

Python



Dia D

Modelagem e Apresentação

Dia 17 de Janeiro

Problemas

- Jogo da Velha
- Predição de Séries Temporais
- Fechadura Eletrônica
- Reconhecimento Facial
- Análise Estatística
- Classificação
- Competição Kaggle
- Modelagem e Simulação Temporal
- Processamento de Imagem
- Detecção de Eventos
- Sensoriamento Micaz
- Pôquer
- Identificação de Objetos
- Alocação de Salas

- Roteamento de Pacotes
- Escalonador de Recurso
- Análise de Sentimento Textual
- Processamento de Texto

Oi

Eu sou Matheus Inacio

Estou aqui para apresentar sobre
Python.

Python Brasil 2018

Você pode entrar em contato por
matheus.inacio@laccan.ufal.br



O que é Python?

Python é uma linguagem poderosa e divertida. Com ela você pode fazer diversas coisas como:

- Construção de sistemas Web com Django, Flask, Pyramid, etc.
- Análise de dados, Inteligência Artificial, Machine Learning e etc com Numpy, Pandas, Matplotlib, etc
- Construção de aplicativos com Kivy e Pybee
- Construção de sistemas desktop com Tkinter, WxPython, etc.



Mais informações em <https://python.org.br/>

O que é VirtualEnv?

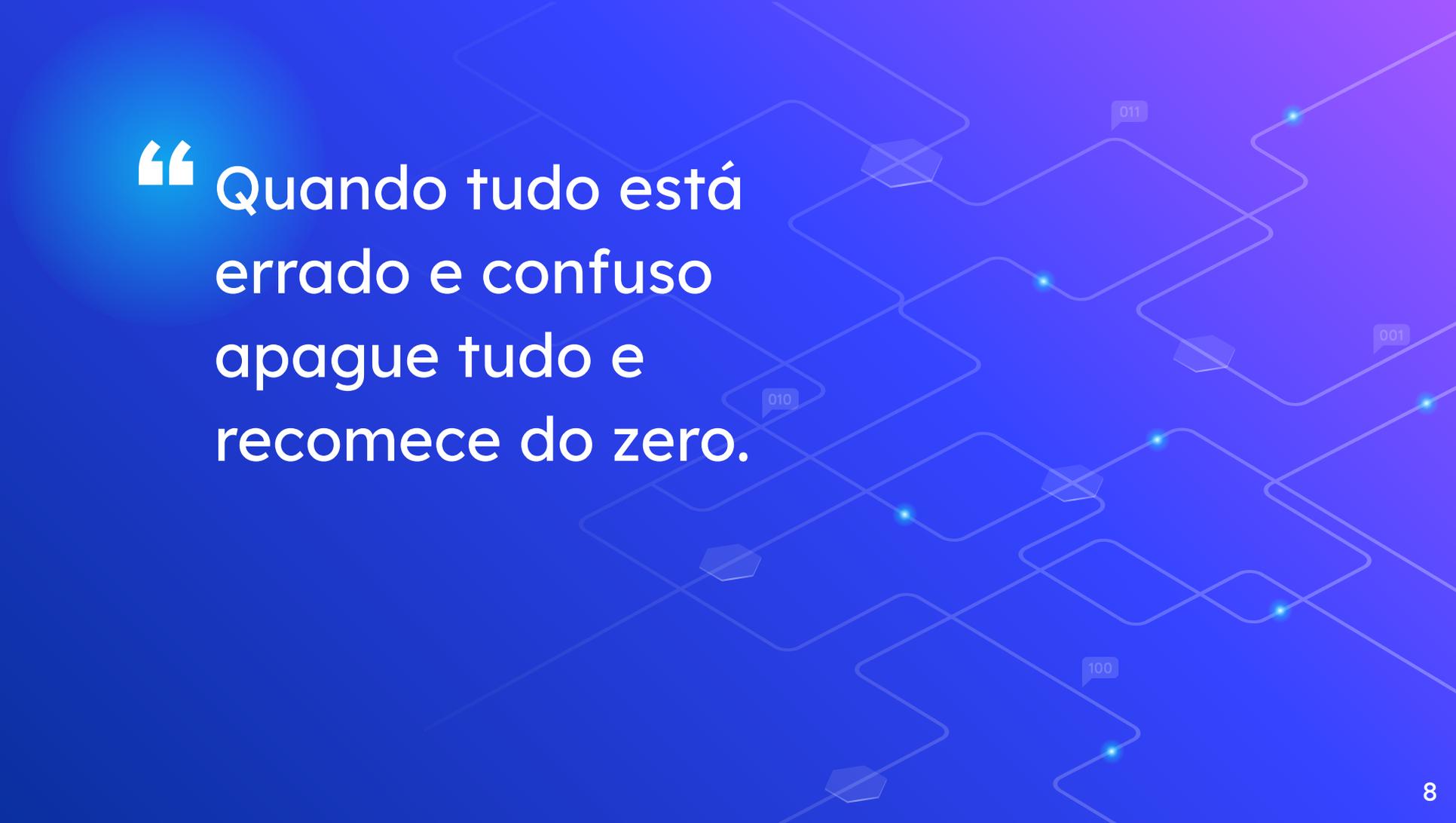
O ideal para projetos Python é que você isole as bibliotecas que você usa, para evitar conflitos entre projetos. Isso é uma boa prática e pode evitar dores de cabeça futuras.

Mais informações em <https://python.org.br/>

Qual Python?

- Python 2 foi o padrão da linguagem por muito tempo.
- Python 3 introduziu algumas mudanças que quebraram a compatibilidade com a versão anterior o que criou a necessidade de se manter duas versões da linguagem.
- Python 2 will not be maintained past 2020
- Python 3 está constantemente evoluindo e recebendo novas funcionalidades, que não estarão presentes na versão anterior.

Mais informações em <https://python.org.br/>



“ Quando tudo está
errado e confuso
apague tudo e
recomece do zero.

1. Hello World

Let's start with course.



Representar um Objeto

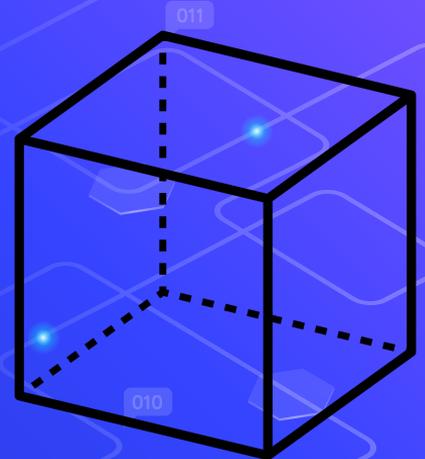


Representar um Objeto

- ⬡ Características?
 - Quantitativo ou Qualitativo?

Representar um Objeto

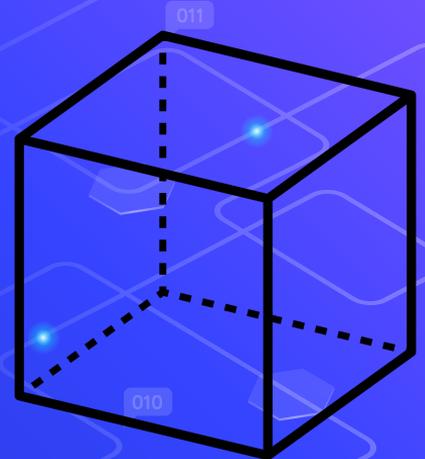
- ⬡ Características?
 - Quantitativo ou Qualitativo?



Representar um Objeto

- ⬡ Características?
 - Quantitativo ou Qualitativo?

- ⬡ Altura

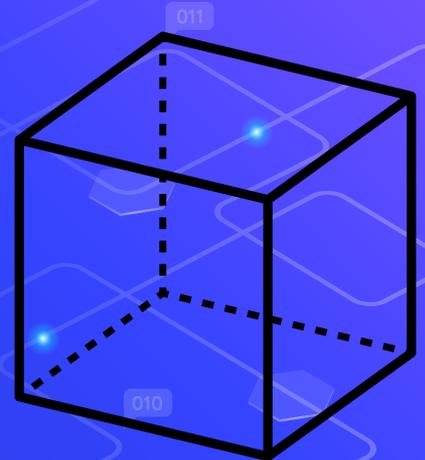


Representar um Objeto

- Características?
 - Quantitativo ou Qualitativo?

○ Altura

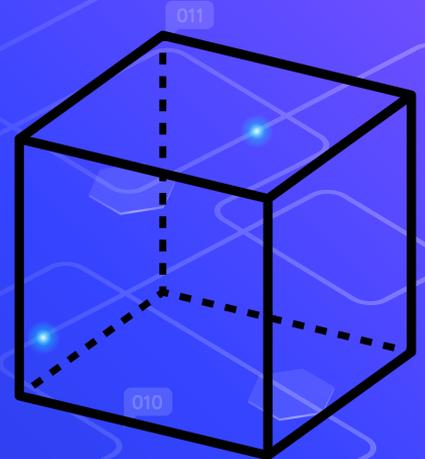
○ Largura



Representar um Objeto

- Características?
 - Quantitativo ou Qualitativo?

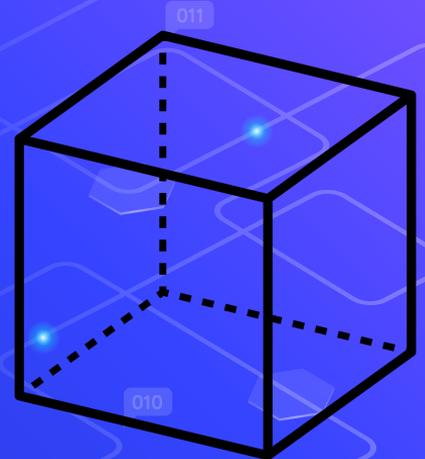
- Altura
- Largura
- Profundidade



Representar um Objeto

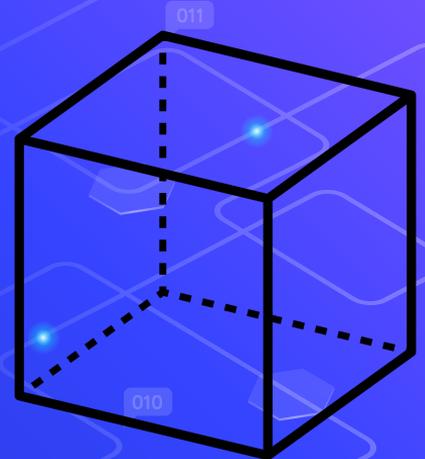
- Características?
 - Quantitativo ou Qualitativo?

- Altura
- Largura
- Profundidade
- Centro



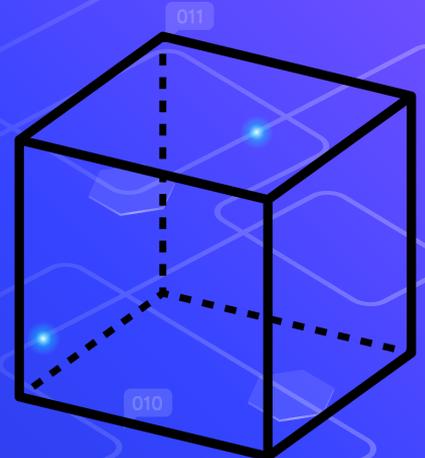
Instancia do Objeto

```
1 height = 10
2 weight = 10
3 deep = 10
4 center_x = 0
5 center_y = 0
6 center_z = 0
7 |
```



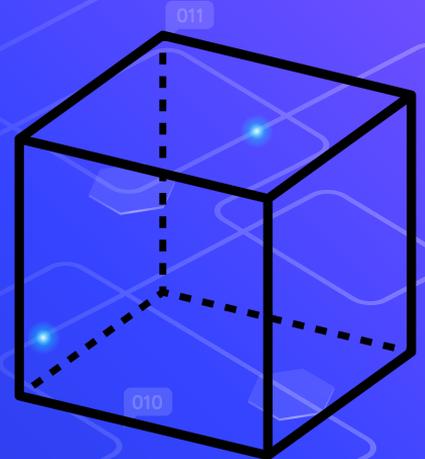
Instancia do Objeto

```
1 height = 10
2 weight = 10
3 deep = 10
4 center_x = 0
5 center_y = 0
6 center_z = 0
7
8 height = 10
9 weight = 10
10 deep = 10
11 center_x = 0
12 center_y = 2
13 center_z = 0
14
15
```



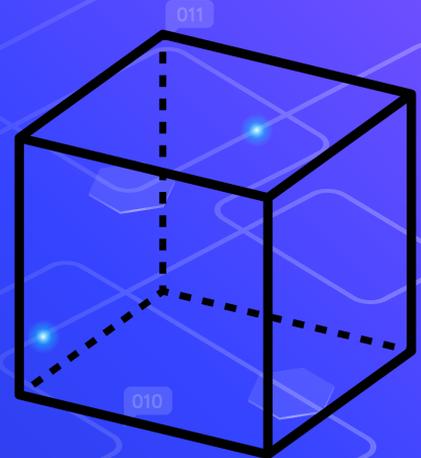
Instancia do Objeto

- Como adicionar **cor** ao nosso quadrilátero?
 - Reescrever todos?
 - Mas só temos dois
 - E se fossem 1000?



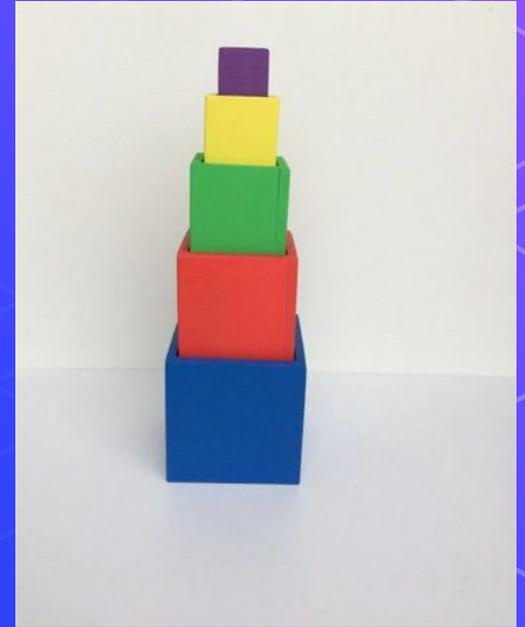
Classe com cor

```
1 class Quad:
2     def __init__(self, height, weight, deep, center):
3         self.height = height
4         self.weight = weight
5         self.deep = deep
6         self.center = center
7         self.color = "#70DB93"
8
9
10 class Center:
11     def __init__(self, x, y, z):
12         self.x = x
13         self.y = y
14         self.z = z
15
16
17 q1 = Quad(10, 10, 10, Center(0, 0, 0))
18 q2 = Quad(10, 10, 10, Center(0, 2, 0))
19
20 q2.color = "#000000"
21
22
```



Instancia do Objeto

- Enderson quer empilhar os cubo do maior para o menor, como podemos fazer isso?



Ordenando cubos

```
1 class Quad:
2     def __init__(self, height, weight, deep, center):
3         self.height = height
4         self.weight = weight
5         self.deep = deep
6         self.center = center
7         self.color = "#70D893"
8
9
10 class Center:
11     def __init__(self, x, y, z):
12         self.x = x
13         self.y = y
14         self.z = z
15
16
17 q1 = Quad(10, 10, 10, Center(0, 0, 0))
18 q2 = Quad(10, 10, 10, Center(0, 2, 0))
19 q3 = Quad(12, 10, 10, Center(0, 0, 0))
20 q4 = Quad(10, 3, 10, Center(0, 2, 0))
21 q5 = Quad(111, 1, 10, Center(0, 0, 0))
22 q6 = Quad(10, 1, 10, Center(0, 2, 0))
23
24 q2.color = "#000000"
25
26 arr = [q1, q2, q3, q4, q5, q6]
27
28 n = len(arr)
29
30 for i in range(n):
31     for j in range(0, n-i-1):
32         if arr[j].deep*arr[j].height*arr[j].weight > arr[j+1].deep*arr[j+1].height*arr[j+1].weight :
33             arr[j], arr[j+1] = arr[j+1], arr[j]
34
35
```

Ordenando cubos

```
1 class Quad:
2     def __init__(self, height, weight, deep, center):
3         self.height = height
4         self.weight = weight
5         self.deep = deep
6         self.center = center
7         self.color = "#70DB93"
8
9     def area(self):
10        | return self.deep * self.height * self.weight
11
12    def __lt__(self, other):
13        | return self.area < other.area()
14
15
16 class Center:
17    def __init__(self, x, y, z):
18        self.x = x
19        self.y = y
20        self.z = z
21
22
23 q1 = Quad(10, 10, 10, Center(0, 0, 0))
24 q2 = Quad(10, 10, 10, Center(0, 2, 0))
25 q3 = Quad(12, 10, 10, Center(0, 0, 0))
26 q4 = Quad(10, 3, 10, Center(0, 2, 0))
27 q5 = Quad(111, 1, 10, Center(0, 0, 0))
28 q6 = Quad(10, 1, 10, Center(0, 2, 0))
29
30 q2.color = "#000000"
31
32 arr = [q1, q2, q3, q4, q5, q6]
33
34 arr.sort()
35
36 |
37
```

Ordenando cubos

```
1 class Quad:
2     def __init__(self, height, weight, deep, center):
3         self.height = height
4         self.weight = weight
5         self.deep = deep
6         self.center = center
7         self.color = "#70DB93"
8
9     def area(self):
10        | return self.deep * self.height * self.weight
11
12    def __lt__(self, other):
13        | return self.area < other.area()
14
15
16 class Center:
17    def __init__(self, x, y, z):
18        self.x = x
19        self.y = y
20        self.z = z
21
22
23 q1 = Quad(10, 10, 10, Center(0, 0, 0))
24 q2 = Quad(10, 10, 10, Center(0, 2, 0))
25 q3 = Quad(12, 10, 10, Center(0, 0, 0))
26 q4 = Quad(10, 3, 10, Center(0, 2, 0))
27 q5 = Quad(111, 1, 10, Center(0, 0, 0))
28 q6 = Quad(10, 1, 10, Center(0, 2, 0))
29
30 q2.color = "#000000"
31
32 arr = [q1, q2, q3, q4, q5, q6]
33
34 arr.sort()
35
36
37
```

```
1 class Quad:
2     def __init__(self, height, weight, deep, center):
3         self.height = height
4         self.weight = weight
5         self.deep = deep
6         self.center = center
7         self.color = "#70DB93"
8
9
10 class Center:
11    def __init__(self, x, y, z):
12        self.x = x
13        self.y = y
14        self.z = z
15
16
17 q1 = Quad(10, 10, 10, Center(0, 0, 0))
18 q2 = Quad(10, 10, 10, Center(0, 2, 0))
19 q3 = Quad(12, 10, 10, Center(0, 0, 0))
20 q4 = Quad(10, 3, 10, Center(0, 2, 0))
21 q5 = Quad(111, 1, 10, Center(0, 0, 0))
22 q6 = Quad(10, 1, 10, Center(0, 2, 0))
23
24 q2.color = "#000000"
25
26 arr = [q1, q2, q3, q4, q5, q6]
27
28 n = len(arr)
29
30 for i in range(n):
31     for j in range(0, n-i-1):
32         if arr[j].deep*arr[j].height*arr[j].weight > arr[j+1].deep*arr[j+1].height*arr[j+1].weight :
33             arr[j], arr[j+1] = arr[j+1], arr[j]
34
35
```

Overloading Operators

Operator ad-hoc Polymorphism

A sobrecarga, no contexto da programação, refere-se à capacidade de uma função ou operador se comportar de maneiras diferentes, dependendo dos parâmetros que são passados para a função ou dos operandos nos quais o operador atua.

Operators

Common Syntax	Special Method Form
$a + b$	<code>a.__add__(b)</code> ; alternatively <code>b.__radd__(a)</code>
$a - b$	<code>a.__sub__(b)</code> ; alternatively <code>b.__rsub__(a)</code>
$a * b$	<code>a.__mul__(b)</code> ; alternatively <code>b.__rmul__(a)</code>
a / b	<code>a.__truediv__(b)</code> ; alternatively <code>b.__rtruediv__(a)</code>
$a // b$	<code>a.__floordiv__(b)</code> ; alternatively <code>b.__rfloordiv__(a)</code>
$a \% b$	<code>a.__mod__(b)</code> ; alternatively <code>b.__rmod__(a)</code>
$a ** b$	<code>a.__pow__(b)</code> ; alternatively <code>b.__rpow__(a)</code>
$a \ll b$	<code>a.__lshift__(b)</code> ; alternatively <code>b.__rlshift__(a)</code>
$a \gg b$	<code>a.__rshift__(b)</code> ; alternatively <code>b.__rrshift__(a)</code>
$a \& b$	<code>a.__and__(b)</code> ; alternatively <code>b.__rand__(a)</code>
$a \wedge b$	<code>a.__xor__(b)</code> ; alternatively <code>b.__rxor__(a)</code>
$a b$	<code>a.__or__(b)</code> ; alternatively <code>b.__ror__(a)</code>
$a += b$	<code>a.__iadd__(b)</code>
$a -= b$	<code>a.__isub__(b)</code>
$a *= b$	<code>a.__imul__(b)</code>
...	...
$+a$	<code>a.__pos__()</code>
$-a$	<code>a.__neg__()</code>
$\sim a$	<code>a.__invert__()</code>
$\text{abs}(a)$	<code>a.__abs__()</code>
$a < b$	<code>a.__lt__(b)</code>
$a \leq b$	<code>a.__le__(b)</code>
$a > b$	<code>a.__gt__(b)</code>
$a \geq b$	<code>a.__ge__(b)</code>
$a == b$	<code>a.__eq__(b)</code>
$a != b$	<code>a.__ne__(b)</code>

Pessoas

Nome	Idade	Peso	Sexo	Nascimento	Email
P1	10	50	M	01/01/2010	p1@a.co
P2	30	80	F	01/01/1990	p2@a.co
P3	5	24	F	01/01/2015	p3@a.co

Atividade 1

Implementar classe *Pessoa*, numa lista com 10 pessoas, depois deve ordenar em pelo primeiro nome e em caso de nome igual, use idade.

20 minutos.

Base de dados

Conjunto de dados relacionados, organizados de forma a permitir a recuperação da informação. Armazenadas por meios ópticos ou magnéticos como discos e podem ser acessadas local ou remotamente.

Base de datos

⬡ json

⬡ csv

⬡ xlsx

Save

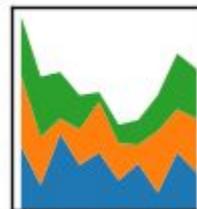
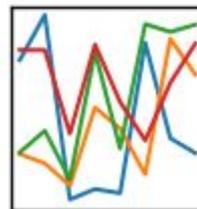
```
1 import pandas as pd
2
3 class Ponto(object):
4     def __init__(self, x, y):
5         self.x = x
6         self.y = y
7
8     def to_dict(self):
9         return {
10            'x': self.x,
11            'y': self.y,
12        }
13
14
15 arr = [Ponto(1, 1), Ponto(0, 0)]
16
17 df = pd.DataFrame.from_records([s.to_dict() for s in arr])
18
19 df.to_csv('test.csv', sep='\t', encoding='utf-8', index=False)
20
21
```

O que é Pandas?

O pandas é uma biblioteca de código aberto, licenciada por BSD, que fornece estruturas de dados de alto desempenho e fáceis de usar, uma ferramentas de análise de dados para a linguagem de programação Python .

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



Atividade 2

Adaptar save.

10 minutos.



Lab 1

1. Crie um gráfico de dispersão da variável peso em relação ao peso desejado. Defina a relação entre essas duas variáveis.
2. Vamos considerar uma nova variável: a diferença entre o peso desejado (`wtdesire`) e o peso atual (`weight`). Crie esta nova variável subtraindo as duas colunas na base de dados e atribuindo-as a um novo objeto chamado `wdiff`.
3. Que tipo de dado está contido na variável `wdiff`? Se uma observação de `wdiff` é 0, o que isso implica com relação ao peso atual e desejado de uma pessoas? E se o valor de `wdiff` for positivo ou negativo?
4. Descreva a distribuição de `wdiff` em termos de seu centro, forma e variação, incluindo qualquer gráfico que você usar. O que isso nos diz sobre como as pessoas se sentem a respeito do seu peso atual?
5. Utilizando sumários numéricos e um gráfico de caixas lado-a-lado, determine se homens tendem a ver seu peso diferentemente das mulheres.
6. Agora chegou a hora de usar a criatividade. Encontre a média e o desvio padrão de `weight` e determine qual a proporção de pesos que estão a um desvio padrão da média.

```
import pandas as pd

# Create a DataFrame
df1 = { 'Name': ['George', 'Andrea', 'micheal',
                'maggie', 'Ravi', 'Xien', 'Jalpa'],
        'score1': [62, 47, 55, 74, 32, 77, 86],
        'score2': [45, 78, 44, 89, 66, 49, 72]}

df1 = pd.DataFrame(df1, columns= ['Name', 'score1', 'score2'])

print("Given DataFrame :\n", df1)

# getting Difference
df1['Score_diff'] = df1['score1'] - df1['score2']
print("\nDifference of score1 and score2 :\n", df1)
```

```
> Given DataFrame :
   Name  score1  score2
0  George     62     45
1  Andrea     47     78
2  micheal    55     44
3  maggie     74     89
4   Ravi     32     66
5   Xien     77     49
6  Jalpa     86     72

Difference of score1 and score2 :
   Name  score1  score2  Score_diff
0  George     62     45         17
1  Andrea     47     78        -31
2  micheal    55     44         11
3  maggie     74     89        -15
4   Ravi     32     66        -34
5   Xien     77     49         28
6  Jalpa     86     72         14
```

Pessoas



Nossa primeira base de dados

Thanks!

Any questions?

