



Python e Web scraping

Aprendendo a extrair
dados da Web

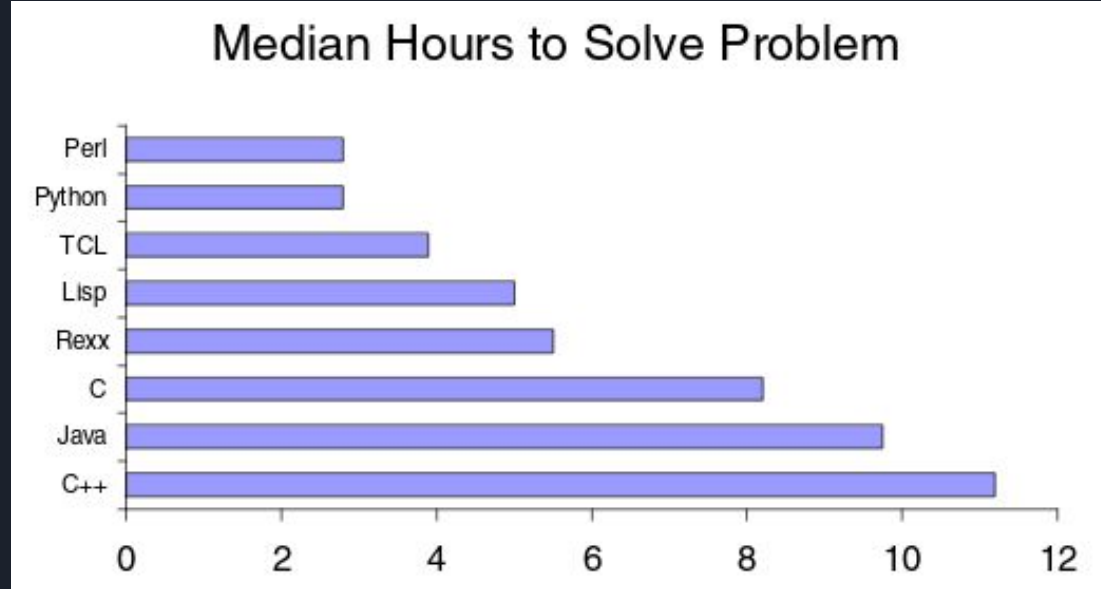
Aula 1 - Introdução ao Python

Felipe Andrade
felipe@potelo.com.br




Python

- Lançada em 1991;
- Foco na produtividade;
- Simples e versátil;
- Tipagem dinâmica;
- Orientada a objetos;
- Funcional;



<https://hackernoon.com/yes-python-is-slow-and-i-dont-care-13763980b5a1>



```
>>> import this
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
```

<https://www.python.org/dev/peps/pep-0020/>

Começando pelo começo

[GO](#)[Socialize](#)[About](#)[Downloads](#)[Documentation](#)[Community](#)[Success Stories](#)[News](#)[Events](#)

Python 3: Fibonacci series up to n

```
>>> def fib(n):
>>>     a, b = 0, 1
>>>     while a < n:
>>>         print(a, end=' ')
>>>         a, b = b, a+b
>>>     print()
>>> fib(1000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987
```


Functions Defined

The core of extensible programming is defining functions. Python allows mandatory and optional arguments, keyword arguments, and even arbitrary argument lists. [More about defining functions in Python 3](#)

[1](#)[2](#)[3](#)[4](#)[5](#)

Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)

<https://www.python.org/>



Python 3.6.3 documentation

Welcome! This is the documentation for Python 3.6.3.

Parts of the documentation:

What's new in Python 3.6?

or all "What's new" documents since 2.0

Tutorial

start here

Library Reference

keep this under your pillow

Language Reference

describes syntax and language elements

Python Setup and Usage

how to use Python on different platforms

Python HOWTOs

in-depth documents on specific topics

Installing Python Modules

installing from the Python Package Index & other sources

Distributing Python Modules

publishing modules for installation by others

Extending and Embedding

tutorial for C/C++ programmers

Python/C API

reference for C/C++ programmers

FAQs

frequently asked questions (with answers!)

<https://docs.python.org/3/>

[GO](#)[Socialize](#)[About](#)[Downloads](#)[Documentation](#)[Community](#)[Success Stories](#)[News](#)[Events](#)

Download the latest version for Windows

[Download Python 3.6.3](#)[Download Python 2.7.14](#)

Wondering which version to use? [Here's more about the difference between Python 2 and 3.](#)

Looking for Python with a different OS? Python for [Windows](#), [Linux/UNIX](#), [Mac OS X](#), [Other](#)

Want to help test development versions of Python? [Pre-releases](#)



<https://www.python.org/downloads/>

Coming in 2017.3

What's New

Features

Docs & Demos

Buy



Python IDE
for Professional Developers

DOWNLOAD NOW

Full-fledged Professional or Free Community

<https://www.jetbrains.com/pycharm/>



```
/* Hello world in C* */
```

```
#include <stdio.h>
```

```
main()
```

```
{
```

```
    printf("Hello World!\n");
```

```
}
```

```
// Hello World in Java
```

```
class HelloWorld {
```

```
    static public void main( String args[] ) {
```

```
        System.out.println( "Hello World!" );
```

```
    }
```

```
}
```

```
# Hello world in Python 3
```

```
print("Hello World")
```

Brincando com os tipos

```
λ Python
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:14:34) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> a = 5
>>> b = 10
>>> a + b
15
>>> type(a)
<class 'int'>
>>> a = "Hello"
>>> b = " World"
>>> a + b
'Hello World'
>>> type(a)
<class 'str'>
>>> c = a == b
>>> c
False
>>> type(c)
<class 'bool'>
>>>
```



Entrando com dados

```
λ Python
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:14:34) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> nome = input("Qual é o seu nome? ")
Qual é o seu nome? Felipe
>>> msg = "Olá " + nome + ", tudo bem?"
>>> print(msg)
Olá Felipe, tudo bem?
>>> a = input("digite um numero: ")
digite um numero: 7
>>> b = input("digite outro numero: ")
digite outro numero: 6
>>> c = a + b
>>> print("A soma de a + b é: "+c)
```



Entrando com dados

```
λ python
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:14:34)
Type "help", "copyright", "credits" or "license" for
>>> a = int(input("digite um numero: "))
digite um numero: 7
>>> b = int(input("digite outro numero: "))
digite outro numero: 6
>>> c = a + b
>>> print("A soma de a + b é: %d" % c)
A soma de a + b é: 13
>>> print(f"A soma de {a} + {b} é: {c}")
A soma de 7 + 6 é: 13
>>>
```



Entrando com dados - Atividade

Pergunte o nome do aluno, peça para ele inserir suas 3 notas, calcule a média e imprima na tela.



Manipulando Strings

```
>>> a = "Hello World"
>>> a[:]
'Hello World'
>>> a[0]
'H'
>>> a[-1]
'd'
>>> a[0:4]
'Hell'
>>> a[:5]
'Hello'
>>> a[7:]
'orld'
>>> a[7:-1]
'orl'
```

Manipulando Strings

```
>>> a.upper()
'HELLO WORLD'
>>> a.lower()
'hello world'
>>> a.capitalize()
'Hello world'
>>> a.replace("World", "Felipe")
'Hello Felipe'
>>> a.split(" ")
['Hello', 'World']
>>> a.split()
['Hello', 'World']
>>> a = "  " + a + "  "
>>> a
'  Hello World  '
>>> a.strip()
'Hello World'
>>> a.rstrip()
'Hello World'
>>> a.lstrip()
'Hello World'
```

Como juntar um array de string através de uma sequência. ex.:
['a', 'b', 'c']
transformar em:
"a - b - c"



Manipulando Strings - Atividade

Pergunte o nome do aluno, peça para ele inserir suas 3 notas de uma só vez separando por vírgula, calcule a média e imprima na tela os 3 primeiros caracteres do nome em maiúsculo com a média.



Condições lógicas

```
>>> a = 5
>>> if a > 0:
...     print("a é maior que 0")
...
a é maior que 0
>>> if a < 0:
...     print("a é menor que 0")
...
>>> if a > 0 and a < 10:
...     print("a está entre 0 e 10")
...
a está entre 0 e 10
>>> if a < 0 or a == 5:
...     print("a é menor que 0 ou igual à 5")
...
a é menor que 0 ou igual à 5
>>> |
```



Condições lógicas

```
>>> a = 7
>>> if a < 0:
...     print("a menor que 0")
... elif a > 0 and a < 5:
...     print("a entre 0 e 5")
... else:
...     print("a maior ou igual à 5")
...
a maior ou igual à 5
>>>
```



Condições lógicas

```
>>> a = "x"
>>> "é um numero" if a.isdigit() else "é uma letra"
'é uma letra'
>>> a = "8"
>>> "é um numero" if a.isdigit() else "é uma letra"
'é um numero'
>>> c = a if a.isdigit() else 0
>>> c
'8'
>>> a = "w"
>>> c = a if a.isdigit() else 0
>>> c
0
>>>
```



Condições lógicas - Atividade

Pergunte o nome do aluno, peça para ele inserir suas 3 notas de uma só vez separando por vírgula, calcule a média e caso seja até 5 imprima “muito ruim”, se for maior que 5 porém menor que 7 imprima “quase lá” e se for maior imprima “parabéns”.



Laço de repetição

```
>>> for x in range(5):  
...     print(x)  
...  
0  
1  
2  
3  
4  
>>> for x in range(3, 7):  
...     print(x)  
...  
3  
4  
5  
6
```

Laço de repetição

```
>>> a = ['pera', 'uva', 'maçã']
>>> for fruta in a:
...     print(fruta)
...
pera
uva
maçã
```

```
>>> for n, fruta in enumerate(a):
...     print(f"{n} - {fruta}")
...
0 - pera
1 - uva
2 - maçã
```

```
>>> for n, fruta in enumerate(a, 1):
...     print(f"{n} - {fruta}")
...
1 - pera
2 - uva
3 - maçã
```

Como imprimir uma mensagem somente se não entrar no loop?



Laço de repetição

```
>>> while True:
...     print(a)
...     a += 1
...     if a < 3:
...         continue
...     break
...
0
1
2
```



Laços de repetição - Atividade

Peça pro usuário digitar um número n , valide se realmente é um número, se não for peça novamente até que seja digitado um número inteiro válido. Com isso Imprima os n primeiros números da sequência fibonacci dizendo se ele é par ou ímpar.



Função

```
>>> def soma(a, b):  
...     print(a+b)  
...  
>>> soma(3, 5)  
8  
>>> def subtrai(a, b):  
...     return a - b  
...  
>>> print(subtrai(8, 5))  
3
```

```
>>> def soma(a: int, b: int) -> int:  
...     return a + b  
...  
>>> x = soma('x', 2)  
Traceback (most recent call last):  
  File "<stdin>", line 1, in <module>  
    x = soma('x', 2)  
  File "<stdin>", line 2, in soma  
TypeError: must be str, not int  
>>>
```



Atividade

Faça uma calculadora com as opções de soma, subtração, multiplicação, divisão e média utilizando funções.



Sugestão de estudo para próxima aula

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>

<https://regexone.com/>

<https://www.json.org/json-pt.html>



Felipe Andrade

felipe@potelo.com.br