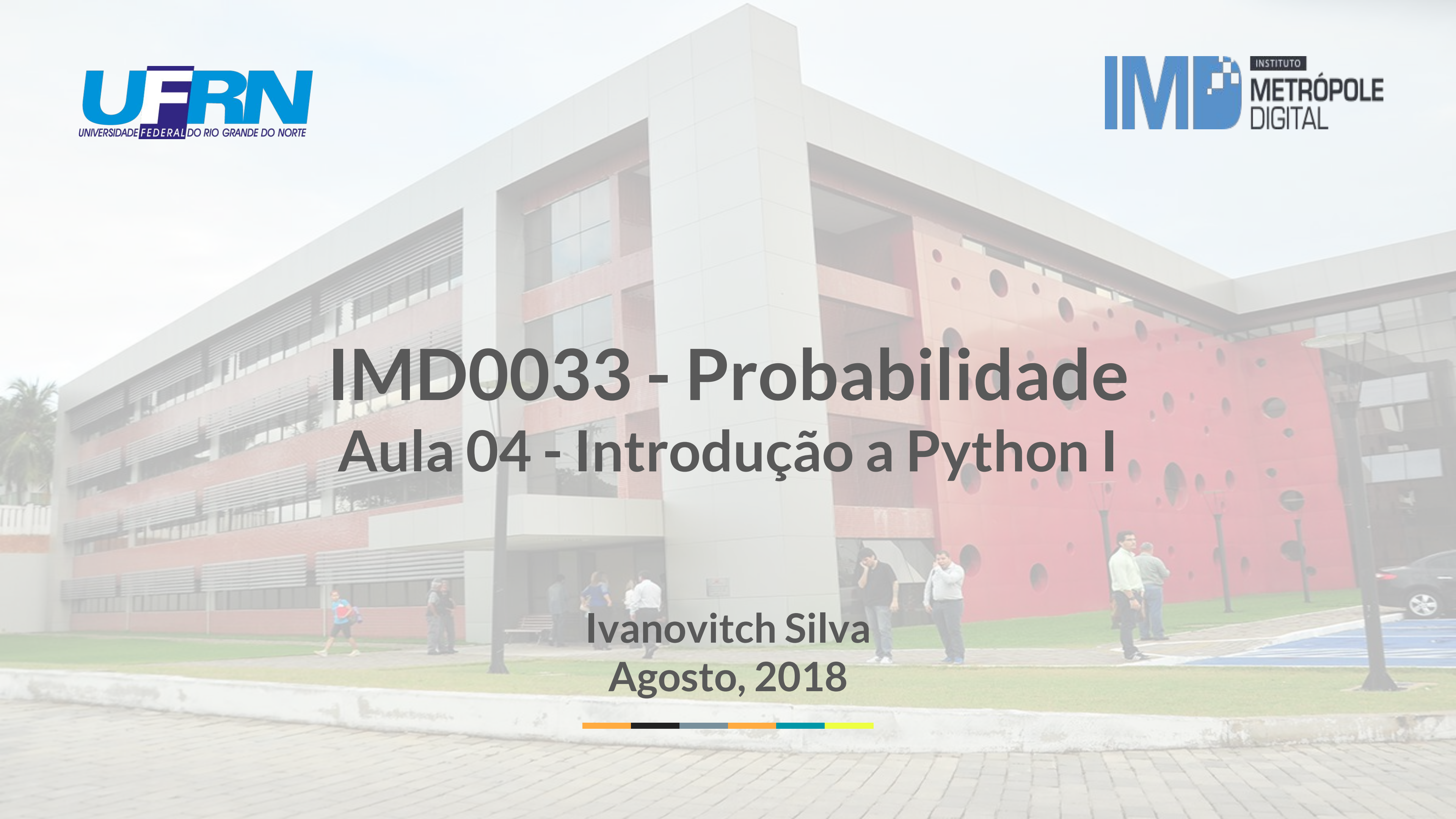


IMD0033 - Probabilidade

Aula 04 - Introdução a Python I

Ivanovitch Silva

Agosto, 2018

Agenda

- Dicionários
- Funções
- Erros
- Desafios

movie_metadata.csv
births.csv
la_weather.csv

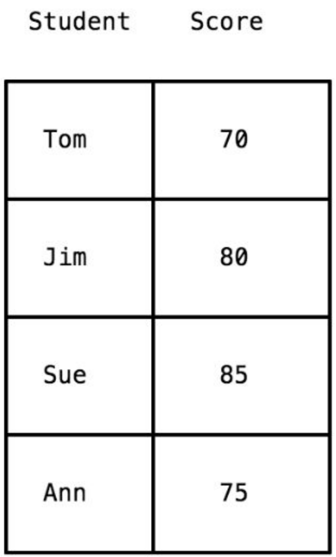
Atualizar o repositório

```
git clone https://github.com/ivanovitchm/imd0033_2018_2.git
```

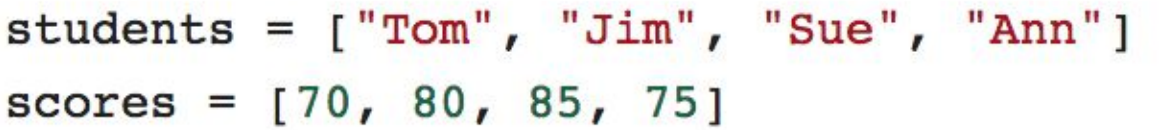
Ou ….

```
git pull
```

Dicionários - Motivação



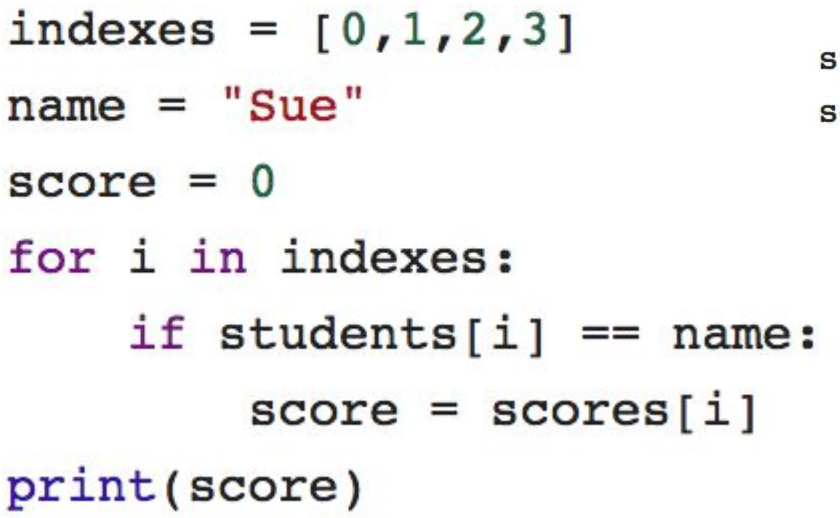
Student	Score
Tom	70
Jim	80
Sue	85
Ann	75



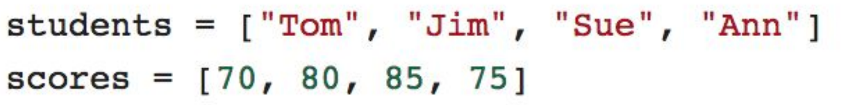
```
students = ["Tom", "Jim", "Sue", "Ann"]
scores = [70, 80, 85, 75]
```

Qual a nota de Sue?

Dicionários - Motivação



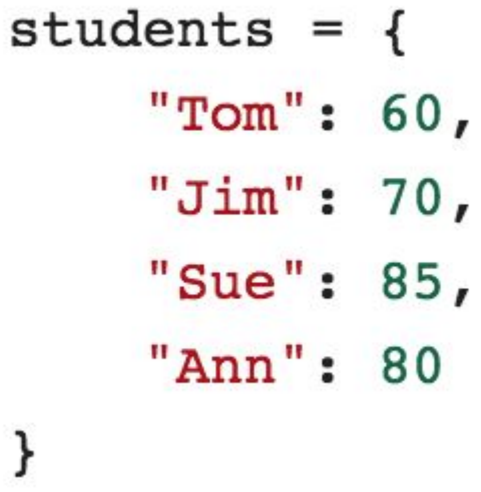
```
indexes = [0,1,2,3]
name = "Sue"
score = 0
for i in indexes:
    if students[i] == name:
        score = scores[i]
print(score)
```



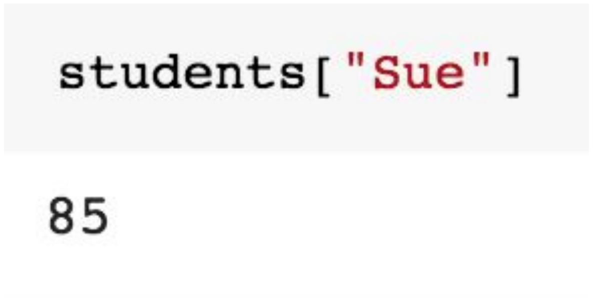
```
students = ["Tom", "Jim", "Sue", "Ann"]
scores = [70, 80, 85, 75]
```

Qual a nota de Sue?

Dicionários



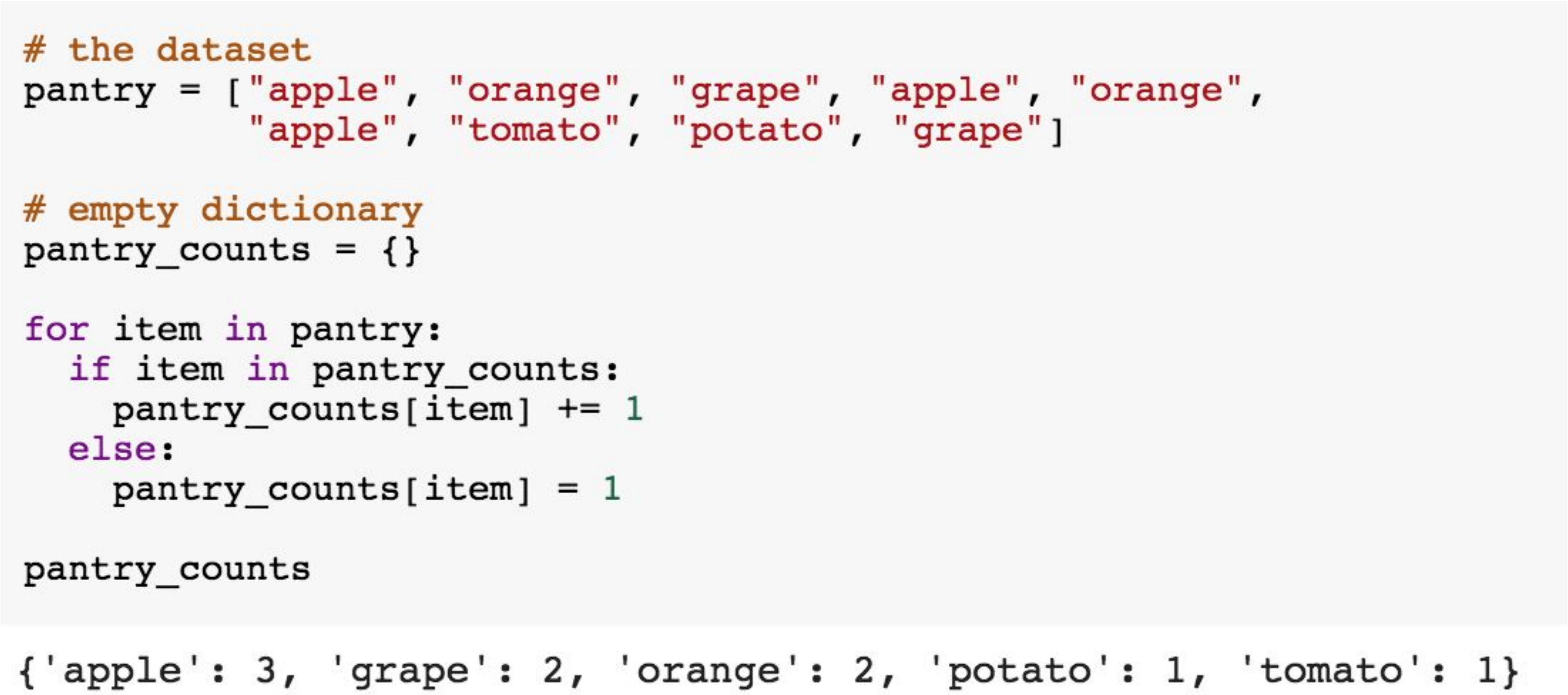
```
students = {
    "Tom": 60,
    "Jim": 70,
    "Sue": 85,
    "Ann": 80
}
```



```
students["Sue"]
```

```
85
```

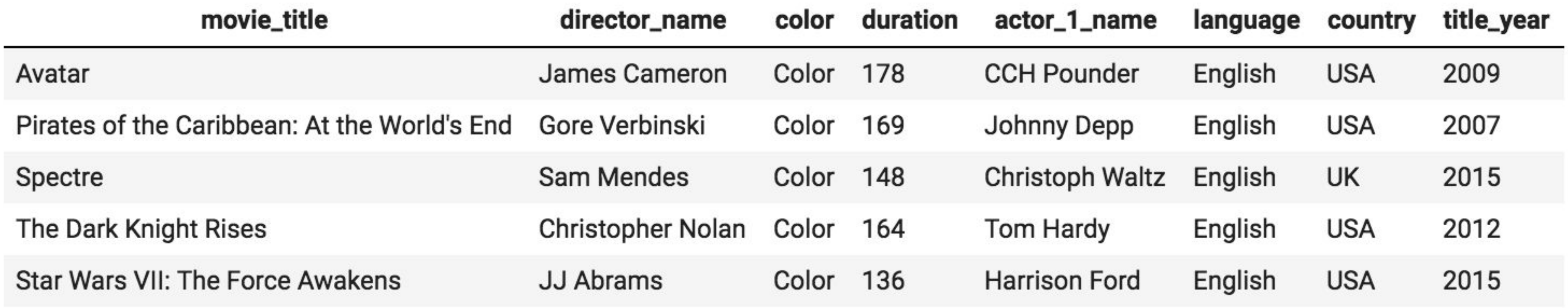
Utilizando dicionários como uma estrutura de contagem



```python
# the dataset
pantry = ["apple", "orange", "grape", "apple", "orange",
          "apple", "tomato", "potato", "grape"]

# empty dictionary
pantry_counts = {}

for item in pantry:
  if item in pantry_counts:
    pantry_counts[item] += 1
  else:
    pantry_counts[item] = 1

pantry_counts
```

```
{'apple': 3, 'grape': 2, 'orange': 2, 'potato': 1, 'tomato': 1}
```

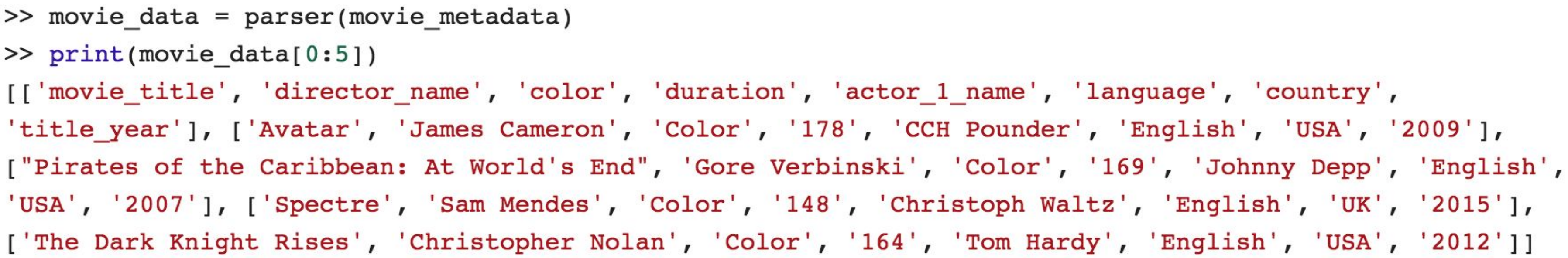
Introdução a funções



movie_title	director_name	color	duration	actor_1_name	language	country	title_year
Avatar	James Cameron	Color	178	CCH Pounder	English	USA	2009
Pirates of the Caribbean: At the World's End	Gore Verbinski	Color	169	Johnny Depp	English	USA	2007
Spectre	Sam Mendes	Color	148	Christoph Waltz	English	UK	2015
The Dark Knight Rises	Christopher Nolan	Color	164	Tom Hardy	English	USA	2012
Star Wars VII: The Force Awakens	JJ Abrams	Color	136	Harrison Ford	English	USA	2015

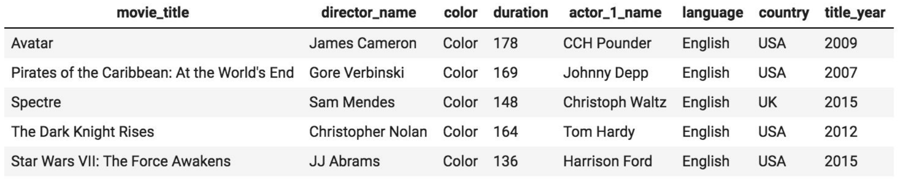
Dados sobre o IMDb

Introdução a funções



```
>> movie_data = parser(movie_metadata)
>> print(movie_data[0:5])
[['movie_title', 'director_name', 'color', 'duration', 'actor_1_name', 'language', 'country',
'title_year'], ['Avatar', 'James Cameron', 'Color', '178', 'CCH Pounder', 'English', 'USA', '2009'],
["Pirates of the Caribbean: At World's End", 'Gore Verbinski', 'Color', '169', 'Johnny Depp', 'English',
'USA', '2007'], ['Spectre', 'Sam Mendes', 'Color', '148', 'Christoph Waltz', 'English', 'UK', '2015'],
['The Dark Knight Rises', 'Christopher Nolan', 'Color', '164', 'Tom Hardy', 'English', 'USA', '2012']]
```

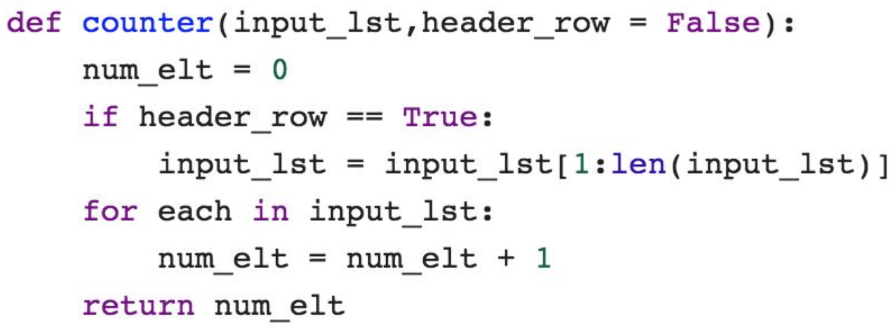
movie_metadata.csv



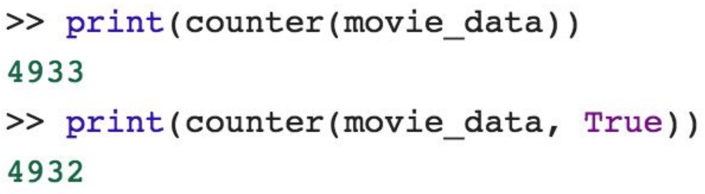
movie_title	director_name	color	duration	actor_1_name	language	country	title_year
Avatar	James Cameron	Color	178	CCH Pounder	English	USA	2009
Pirates of the Caribbean: At the World's End	Gore Verbinski	Color	169	Johnny Depp	English	USA	2007
Spectre	Sam Mendes	Color	148	Christoph Waltz	English	UK	2015
The Dark Knight Rises	Christopher Nolan	Color	164	Tom Hardy	English	USA	2012
Star Wars VII: The Force Awakens	JJ Abrams	Color	136	Harrison Ford	English	USA	2015

parser()

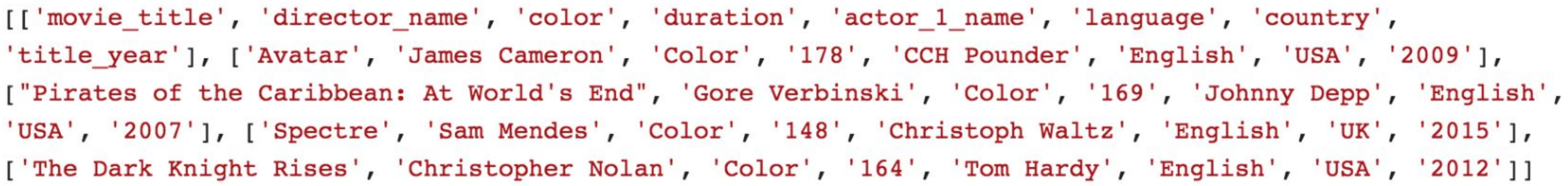
Funções



```
def counter(input_lst,header_row = False):
    num_elt = 0
    if header_row == True:
        input_lst = input_lst[1:len(input_lst)]
    for each in input_lst:
        num_elt = num_elt + 1
    return num_elt
```

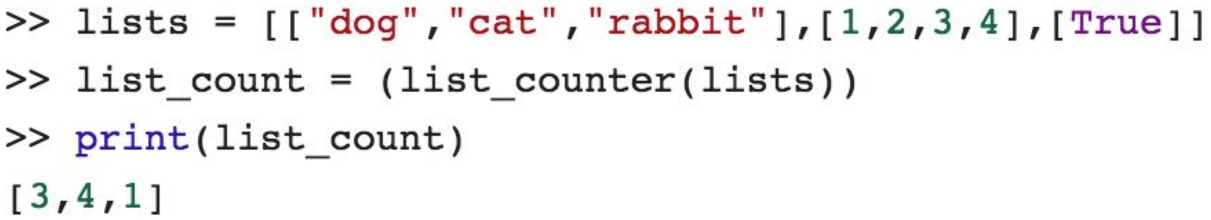


```
>> print(counter(movie_data))
4933
>> print(counter(movie_data, True))
4932
```

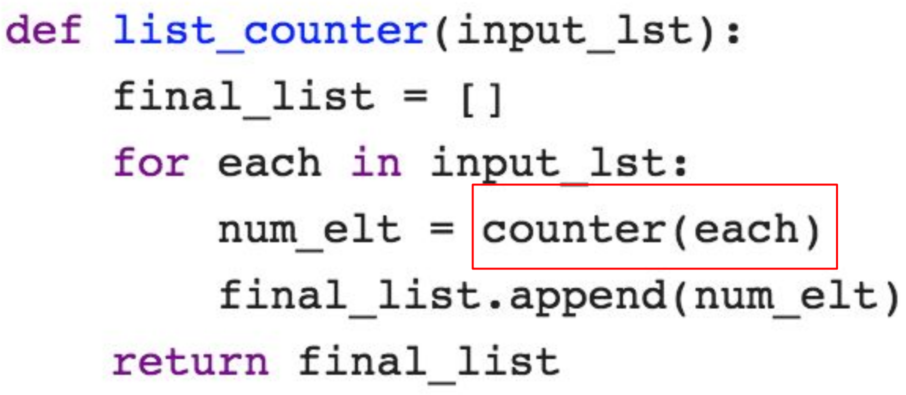


```
[['movie_title', 'director_name', 'color', 'duration', 'actor_1_name', 'language', 'country',
'title_year'], ['Avatar', 'James Cameron', 'Color', '178', 'CCH Pounder', 'English', 'USA', '2009'],
["Pirates of the Caribbean: At World's End", 'Gore Verbinski', 'Color', '169', 'Johnny Depp', 'English',
'USA', '2007'], ['Spectre', 'Sam Mendes', 'Color', '148', 'Christoph Waltz', 'English', 'UK', '2015'],
['The Dark Knight Rises', 'Christopher Nolan', 'Color', '164', 'Tom Hardy', 'English', 'USA', '2012']]
```

Chamando uma função dentro de outra função

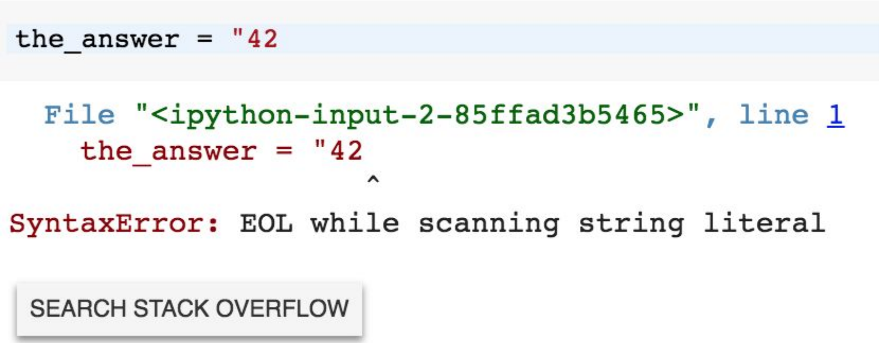


```
>> lists = [["dog","cat","rabbit"],[1,2,3,4],[True]]
>> list_count = (list_counter(lists))
>> print(list_count)
[3,4,1]
```



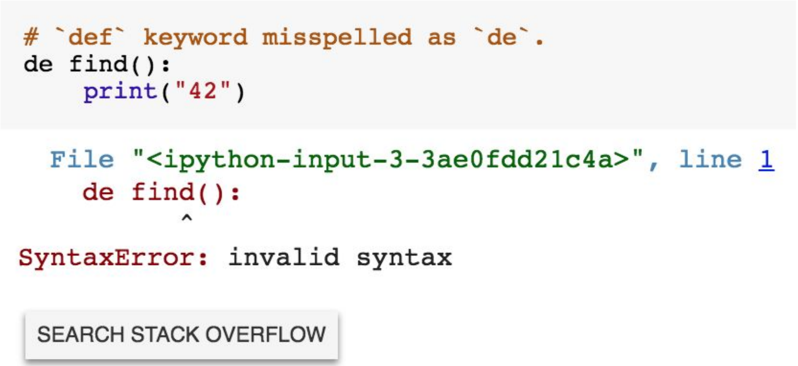
```
def list_counter(input_lst):
    final_list = []
    for each in input_lst:
        num_elt = counter(each)
        final_list.append(num_elt)
    return final_list
```

Depurando erros



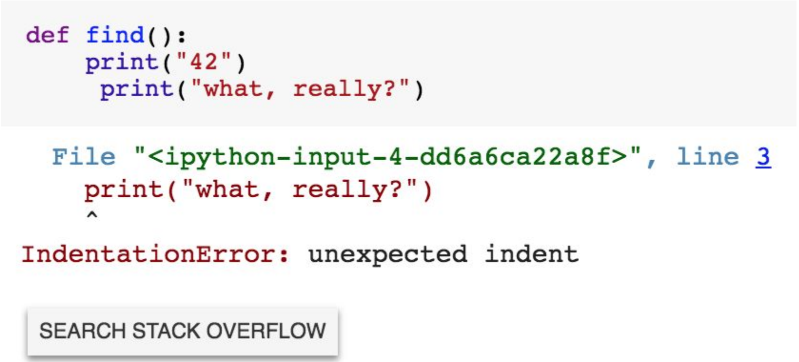
```
the_answer = "42
```

```
  File "<ipython-input-2-85ffad3b5465>", line 1
    the_answer = "42
                     ^
SyntaxError: EOL while scanning string literal
```



```
# `def` keyword misspelled as `de`.
de find():
    print("42")
```

```
  File "<ipython-input-3-3ae0fdd21c4a>", line 1
    de find():
          ^
SyntaxError: invalid syntax
```

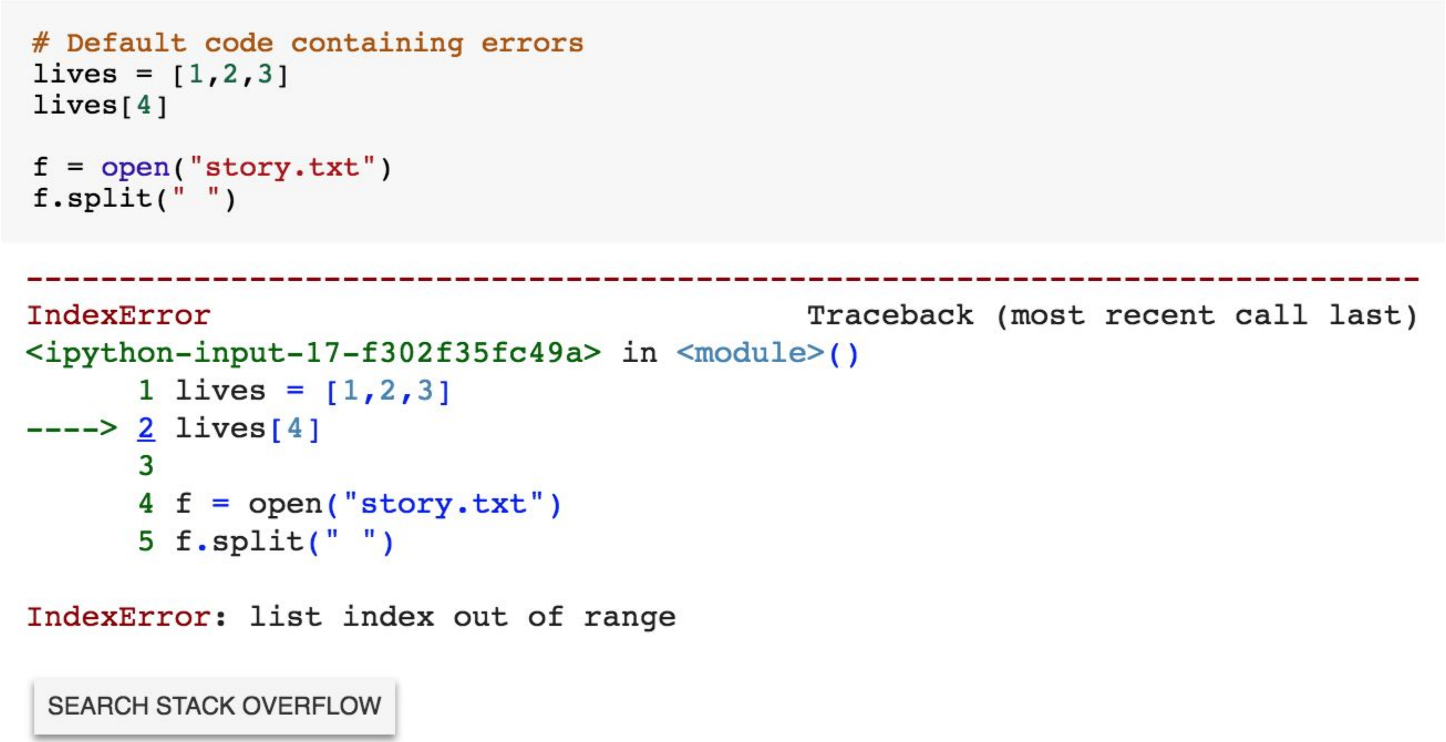


```
def find():
    print("42")
     print("what, really?")
```

```
  File "<ipython-input-4-dd6a6ca22a8f>", line 3
    print("what, really?")
    ^
IndentationError: unexpected indent
```

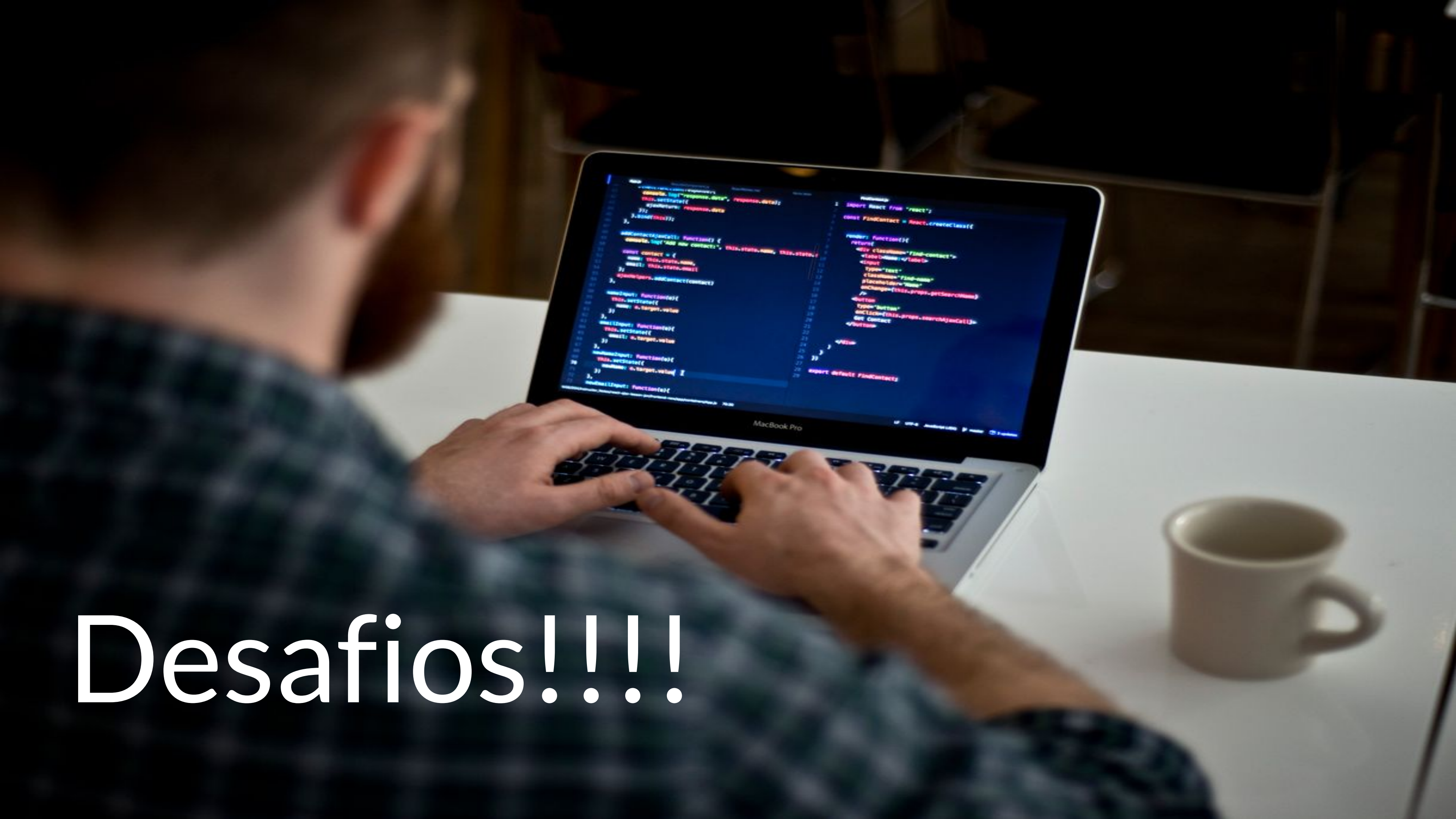
Respirar()
Contar(5)
Ler()

Depurando erros



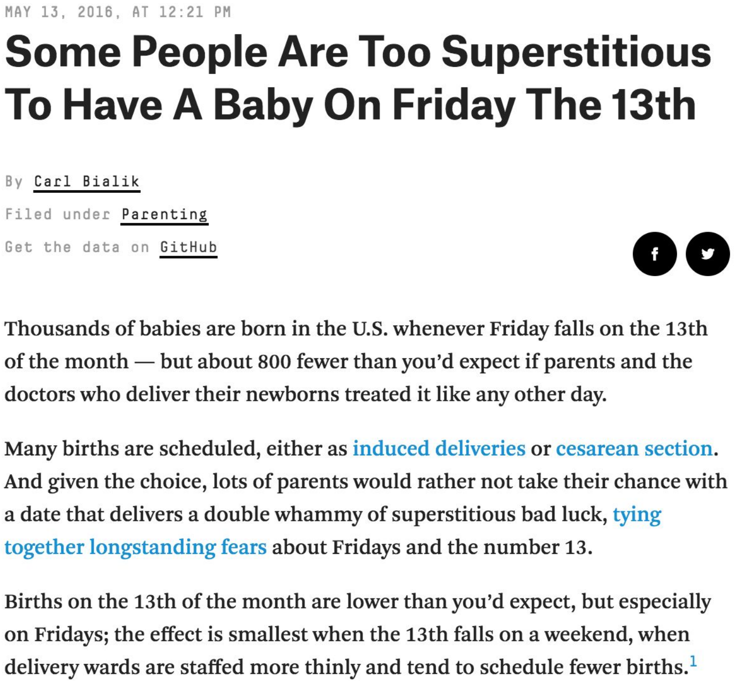
```
# Default code containing errors
lives = [1,2,3]
lives[4]

f = open("story.txt")
f.split(" ")
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-17-f302f35fc49a> in <module>()
      1 lives = [1,2,3]
----> 2 lives[4]
      3 
      4 f = open("story.txt")
      5 f.split(" ")

IndexError: list index out of range
```



Desafios!!!!

Desafio (#nascimento por mês ou dia da semana)



MAY 13, 2016, AT 12:21 PM

Some People Are Too Superstitious To Have A Baby On Friday The 13th

By Carl Bialik

Filed under Parenting

Get the data on GitHub

Thousands of babies are born in the U.S. whenever Friday falls on the 13th of the month — but about 800 fewer than you'd expect if parents and the doctors who deliver their newborns treated it like any other day.

Many births are scheduled, either as induced deliveries or cesarean section. And given the choice, lots of parents would rather not take their chance with a date that delivers a double whammy of superstitious bad luck, tying together longstanding fears about Fridays and the number 13.

Births on the 13th of the month are lower than you'd expect, but especially on Fridays; the effect is smallest when the 13th falls on a weekend, when delivery wards are staffed more thinly and tend to schedule fewer births.[1]

FiveThirtyEight

Desafio (...)

https://github.com/torvalds/linux/

Repositório contém a evolução do Linux nos últimos 13 anos.

```
git log --encoding=latin-1 --pretty="%at,%aN" > log.csv
```

Quantas pessoas contribuíram ao projeto?
Top 10 contribuidores?