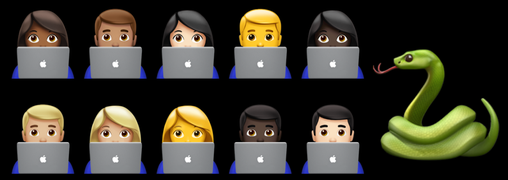
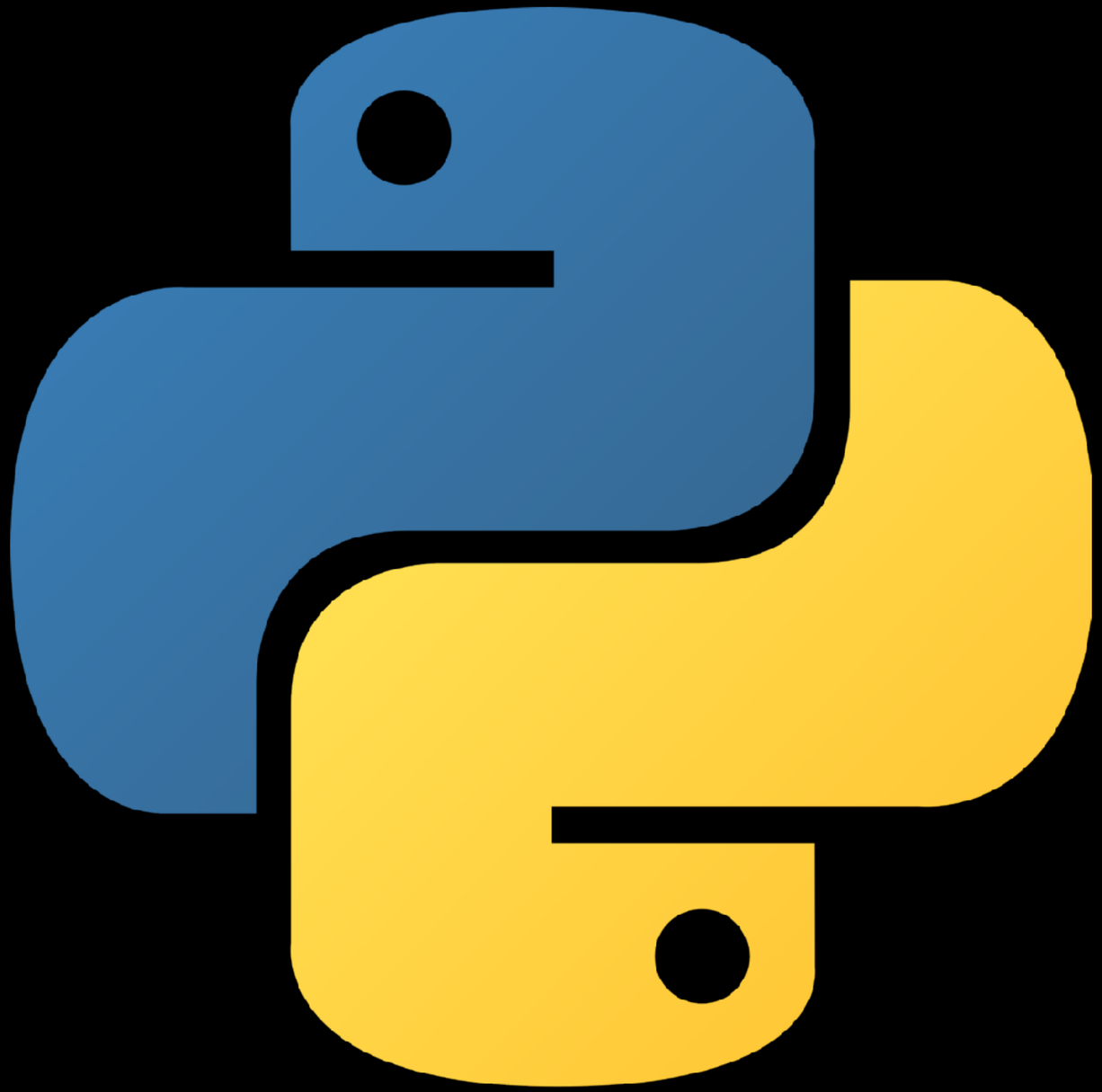
Python para Pessoas (P^3)



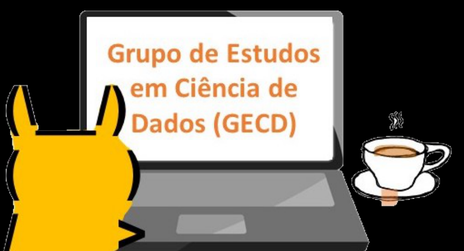
21/09/2019

O que é *Python*?



Receitas

- Antes de começar, vamos pensar em comida
- Quando queremos fazer um novo prato, pegamos a receita.
- No que consiste uma receita?



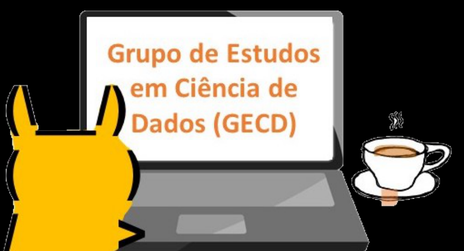
Polenta!

Ingredientes

- 2 Litros de água
- 400g de fubá
- 2 colheres de sopa de manteiga
- 1 colher de sopa de sal

Preparo

1. Em uma panela, leve a água ao fogo e acrescente o sal e a manteiga
2. Quando iniciar a fervura, acrescente o fuba e mexa sem parar para não empelotar.
3. Deixe a polenta cozinhar por 30 minutos em fogo baixo.
4. Despeje a polenta em um refratário.



Fonte: Tudo Gostoso

Porque o nome é *Python*?

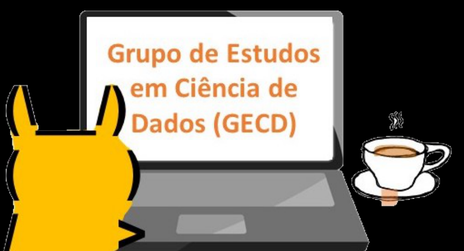
Monty Python's Flying Circus

Objetivos

- O que vamos aprender hoje?
  - Programar em *Python*!

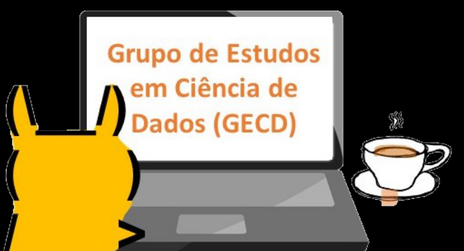
(e lógica de programação)

- O que precisa?
  - Vontade!



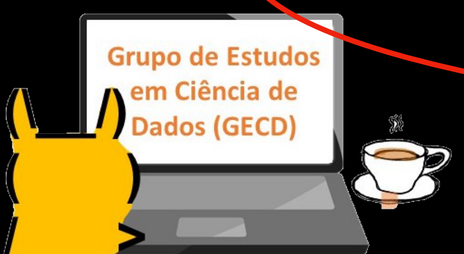
Porquê Python?

- Permite pensar só na resolução do problema
- Interativa
- Muitas bibliotecas!
- Combina com R



Produtividade

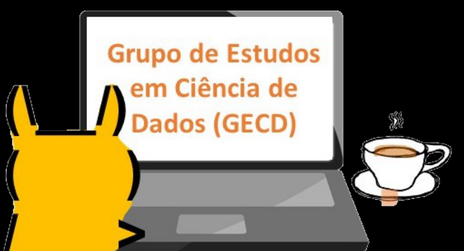
C	2 anos
Java	6 meses
Python	**1 mês**



Fonte: Python P/ Zumbis

Simplicidade

- Programar = Logica + Sintaxe
- Aprender a programar te habilita a aprender outras linguagens
- Com Python não precisamos compilar (algumas outras linguagens precisam)
- Vamos trabalhar com **Python 3**



Interativa

- Vamos começar a praticar, digite `python3` no terminal/VSCode e pressione **enter**:

```
Python 3.7.4 (default, Jul  9 2019, 16:48:28)
[GCC 8.3.1 20190223 (Red Hat 8.3.1-2)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

- O `>>>` é um indicador esperando para a próxima linha de código em python

- Algumas vezes você vai ver `...` como um indicador também.

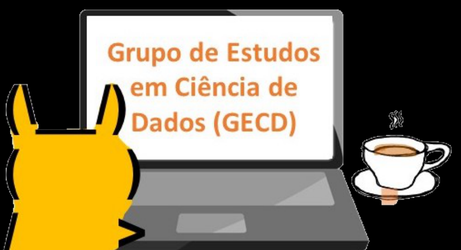
- Para sair, digite `exit()` e pressione **enter**.

Primeiro Programa!

- Abra o Python (sempre vai s… digite `python3`
- Digite a seguinte linha:

```
>>> print('Olá Mundo!')
```

- O que aconteceu?



Números

- Inicie o Python e digite as seguintes somas:

```
>>> 2 + 2
```

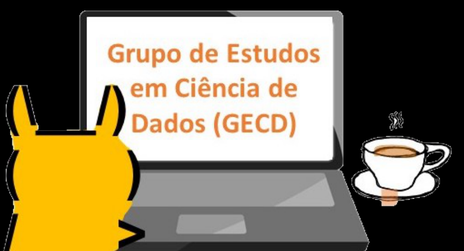
```
>>> 1.5 + 1.25
```

- Subtraia:

```
>>> 4 - 2
```

```
>>> 100 - .5
```

```
>>> 0 - 2
```



Números

- Multiplicação

```
>>> 2 * 3
```

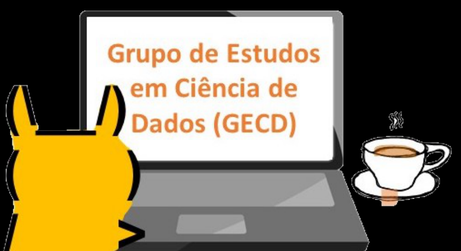
```
>>> 3 * 4.1
```

- Divisão:

```
>>> 4 / 2
```

```
>>> 1 / 2
```

```
>>> 3/4 + 1/4
```



Cadeias

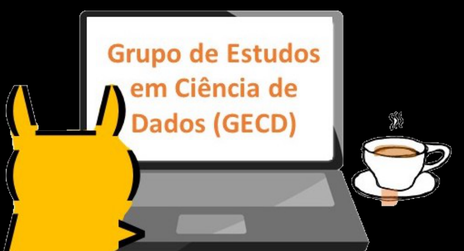
- Cadeias de Caracteres (palavras, frases, etc..) devem estar entre **aspas:**

```
>>> "Olá"
>>> “GECD Foz do Iguaçu"
```

- **Strings** podem ser combinadas:

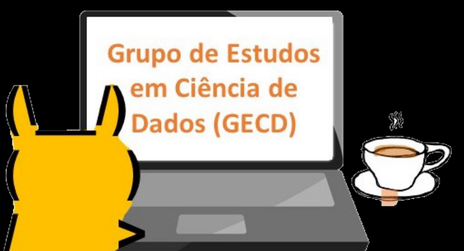
```
>>> “Olá" + “Mundo!"
>>> “Olá " + “Mundo!"
>>> “Olaaaa " + “Mundooooooooo”
```

Strings!



Dicas

- Pressione a seta pra cima ↑ algumas vezes no Python
- O **interpretador** Python salva o histórico do que você digita
- Caso queira reexecutar o comando, é só ir na linha e pressionar **enter**



Intermission

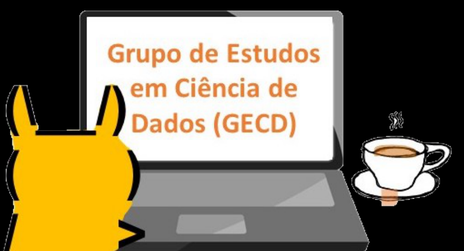
Comandos

- Comandos* consistem em trazer algumas praticidades e/ou facilidades pra usar o python
- Dois exemplos que iremos usar agora:

`>>>print` : mostra na tela uma informação

`>>>type` : mostra qual tipo da informação

***são funções, mas a gente vai simplificar neste momento**



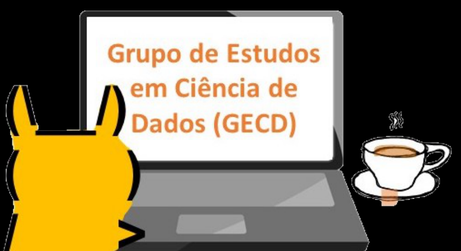
Saída

- Para mostrar algum conteúdo na tela, usamos o comando `print`:

```
>>>print(“Olá Mundo!”)
```

- O que apareceu?
- Qual foi a diferença para o caso **14**?

```
>>>print(“Olá “ + "Mundo!”)
```



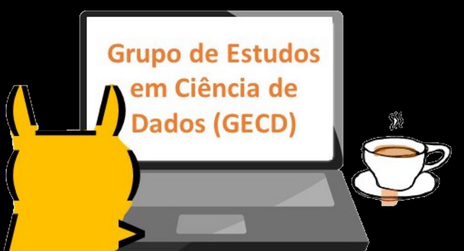
Tipo

- Para mostrar o tipo dos dados na tela, usamos o comando `type`:

```
18.>>>type(1)
```

- O que apareceu?

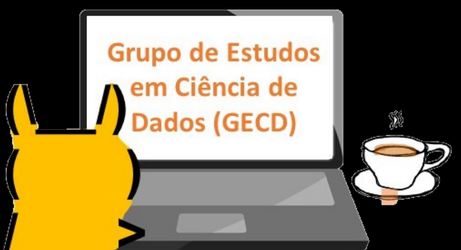
```
19.>>>type(“Olá “ + "Mundo!”)
```



Tipo

- Outros testes para fazer com o `type`:

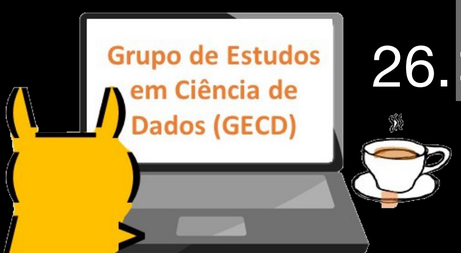
```
>>>type(1.0)

>>>type(2)

>>>type(“2”)

>>>type(3/4)
```



Variáveis

- **Variáveis** são nomes que referenciamos informações (ou um conjunto destas).
- Nomear uma informação/valor é chamado de **atribuição**
- **Os nomes não podem conter espaços** e **DEVEM** começar com uma **LETRA.**

```
>>>type(4)
>>>x = 4
>>>x
>>>type(x)
```



Variáveis

- Note que se você apenas digitar um valor, o interpretador vai mostra na tela.
- Caso você atribua um valor para uma variável, nada é mostrado na tela.

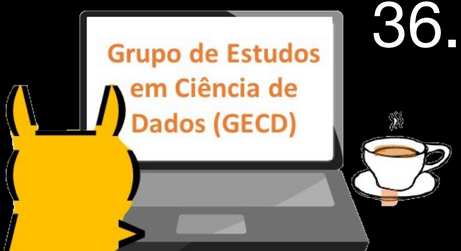
```
>>> 4
>>> x = 4
>>> x
>>> x = 5
>>> x
```

Podemos mudar o valor quando quiser (por isso “variável”)

Variáveis

- Você pode fazer operações com Variáveis:

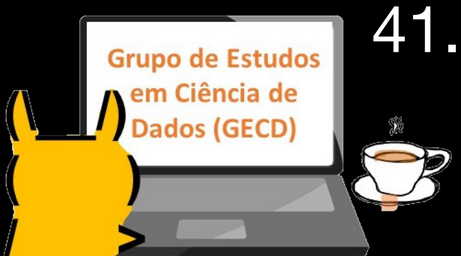
```
>>> x = 3

>>> y = 4

>>> 2 * x - 1 * y

>>> (2 * x) - (1 * y)

>>> soma = x + y
```



Variáveis

- Você pode armazenar palavras/frases em Variáveis:

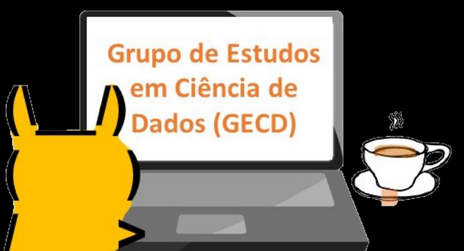
```
>>> p1 = "Olá "
>>> p2 = "Mundo!"
>>> frase = p1 + p2
>>> print(p1 + p2)
>>> print(frase)
```



Variáveis

- Tipos de variáveis que iremos focar agora:

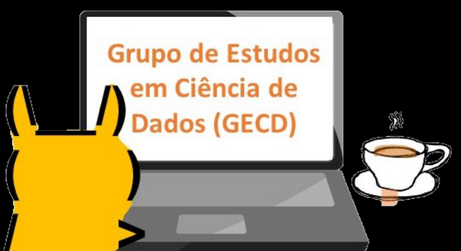
Tipo	Observação
int	Números Inteiros (+ e -)
float	Números reais
str	Cadeia de caracteres
bool	Verdadeiro/Falso



Booleano

- Booleano é um tipo que somente tem dois valores, Verdadeiro/Falso:

```
42.>>> True

43.>>> type(True)

44.>>> False

45.>>> type(False)
```



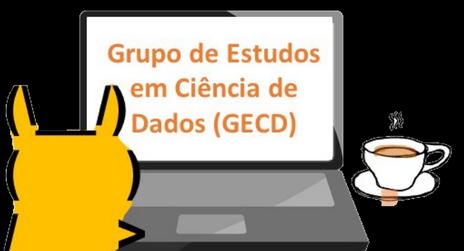
Erros

- O que acontece quando você digita:

```
>>> z
```

```
>>> “Olá" + 1
```

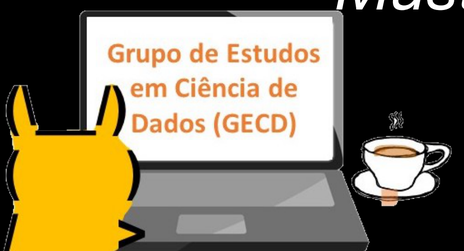
- Um ***traceback*** (mensagem de erro) ocorre com:
  - **Tipo do Erro,** uma “mensagem de ajuda” e o caminho completo (vamos ver mais pra frente o que isso quer dizer).



Erros

```
>>> z
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'z' is not defined
>>> "ola" + 1
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: must be str, not int
>>>
```

- O último ***traceback*** diz:
  - **TypeError**: tipo do erro que ocorre
  - *Must be str, not int*: é a “mensagem de ajuda”



Comparações

- Você pode comparar valores, pra saber se são iguais:

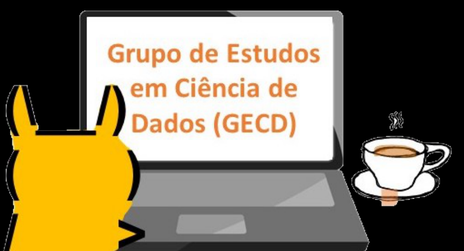
```
>>> 0 == 0

>>> 0 == 1
```

ATENÇÃO!

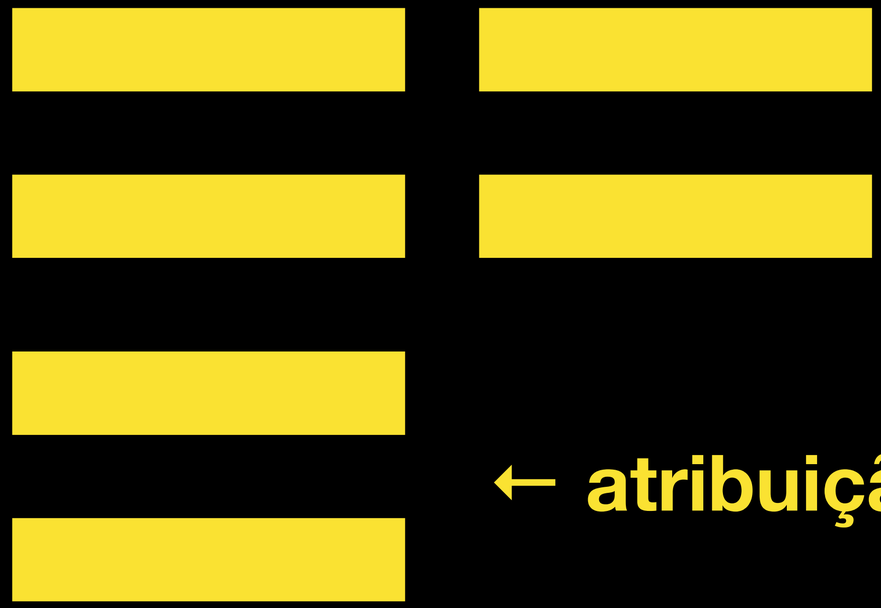
== (igual igual) é para *testar valores* (compará-los)

= (igual) é para **atribuição**

CUIDADO ISSO PODE LEVAR A ERROS/BUGS

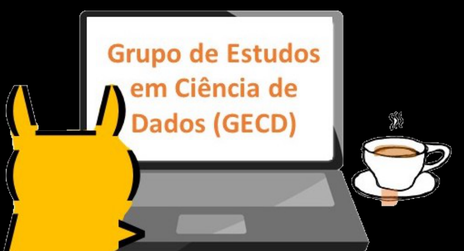


Comparações

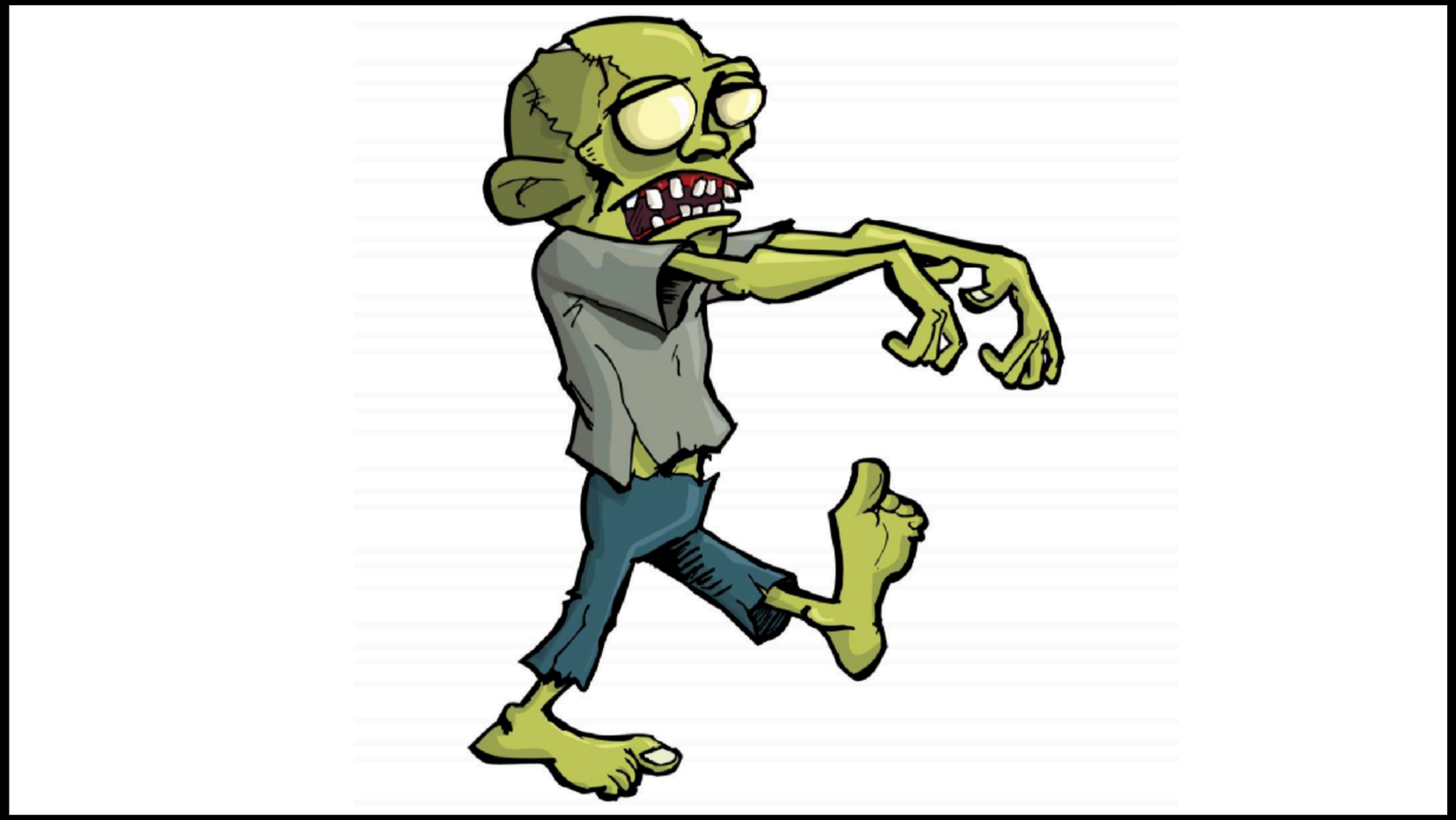


← **comparação** de valor

← **atribuição** de valor



Qual é a velocidade de vôo de uma andorinha africana?



Agora vamos pra apresentação do Mestre Massanori

Diretamente de *Python para Zumbis*

NOBODY EXPECTS THE

SPANISH INQUISITION!


Grupo de Estudos
em Ciência de
Dados (GECD)
Fim da aula de hoje!