

Universidade Federal do Pará
Instituto de Ciências Exatas e Naturais
Centro Acadêmico de Ciência da
Computação

Curso de Introdução a Linguagem de
Programação Python

Palestrante: Diego Damasceno
damasceno.diego@gmail.com
<http://damascenodiego.wordpress.com>



Objetivo do Mini Curso

- Proporcionar uma visão geral da linguagem
- Domínio das Estruturas Básicas da Linguagem
- Servir de base para um futuro aprendizado mais aprofundado

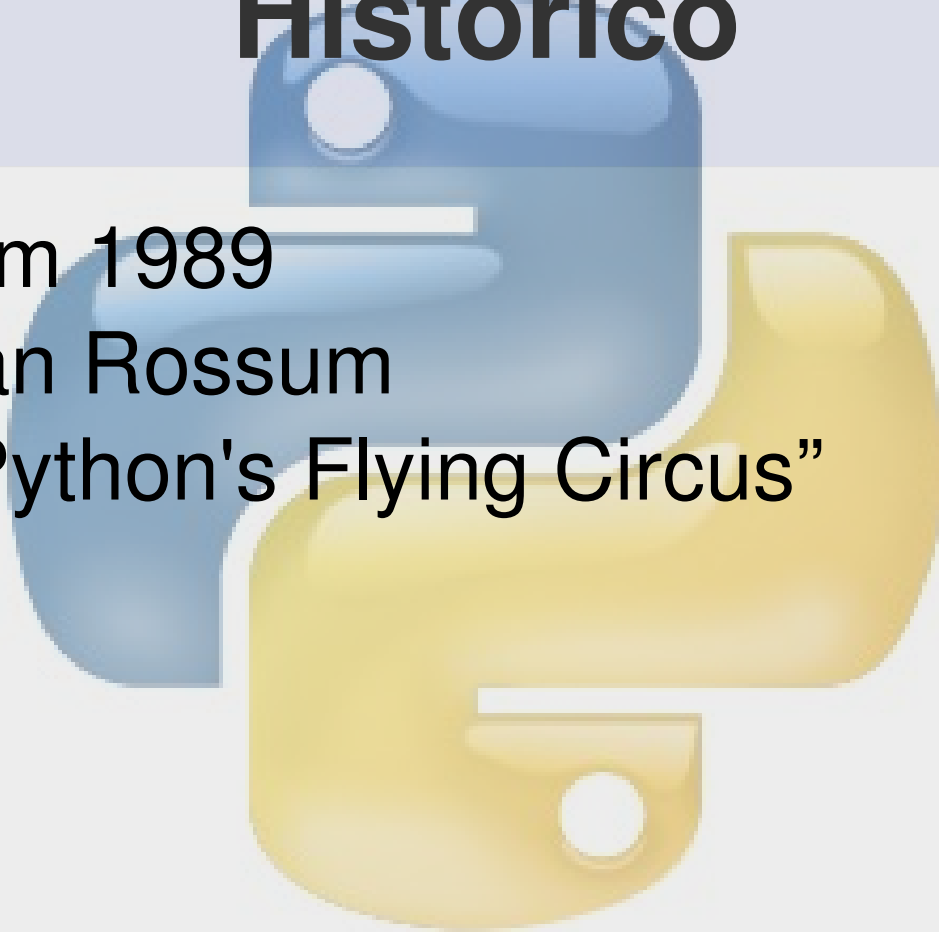
Tópicos Abordados

A large, semi-transparent watermark of the Python logo is centered in the background. The logo consists of two interlocking snakes, one blue and one yellow, with a white outline. The snakes are positioned behind the text, with the blue snake on the left and the yellow snake on the right.

- Histórico
- Características Gerais da Linguagem
- Sintaxe
 - Tipos de Dados
 - Estruturas de Dados
 - Estruturas de Controle
- Referências

Histórico

- Criada em 1989
- Guido van Rossum
- “Monty Python's Flying Circus”



Histórico

Há mais de seis anos, em dezembro de 1989, eu estava procurando por um projeto de programação como "hobby" que me mantivesse ocupado durante a semana próxima ao Natal. Meu escritório... estaria fechado, mas eu tinha um computador em casa, e não muito mais do que isso em mãos. Eu decidi escrever um interpretador para a nova linguagem de scripting sobre a qual eu vinha pensando ultimamente: uma descendente da ABC que agradaria a hackers de Unix/C. Eu escolhi Python como um título provisório para o projeto, sendo que eu estava num humor um pouco irreverente (e sendo também um grande fã do Monty Python's Flying Circus).

— Introdução de Programming Python, por Mark Lutz, O'Reilly

Características Gerais

The Python logo is a large, stylized 'P' shape composed of two interlocking snakes. The top half is blue and the bottom half is yellow. It is positioned in the background of the slide, behind the text.

- Multi-Paradigma (Imperativa, OO e Funcional)
- Propósito Geral
- Interpretada (Script)
- Multiplataforma
 - Windows
 - MacOS
 - Linux
 - Web
 - S60

Aplicações

- Scripting

```
#!/usr/local/bin/python

import string, sys

# If no arguments were given, print a helpful message
if len(sys.argv)==1:
    print 'Usage: celsius temp1 temp2 ...'
    sys.exit(0)

# Loop over the arguments
for i in sys.argv[1:]:
    try:
        fahrenheit=float(string.atof(i))
    except string.atof_error:
        print repr(i), "not a numeric value"
    else:
        celsius=(fahrenheit-32)*5.0/9.0
        print '%i\260F = %i\260C' % (int(fahrenheit), int(celsius+.5))
```

Aplicações

- Aplicações Web (Python Twitter)



Aplicações

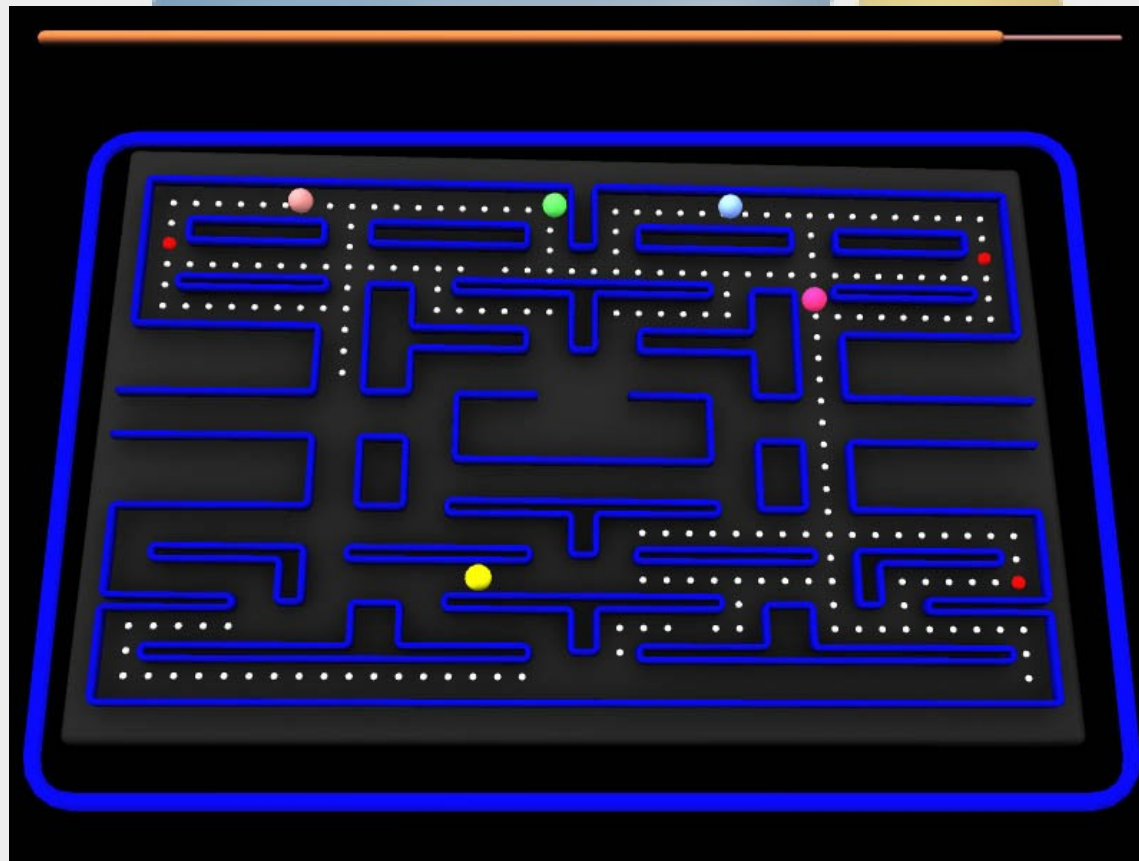
- Aplicações Web – Django

The Python logo, consisting of two interlocking snakes, one blue and one yellow, is positioned in the background of the slide.

django[®]

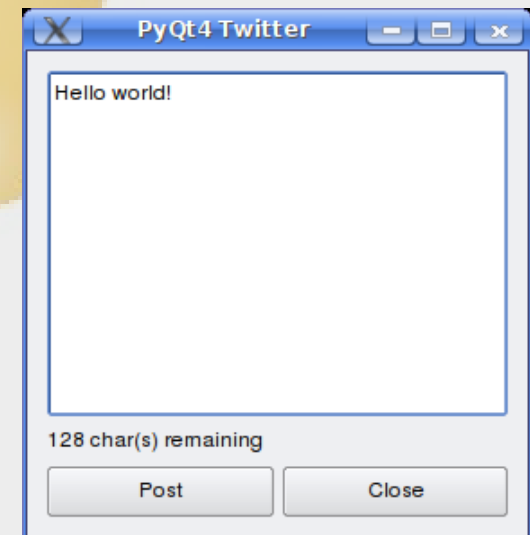
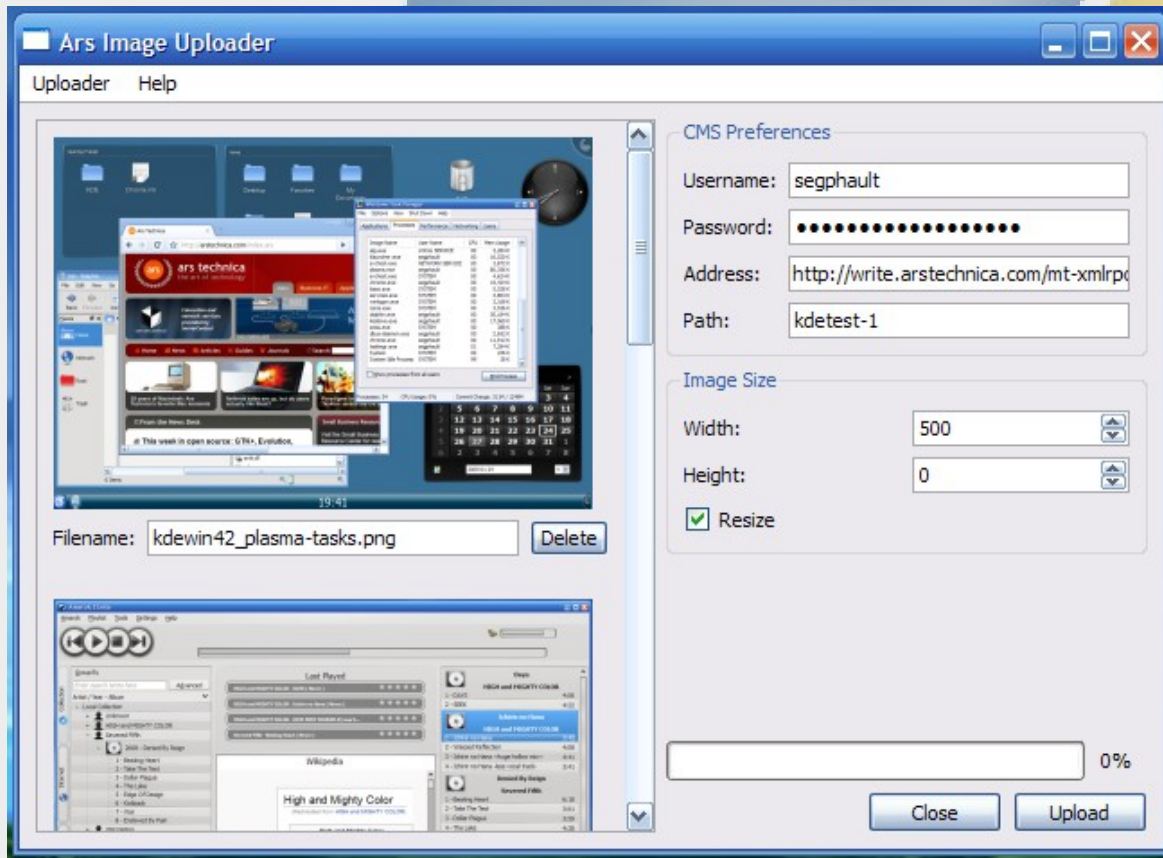
Aplicações

- Computação Gráfica



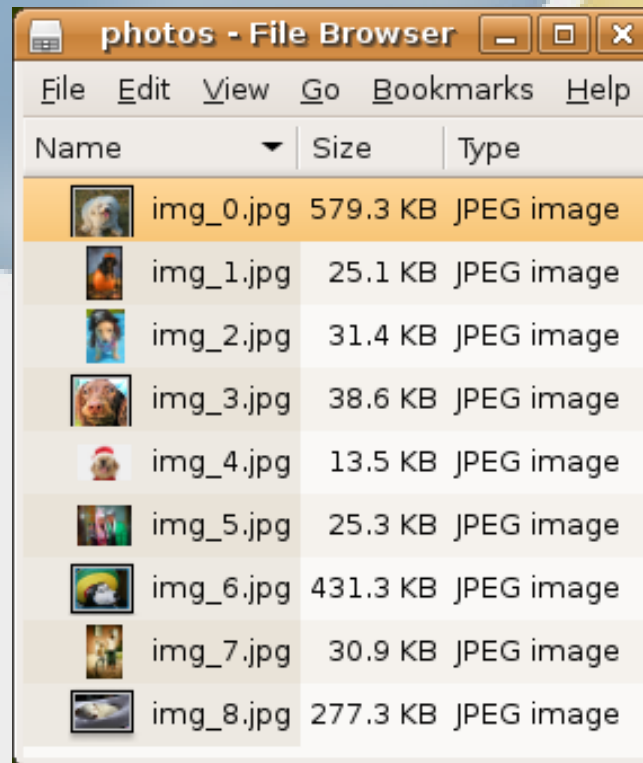
Aplicações

- GUI - PyQt



Aplicações

- GUI - PyGTK



Aplicações

- Jogos - PyGame



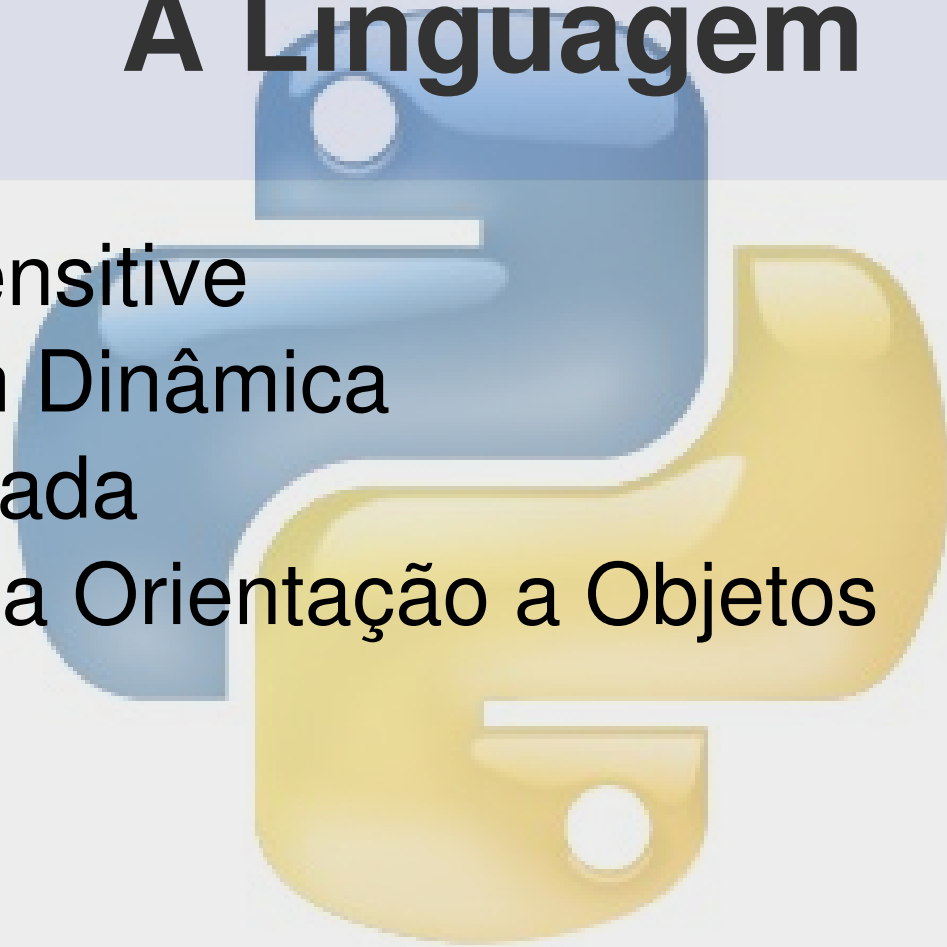
Aplicações

- Computação Científica (NumPy, SciPy...)



A Linguagem

- Case Sensitive
- Tipagem Dinâmica
- Interpretada
- Suporte a Orientação a Objetos



Hands On Python !



Abrir o Ambiente



```
$ python <Enter>
```

PyDev - Eclipse



<http://pydev.org/>

PyDev - Eclipse

PyDev - coilib50/source/python/coilib50/basic/class/_tests/test_class.py - Eclipse SDK

File Edit Source Refactoring Navigate Search Project Pydev Run Window Help

Pydev Package Expl

test_class.py

```
28 #check when cache is active
29 self.assertEqual(set([1, _E, _D, _C, _B, _A, object]),
30 self.assertEqual(set([_E, _D, _C, _B, _A, object]), Ge
31
32 self.assertEqual(set([_D, _C, _B, _A, object]), GetClas
33 self.assertEqual(set([_C, _B, object]), GetClassHierar
34
35 #check when cache is active
36 self.assertEqual(set([_A, object]), GetClassHierarchy(
37 self.assertEqual(set([_A, object]), GetClassHierarchy(
38
39
40 def testIsInstance(self):
41 '''
42 Check if IsInstance works with class name.
43 '''
44 self.assert_(IsInstance(_C(), '_B'))
45 ERROR_NOT_DEFINED_VARIABLE
46 self.assert_(IsInstance(_C(), ('_B',)))
47 self.assert_(not IsInstance(_C(), ('_A',)))
48 self.assert_(IsInstance(_C(), ('_A', '_B')))
49 self.assert_(not IsInstance(_C(), ('_A', '_D')))
50
51
52 def testIsSubclass(self):
```

Outline

- unittest
 - GetClassHierarchy, IsI
 - _A
 - _B
 - _C
 - _D
 - _E
 - Test
 - testClassHierarchy
 - testIsInstance
 - testIsSubclass
 - profileIsInstance
 - timeit

Problems Error Log Console

<terminated> C:\temp\coilib50\source\python\coilib50\basic\class_tests\test_class.py

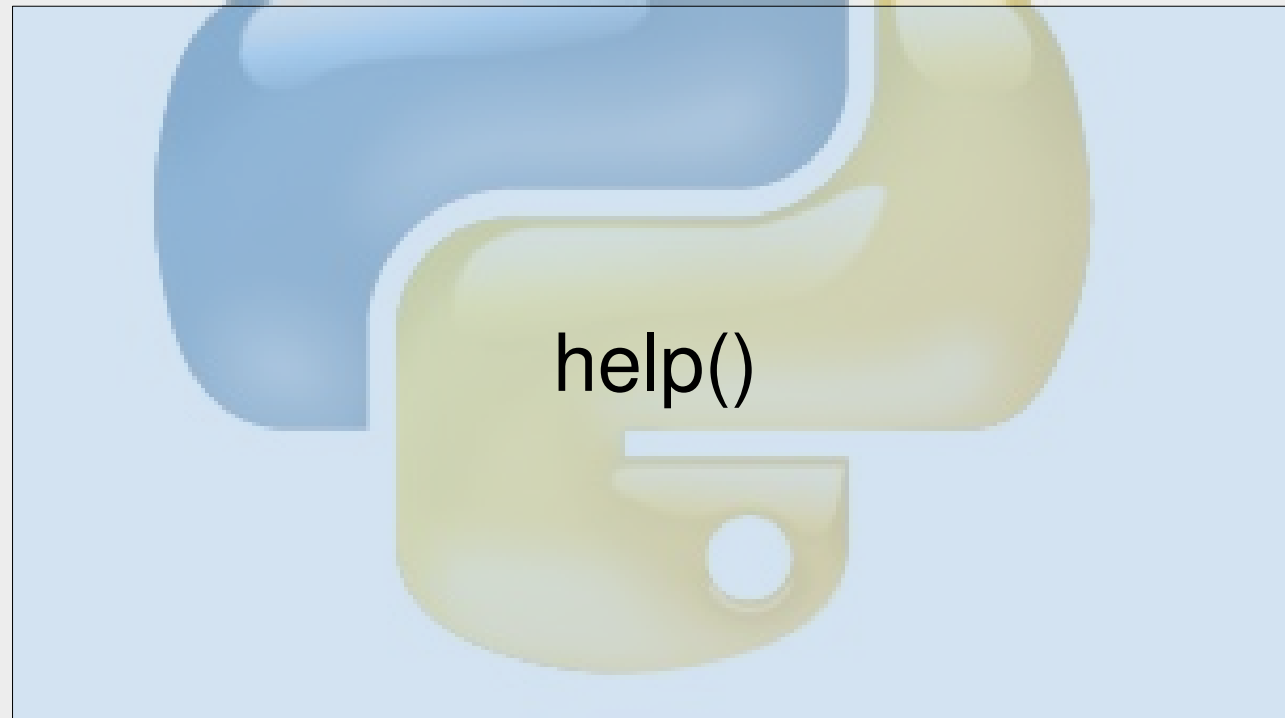
Traceback (most recent call last):

File "C:\temp\coilib50\source\python\coilib50\basic\class_tests\test class.py", line 2

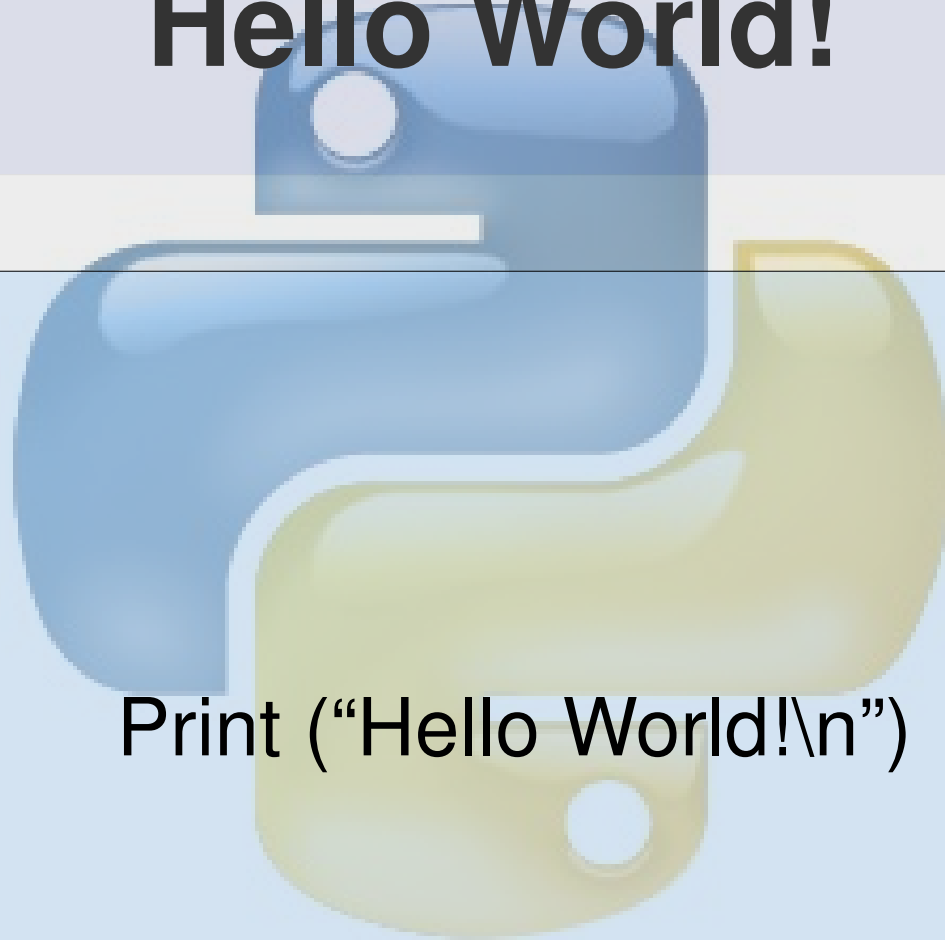
```
self.assertEqual(set([1, _E, _D, _C, _B, _A, object]), GetClassHierarchy(_E))
```

AssertionError: set([1, <class '__main___.E'>, <class '__main___.B'>, <class '__main___.D

Duvidas ?

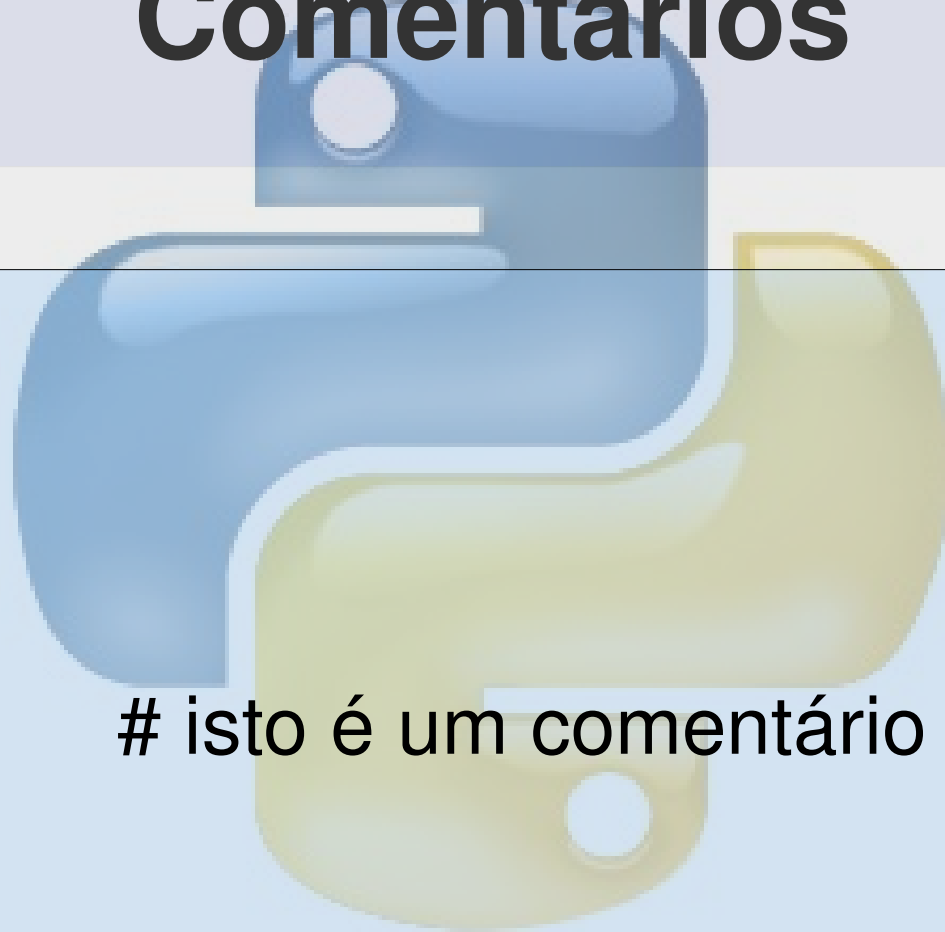


Hello World!



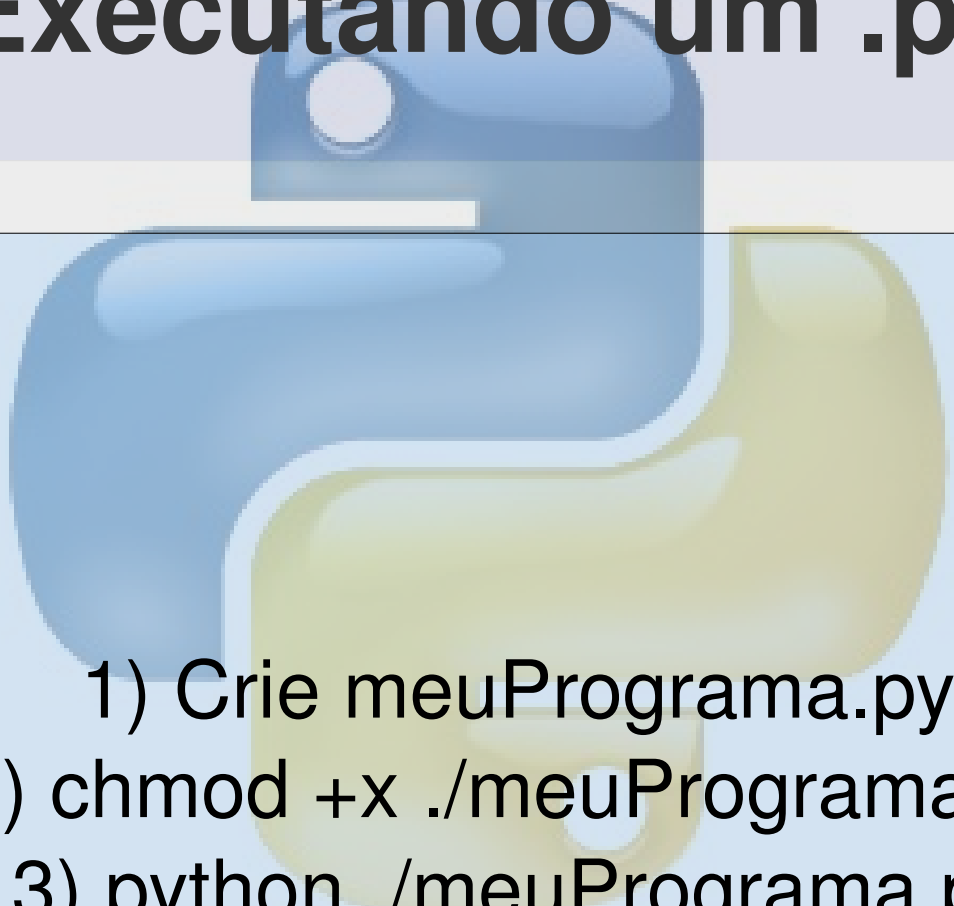
```
Print ("Hello World!\n")
```

Comentários



```
# isto é um comentário
```

Executando um .py

- 
- The Python logo, consisting of two interlocking snakes, one blue and one yellow, is centered in the background of the slide.
- 1) Crie meuPrograma.py
 - 2) `chmod +x ./meuPrograma.py`
 - 3) `python ./meuPrograma.py`

Exercícios

- Faça um programa que escreva “Ola cacc!”
- Faça um programa que escreva seu nome.

Entrada de Dados Via Teclado

```
X = raw_input("digite uma palavra")
```

```
Y = input("digite um valor numerico")
```

Tipagem Dinâmica

X= 50

type(X)

X="meu nome"

type(X)

X= 1.2345

type(X)

Operações Básicas

$3+4$ # Adição

$10-9$ # Subtração

$23*10$ # Multiplicação

$100/2$ # Divisão

$3\%2$ # Resto

Exercício

A large, semi-transparent watermark of the Python logo is centered in the background of the slide. The logo consists of two interlocking snakes, one blue and one yellow, forming a stylized 'P' shape.

- Faça um programa que pergunte seu nome, ano de nascimento e mostre seu nome e sua idade.
- Faça um Programa que peça o raio de um círculo, calcule e mostre sua área.
- Faça um Programa que pergunte quanto você ganha por hora e o número de horas trabalhadas no mês. Calcule e mostre o total do seu salário no referido mês.

Strings - Atribuição

S1=' '

S2="the book's on"

S3="the table"

Bloco="""o texto começa aqui
tem mais aqui
aqui
e termina aqui! """"

Strings - Concatenação

```
S1=' '
```

```
S2="the book's on"
```

```
S3="the table"
```

```
S2+S1+S3
```

```
mikeDoMosqueiro= "vai"
```

```
mikeDoMosqueiro*100
```

Strings - Index

```
nome= "mini curso do cacc"  
print nome[:10]  
print nome[-3]  
print nome[5:10]
```

Exercícios

- Faça um programa que com base numa variável **abc**, como segue abaixo, escreva seu nome usando as letras dela usando os índices das letras referentes ao seu nome.

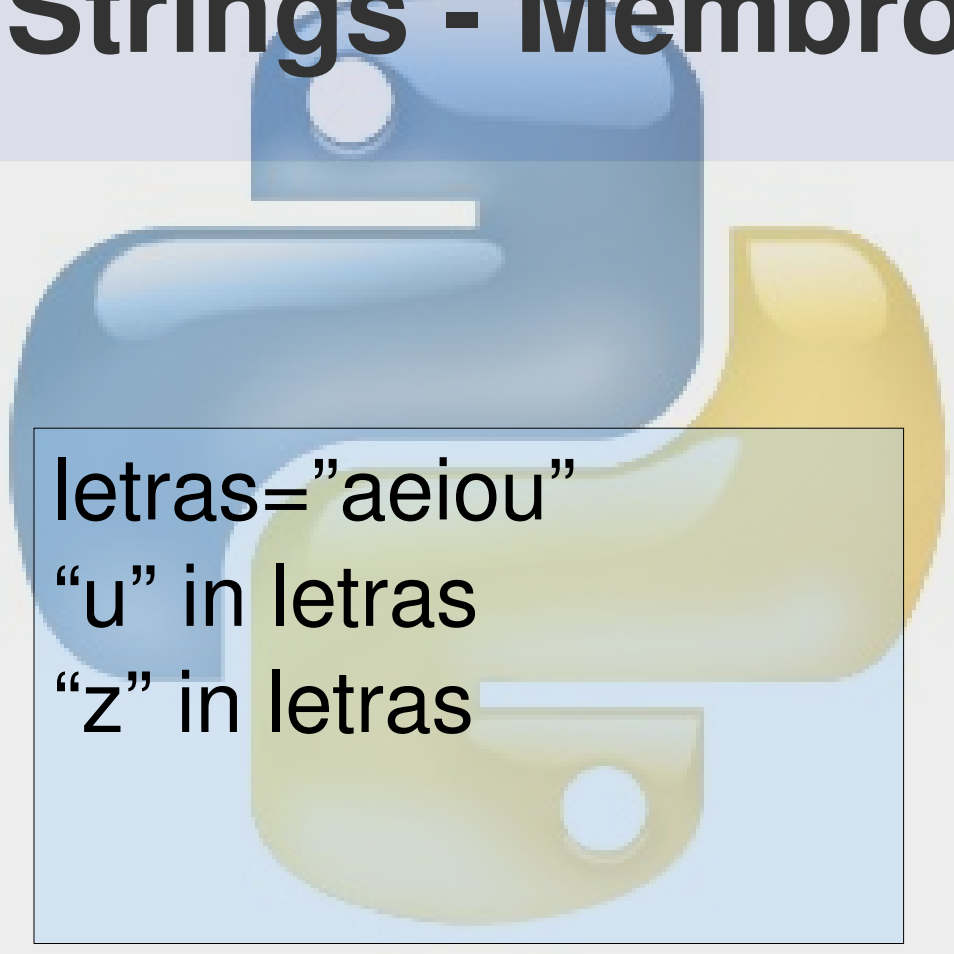
```
abc='abcdefghijklmnopqrstuvwxyz'
```

- Faça um programa que receba seu primeiro e ultimo nome e escreva somente a primeira letra dele.

Strings - Iteração

```
nome= "mini curso do cacc"  
for i in nome:  
    print i
```

Strings - Membro

The Python logo, consisting of two interlocking snakes, one blue and one yellow, is positioned in the background of the slide. A semi-transparent light blue box is overlaid on the logo, containing text.

```
letras="aeiou"
```

```
"u" in letras
```

```
"z" in letras
```

Strings – Formatação

```
x= "abcd"  
x.upper()  
x.lower()  
num="77"  
len(x)
```

Exercícios

- Faça um programa que receba seu nome e sobrenome, concatene os dois e escreva eles alternando cada letra em maiúscula e minúscula.
- Faça um programa que imprima seu nome ao contrário.

Listas



```
Lista=["yo"]  
Lista.append("hehe")  
Lista.pop()  
Lista.reverse()  
Lista.sort()  
range(100)  
xrange(10000)
```

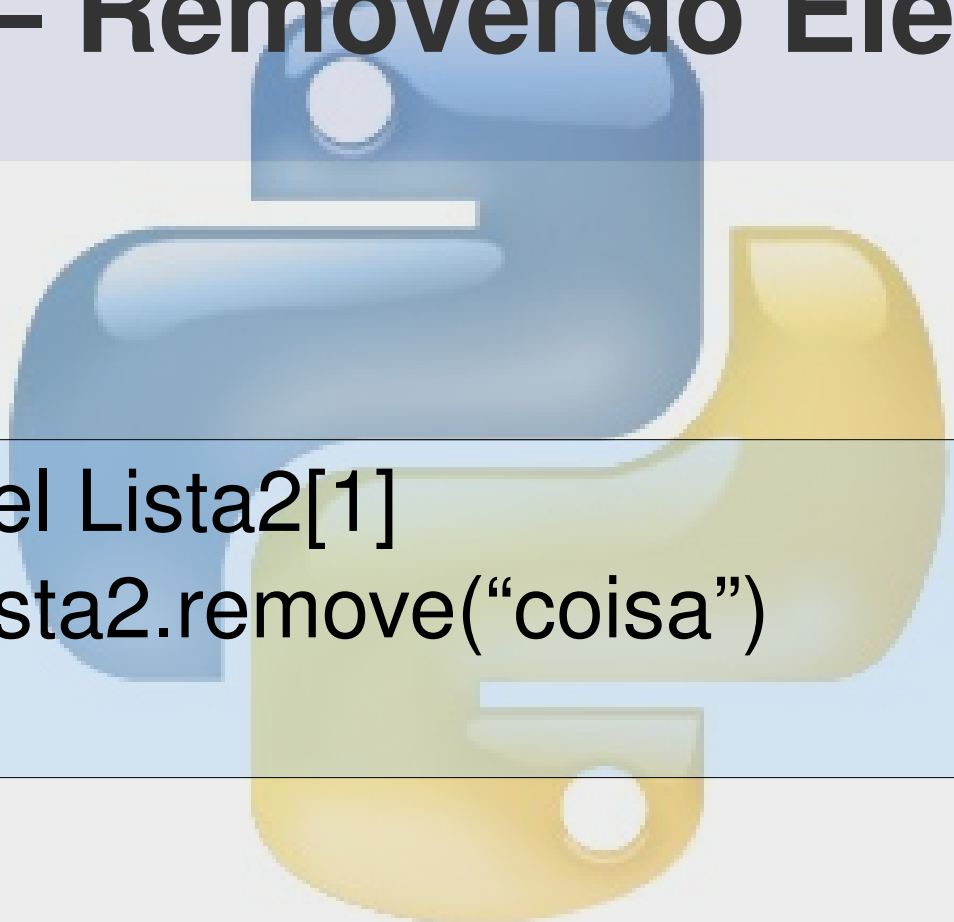
Listas – Adicionando Elementos

```
Lista1.append("item")  
Lista2.append("coisa")  
Lista2.extend(Lista1)  
Lista1[1]="treco"  
Lista1.insert(2, "outraCoisa")  
Lista1 += ["maisUmaCoisa"]
```

Exercícios

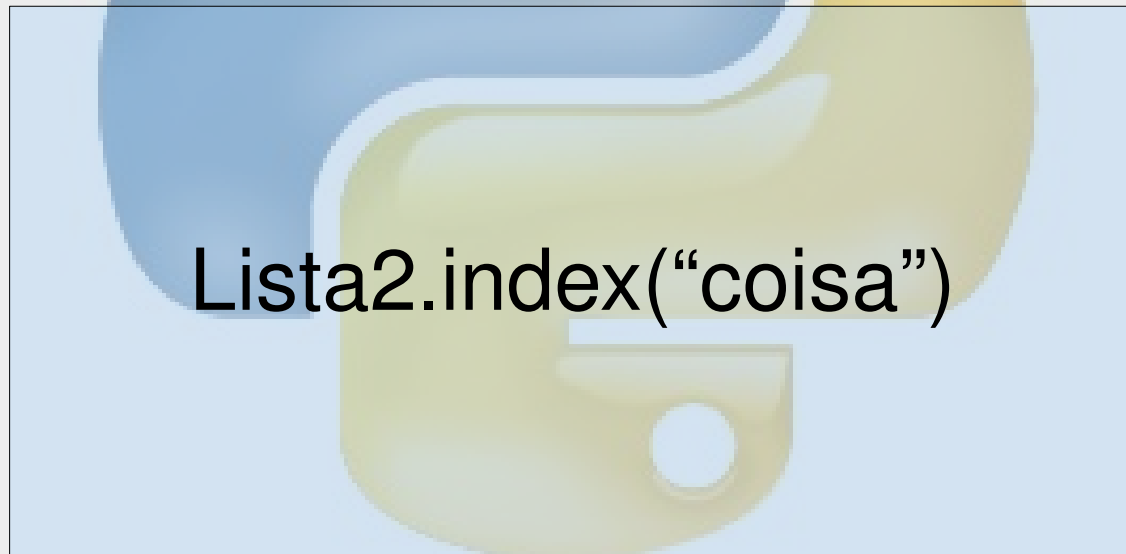
- Faça um programa que gere três sequências de números e após isso concatene as três e inverta a lista concatenada.

Listas – Removendo Elementos



```
Del Lista2[1]  
Lista2.remove("coisa")
```


Listas – Buscando Índices de Elementos



Exercícios

- Faça um programa que simule a inserção de 5 itens e a retirada de 2 desses 5 em uma mochila e imprima a lista de itens contidos nela e retirados.
- Faça um programa que adicione nomes a uma lista e obtenha a posição de dois desses nomes na lista.

Tuplas

```
tupla01=("primeirow","segundow","tres",4)  
tupla01[2]
```

Exercícios

- Faça um programa que receba uma matriz 3x2 e imprima na forma de um sistema usando uma tupla com os valores “x1”, “x2” e “=”.

Endentação

A large, semi-transparent Python logo watermark is centered in the background of the slide. It consists of two interlocking snakes, one blue and one yellow, forming a stylized 'P' shape.

#Espaços em branco fazem a endentação

```
if(x==10):  
    print "eh dez"  
else:  
    print "naum eh dez"
```

Estrutura de Decisão - if

```
if <test>:  
    <bloco do TRUE>  
else:  
    <bloco do FALSE>
```

Estrutura de Decisão - Switch

Não há switch!

```
if <test>:  
    <bloco do TRUE>  
elif <test>:  
    <bloco do FALSE>  
elif <test>:  
    <bloco do FALSE>  
(...)  
else:  
    <bloco do FALSE>
```

Estrutura de Repetição - FOR

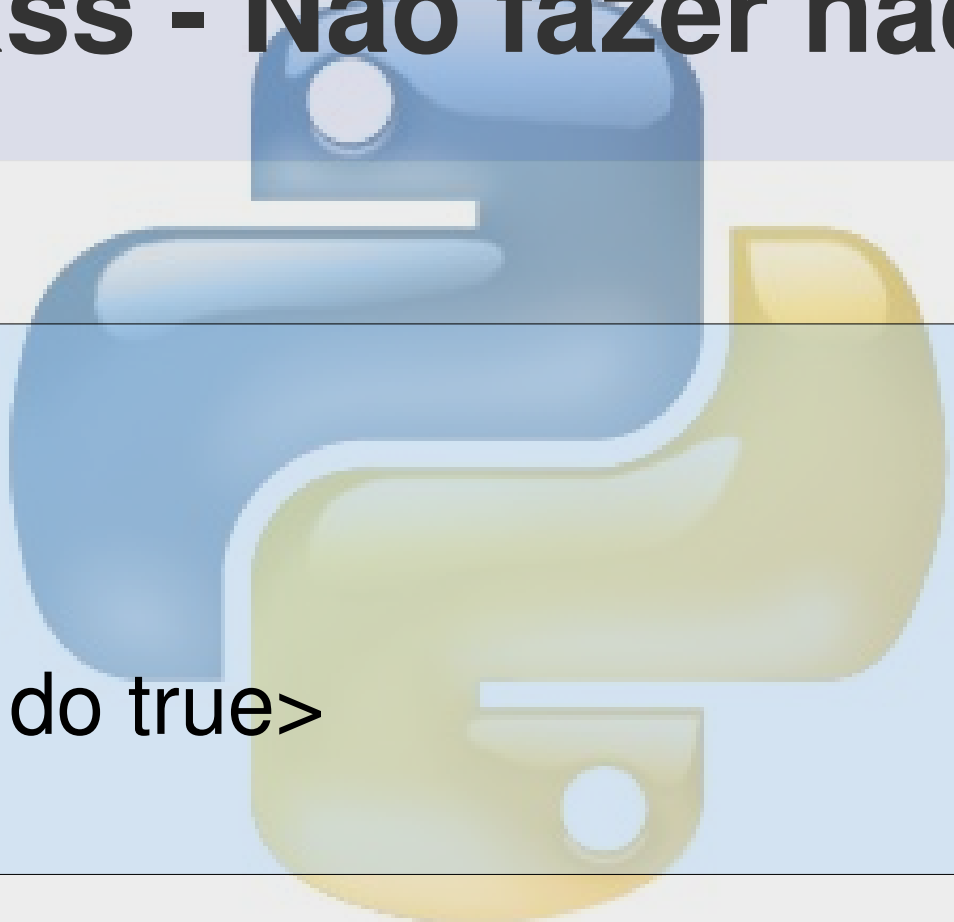
```
for <indice> in <objeto>:  
    <bloco do for>  
    ...  
    ...  
    ...  
    if <test>: break  
    if <test>: continue  
else:  
    <bloco caso não ocorra break>
```


Estrutura de Repetição - WHILE

```
while <test>
    <bloco do for>
    ...
    ...
    ...
    if <test>: break
    if <test>: continue
else:
    <bloco caso não ocorra break>
```

Pass - Não fazer nada!

```
if <test>:  
    pass  
else:  
    <bloco do true>
```

A large, semi-transparent Python logo watermark is centered in the background of the slide. It consists of two interlocking snakes, one blue and one yellow, forming a stylized 'P' shape.

Dicionários

```
dict01 = {}  
dict02 = {"primeiro": "andre",  
"segundo": "bené", "terceiro": "eloi"}  
dict02.keys()
```

Desafio !

- Leet speak generator. Leet é uma forma de se escrever o alfabeto latino usando outros símbolos em lugar das letras, como números por exemplo. A própria palavra leet admite muitas variações, como l33t ou 1337. O uso do leet reflete uma subcultura relacionada ao mundo dos jogos de computador e internet, sendo muito usada para confundir os iniciantes e afirmar-se como parte de um grupo. Pesquise sobre as principais formas de traduzir as letras. Depois, faça um programa que peça uma texto e transforme-o para a grafia leet speak.

Definindo Funções

A large, semi-transparent watermark of the Python logo is centered on the slide. The logo consists of two interlocking snakes, one blue and one yellow, forming a stylized 'P' shape.

```
def nomeDaFuncao(<listaDeParametros>):  
    <bloco da funcao>
```

Exercícios

- Faça um programa que leia um número indeterminado de valores, correspondentes a notas, encerrando a entrada de dados quando for informado um valor igual a -1 (que não deve ser armazenado). Após esta entrada de dados, faça:
 1. Mostre a quantidade de valores que foram lidos;
 2. Exiba todos os valores na ordem em que foram informados, um ao lado do outro;
 3. Exiba todos os valores na ordem inversa à que foram informados, um abaixo do outro;
 4. Calcule e mostre a soma dos valores;
 5. Calcule e mostre a média dos valores;
 6. Calcule e mostre a quantidade de valores acima da média calculada;
 7. Calcule e mostre a quantidade de valores abaixo de sete;
 8. Encerre o programa com uma mensagem;

Desafio !

- Faça um programa que simule uma mochila que suporte até 12 itens, e que você pode adicionar e retirar itens nela até enchê-la ou enviar a palavra “fechar”.
- Faça um programa que receba dois nomes e veja quais letras eles tem e comum, caso não tenham nenhuma, não escreva nada.

Orientação de Objetos – Classes e Métodos

```
class Pessoa:  
    def setNome(self,v):  
        self.nome=v  
    def setIdade(self,v):  
        self.idade=v  
    def getNome(self):  
        return self.nome  
    def getIdade(self):  
        return self.idade  
    def __init__(self,n,i):  
        self.nome=n  
        self.idade=i
```


Orientação de Objetos – Criando Instância

```
x=Pessoa("fulano",200)  
x.getNome()  
x.getIdade()
```

Exercícios

- Crie uma classe Casa que armazene o nome da rua, número da casa, nome do proprietário e o bairro da casa e imprima tudo isso com um método info()

Arquivamento - Ler

```
file= open("./meusDados.txt", "r")  
file.read()  
file.readline()  
file.seek(0)  
file.close()
```

Arquivamento - Gravar

```
file= open("./meusDados.txt", "w")  
file.write("texto")  
file.write(variavelQualquer)  
file.close()
```

Arquivamento - Concatenar

```
file= open("./meusDados.txt", "a")  
file.write("texto")  
file.write(variavelQualquer)  
file.close()
```

Exercícios

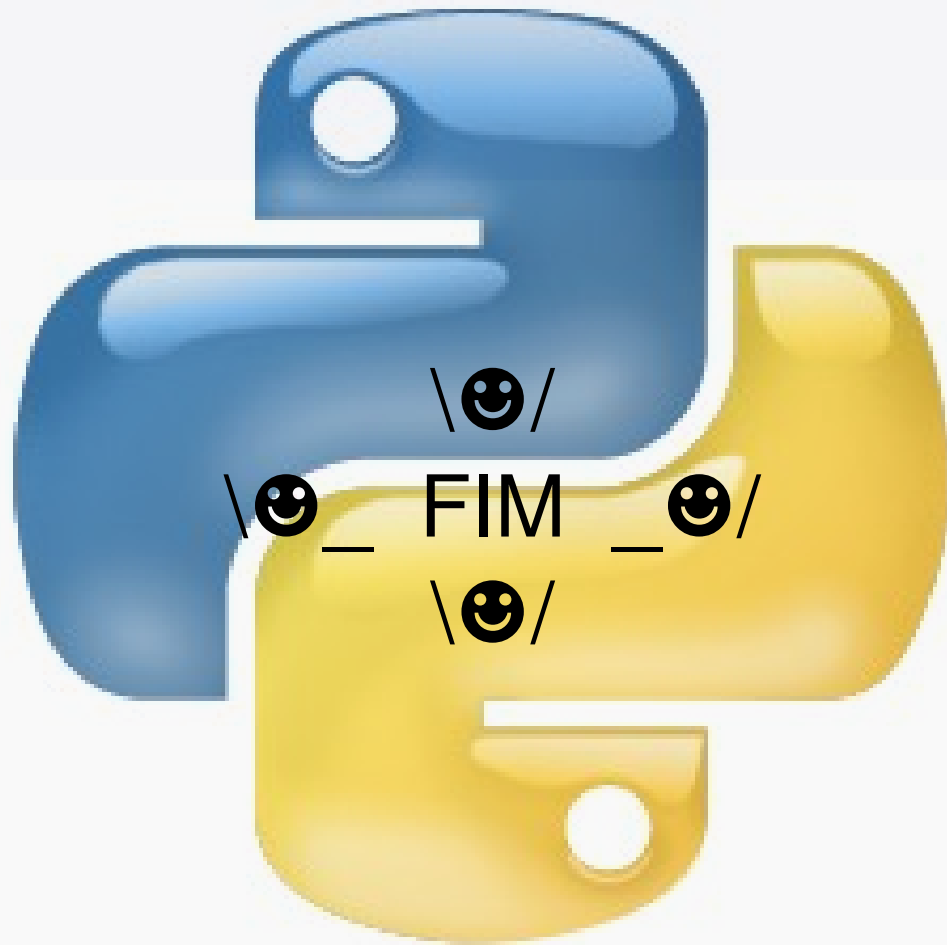
- Digite um programa que receba pelo teclado <seu nome> <telefone> <email> e armazene em um arquivo no seguinte formato:

Nome: <seu nome>

Email: <email>

Telefone: <telefone>

- Feche o arquivo e depois tente recuperar os dados salvos



Referências

A large, semi-transparent watermark of the Python logo is centered on the slide. The logo consists of two interlocking snakes, one blue and one yellow, forming a stylized 'P' shape.

- <http://www.python.org>
- <http://identi.ca/group/python>
- <http://python-ebook.blogspot.com/>
- <http://diveintopython.org/>
- <http://cgkit.sourceforge.net/>
- `irc: #python@freenode`