Mini-curso Python

BRUTE UDESC

Vinicius Gasparini
Peter Brendel

Python

- Nome inspirado do grupo inglês **Monty Python**
- Criada em 1991 por Guido von Rossum
- Versão atual **3.8.0** (lançada a 6 dias atrás)
- Objetivos da linguagem
  - Clareza e simplicidade de código
  - Portabilidade
  - Multi-propósito
  - Multi-paradigma

Exemplos de uso

dontpady

https://github.com/vgasparini/dontpady

Exemplos de uso

Asteroids Challenge

https://github.com/bruteudesc/top-asteroids-challenge

Exemplos de uso

bigodera bot

https://github.com/vgasparini/bigodera-bot

Exemplos de uso

scraper

https://github.com/vgasparini/ic_scraper

Exemplos de uso

CGR

TEG

LFA

ANN

Multi-propósito

- Inicialmente pensada para ser usada como linguagem de **script de shell**
- Atualmente é usada para:
  - **Desktop (**Tk, wxPython, Jython, IronPython)
  - **Aplicações web** (Django, Flask, Grok)
  - **Embarcados** (MicroPython)
  - **Maratona** (big integer, string)
- Usado no Youtube, Google, Instagram, Spotify, Reddit

Multi-paradigma

- **Imperativa**
  - Funções, estruturas de controle, módulos
- **Orientada a objetos**
  - Classes, objetos
- **Funcional**
  - Manipulação de listas

Vamos pro código

ctrl + alt + t -> Terminal
python3 -> Interpretador Python

Expressões

- Aritméticas
  - +, -, *, /, %
  - // (divisão inteira)
  - ** (exponenciação)
- Comparação
  - >, <, >=, <=, ==, !=
- Lógicas
  - and, or, not

Tipos

- Inteiros: int( )
  - 1, 3, 10111111111111111111111111111111...
- Reais: float( )
  - 1.3, 100.1111, 1.3e-10
- Complexo: complex( )
  - 1+9j
- String: str( )
  - ‘affonso’, “pet udesc”
- Booleano: bool( )
  - True, False (True pode ser também qualquer valor diferente que 0)

Funções básicas - input

- **input(“insira um valor”)**
  - Retorna uma **string**
- **Casting**
  - Como retorna uma **string**, devemos fazer um **typecast.**
  - int(input(“insira um valor”)

Funções básicas - print

- **print(“aqui esta o valor”, var)**
  - Imprime na tela
- **print(“aqui esta o valor {} no meio do texto”.format(var))**
- **print(“aqui esta o valor %d no meio do texto” % (var))**
- **print(“aqui esta o valor %.4f no meio do texto com 4 casas” % (var))**

Funções básicas - atribuição

- a = 10
- a,b = 10,15
- a,b = [12, 22]
- a += 1
- a -= 1
- a *= 2
- a /= 3
- a //= 3
- a%=6

Funções básicas - string

- a = ‘oi’ + ‘ ‘ + “tudo bom” + ‘?’
- a = ‘A’ * 50
- ‘arroz’ in ‘arroz com feijão’
- ‘arroz’[0] == ‘a’
- ‘arroz’[1:3] == ‘rr’
    - índices em Python são [a, b)
- ‘arroz’[-1] == ‘z’
- ‘arroz’[:2] == ‘ar’

Funções básicas - controle

- if, elif, else
  - a = 3
  - if a<3:
    - arroz
  - elif a>3:
    - feijao
  - else:
    - batata

Funções básicas - controle

- while, for
  - a = 0
  - while a<10:
    - a += 1
  - for j in range(10):
    - a -= 1

Funções básicas - controle

- range
  - range(5)
    - [0, 1, 2 ,3 ,4]
  - range(2,5)
    - [2, 3, 4]
  - range(2,5,2)
    - [2, 4]

Função (tão básicas quanto sua imaginação)

```
def soma(a, b):

    return a+b
```

Funções e métodos

Métodos são funções inerentes a um objeto

TUDO EM PYTHON É OBJETO!!!!111!!!ONZE

- a = ‘string bolada’
- dir(a)
    - Todos os métodos disponíveis do objeto str

Métodos úteis de string

a.capitalize()
a.lower()
a.upper()
a.count('oi')
a.replace('oi', 'tchau')
a.find('oi')
a.isalpha()
a.isnum()
a.isdecimal()

a.split(' ')
a.swapcase()
a.islower()
a.isupper()
a.zfill(10)
a.title()
a.join(['a', 'b', 'c']

Listas

É um objeto que guarda objetos e os indexa em 0 de maneira aninhada

Possuem tamanho variado e podem possuir diversos tipos mesclados

Listas - métodos úteis

- Declaração
  - a = [ ]
  - a = [0]
  - a = list(obj)

a[index]
a.sort()
a.reverse()
a.clear()
a.copy()
a.count(obj)
a.index(obj)
a.insert(pos, obj)
a.append(obj)
a.pop(pos)
a.pop()

Dicionários

É um objeto que associa pares de objetos - chave e valor

Possuem tamanho variado e podem possuir diversos tipos mesclados, porém os objetos de chave precisam ser **hashable**

Trick = use o tipo **tuple** ao atribuir uma lista como chave

Dicionários - métodos úteis

- Declaração
  - a ={}
  - a = { ‘oi’ : ‘tchau’ }
  - a[‘oi’] = 2
  - a = dict()
- **Trick**
  - b = [1, 2]
  - a = dict()
  - a[tuple(b)] = ‘tchau’

a[key]
a.get(key)
a.keys()
a.pop()
a.popitem(key)
a.values()
a[key] = novo_valor

BRUTE
competitive programming

Arquivos

- **open(**‘nome_arquivo’, ‘w+’)
  - Modos: vários, mas uso o w+ que ele lê e escreve
- **readline()** -> Retorna uma string de uma linha
- **read()** -> Retorna uma string com o arquivo inteiro
- **write(**string**)**
- **close()**

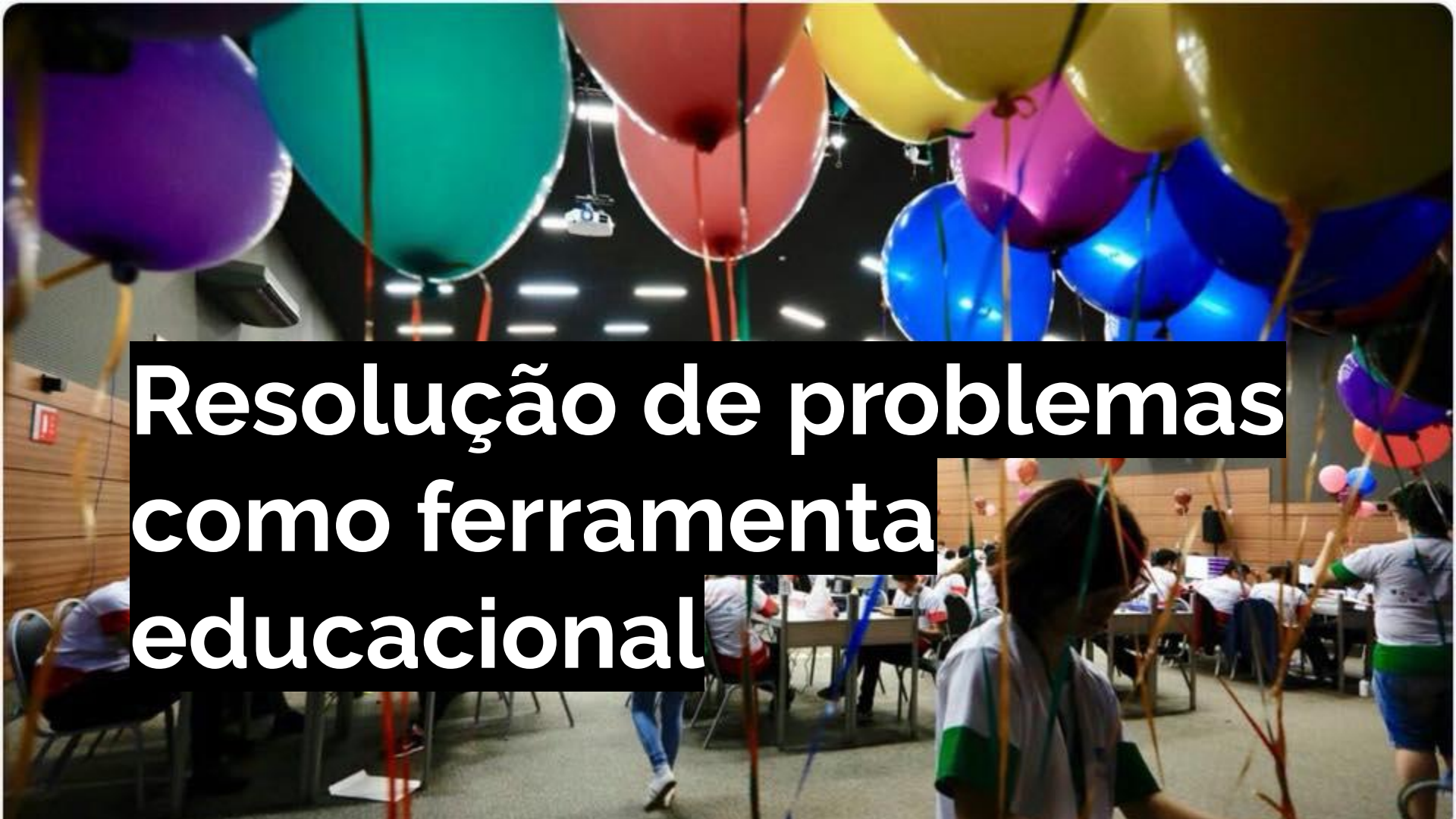
Arquivos

- **file = open(**‘teste’, ‘w+’)
- for line in file:
  - print(line)
- file.close()

- **file = open**(‘teste1’, ‘w+’)
- file.write(‘dae’)
- file.close()

Orientação a Objetos

Talk is cheap, show me the code



Resolução de problemas como ferramenta educacional

Link do Contest

Obrigado

Sextas-feiras
F301
09:00 - 21:00