

# Programação de Computadores com Python

## Aula 00

História e Operadores  
Matemáticos

# Cronograma de Aulas

03/03/2020	História e Operadores Matemáticos
06/03/2020	Lista de Exercícios 01
10/03/2020	Operadores Lógicos e Estruturas de Decisão
13/03/2020	Lista de Exercícios 02
17/03/2020	Listas e String
20/03/2020	Lista de Exercícios 03
24/03/2020	Estruturas de Repetição
27/03/2020	Lista de Exercícios 04
31/03/2020	Prova 1
07/04/2020	Solução Prova 1

# Surgimento do Python

- A ideia surgiu em 1982 no Centro de Matemática e Ciência da Computação em Amsterdã;
- Em 1987 Guido percebeu a necessidade de uma linguagem para escrever programas intermediários, algo entre o C e o Shell Script;
- 1989 iniciou-se o desenvolvimento da linguagem;
- **Monty Python's Flying Circus e O'Reilly;**



# A Linguagem de Programação Python

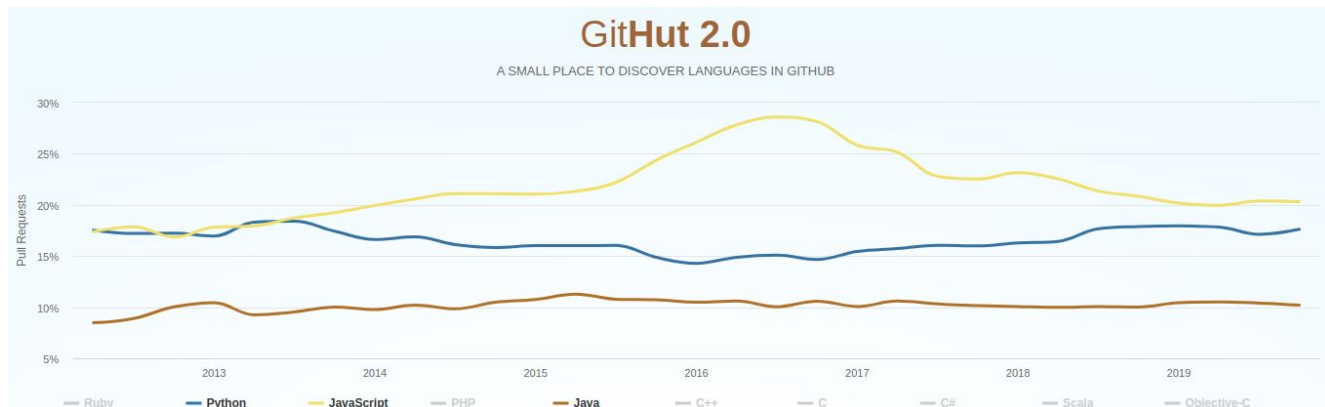
## Programming Language Hall of Fame

The hall of fame listing all "Programming Language of the Year" award winners is shown below. The award is given to the programming language that has the highest rise in ratings in a year.

Year	Winner
2019	🏆 C
2018	🏆 Python
2017	🏆 C

Feb 2020	Feb 2019	Change	Programming Language	Ratings	Change
1	1		Java	17.358%	+1.48%
2	2		C	16.766%	+4.34%
3	3		Python	9.345%	+1.77%

# A Linguagem de Programação Python



Python é uma linguagem de alto nível, interpretada, multiparadigma, multiuso e **OPEN SOURCE**

# Zen Python

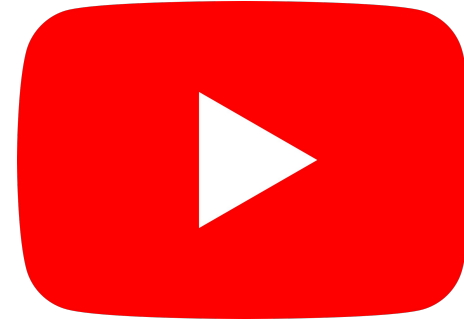
```
ratopythonista@ratopythonista:~$ python3
Python 3.6.9 (default, Nov  7 2019, 10:44:02)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import this
The Zen of Python, by Tim Peters

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!
```

# Para que fins é possível utilizar Python?



# Empresas que Utilizam Python





*“O Python nos permite produzir recursos de manutenção em tempo recorde com um mínimo de desenvolvedores.”*

Cuong Do, Software Architect at Youtube

# Variáveis

Python é uma linguagem fortemente tipada! Entretanto a tipagem é dinâmica, ou seja, não é preciso informar o tipo de variável na declaração

- bool
- int
- float
- str
- list
- dict
- set
- tuple

# Operadores Matemáticos

Python suporta várias operações matemáticas simples por padrão, para coisas mais complexas é possível utilizar a biblioteca *math*

+	Soma
-	Subtração
*	Multiplicação
/	Divisão
//	Divisão Inteira
**	Potenciação
%	Resto da Divisão

```
import math
```

```
raio = input()
```

```
raio, pi = int(raio), math.pi
```

```
area = pi*raio**2
```

```
print(area)
```

Exemplo de Código Python