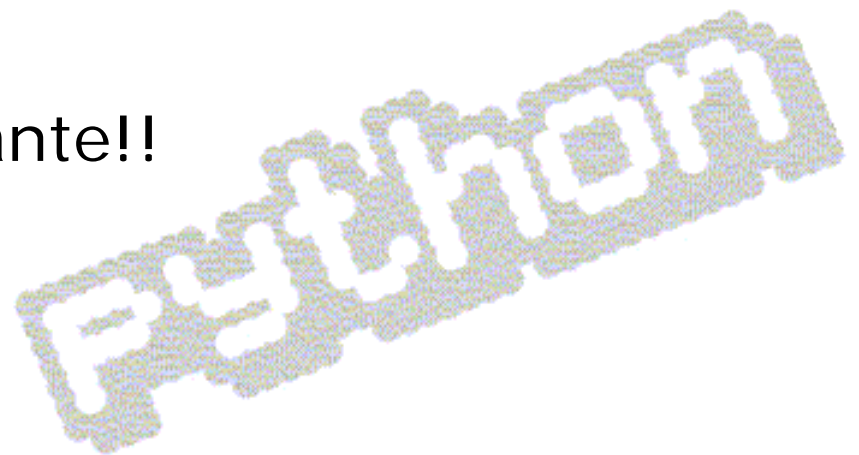
...

...

Python

Finalmente

- Não se esqueça
 - Python é Free
 - RAD
 - Orientada a Objeto
 - Multiplataforma
 - Alto Nível
 - Documentação Abundante!!
- Obrigado!!!!



Onde encontrar?

- Diogo...
diogobranquinho@gmail.com
- Python
www.python.org
www.pythonbrasil.com.br
- O resto
www.google.com.br

