



Aula 1 – Intermediate

2022.1 – Grupo 1

Dicionários

Conjunto **chave-valor**

SINTAXE

```
dicionário = {chave1:valor1, chave2:valor2, ...}
```

ex:

```
calçado= {'Joao': 36, 'Maria':34}
```

ACESSAR VALORES

```
print(dicionário['chave'])
```

ex:

```
print(calçado['Maria'])
```

34

ADICIONANDO VALORES

```
dicionário ['chave'] = valor
```

ex:

```
calçado ['Luís'] = 37
```

DELETANDO VALORES

```
del(dicionário['chave'])
```

ex:

```
del(dicionário['chave'])
```

Matplotlib

Line Plot

Como criar

```
import matplotlib.pyplot  
as plt  
  
plt.plot(x,y)  
plt.show()
```

Uso

tendências e movimentos ao longo do tempo

Scatter Plot

Como criar

```
import matplotlib.pyplot  
as plt  
  
plt.scatter(x,y)  
plt.show()
```

Uso

relação entre **causa e efeito** entre duas variáveis numéricas

Histogram

Como criar

```
import matplotlib.pyplot  
as plt  
  
plt.hist(data,bins)  
plt.show()
```

Uso

distribuição dos dados/**frequência**

Customização: `plt.xlabel()`, `plt.ylabel()`, `plt.title()`, `plt.xticks()`, `plt.yticks()`, `plt.grid(True)`

NumPy

IMPORTANDO O NUMPY

```
import numpy as np
```

SINTAXE

```
variavel_lista = np.array([])
```

ex:

```
variavel_lista = np.array([1,2,3])
```

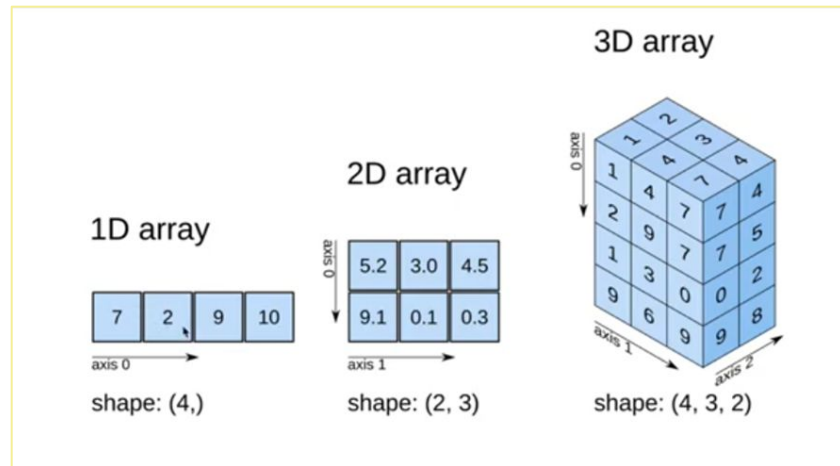


Diagrama de Tipos de Arranjos

FUNÇÕES

```
np.função()
variavel.função()
```

ex:

```
array_1.size
np.delete
```

Funções Estatísticas e Álgebra Linear

NumPy

Método/Função*	Descrição
<code>sum</code>	Soma todos os elementos de um <i>array</i> ou ao longo de um eixo.
<code>mean</code>	Realiza a média aritmética de todos os elementos do <i>array</i> ou ao longo de um eixo.
<code>std</code>	Calcula o desvio padrão de todos os elementos do <i>array</i> ou ao longo de um eixo.
<code>min, max</code>	Verifica o valor mínimo e máximo entre todos os elementos do <i>array</i> ou ao longo de um eixo.
<code>argmin, argmax</code>	Encontra os índices dos elementos mínimo e máximo, respectivamente.
<code>det</code>	Calcula o determinante de uma matriz.
<code>eig</code>	Calcula os autovalores e os autovetores de uma matriz quadrada.
<code>inv</code>	Calcula a inversa de uma matriz quadrada.
<code>solve</code>	Resolve o sistema linear $Ax=b$ para x , em que A é uma matriz quadrada.
<code>fft</code>	Calcula a transformada discreta de Fourier para uma dimensão.

Funções Unárias

NumPy

Função unária	Descrição
<code>abs, fabs</code>	Calcula o valor absoluto de números inteiros, de ponto flutuante e complexos para todos os elementos.
<code>sqrt</code>	Calcula a raiz quadrada de cada elemento do <i>array</i> .
<code>square</code>	Calcula o quadrado (x^2) de cada elemento do <i>array</i> .
<code>exp</code>	Calcula o exponencial (e^x) de cada elemento do <i>array</i> .
<code>log</code>	Calcula o logaritmo natural (base e) de todos os elementos.
<code>rint</code>	Arredonda os elementos para o inteiro mais próximo, preservando o <i>dtype</i> .
<code>isnan</code>	Retorna um array booleano indicando se cada elemento é vazio - NaN (do inglês <i>Not a Number</i>).
<code>cos, cosh, sin, sinh, tan, tanh</code>	Funções trigonométricas regulares e hiperbólicas.

Funções Binárias

NumPy

Função binária	Descrição
<code>add</code>	Realiza a soma dos elementos correspondentes em <i>arrays</i> .
<code>subtract</code>	Realiza a subtração dos elementos do segundo <i>array</i> do primeiro.
<code>multiply</code>	Realiza a multiplicação vetorial entre os <i>arrays</i> .
<code>divide</code>	Realiza a divisão dos elementos dos <i>arrays</i> .
<code>power</code>	Eleva os elementos do primeiro <i>array</i> pela potência dos elementos equivalentes do segundo <i>array</i> .
<code>greater, greater_equal, less, less_equal, equal, not_equal</code>	Faz uma comparação para todos os elementos resultando em um array booleano (equivalente aos operadores relacionais <code>></code> , <code>>=</code> , <code><</code> , <code><=</code> , <code>==</code> e <code>!=</code>).

Simulações em Python

MÓDULO RANDOM

```
import numpy as np
```

- **Módulo do Numpy**
- **Geração de números aleatórios**

```
np.random.rand()
```

```
np.random.randint(1,7)
```

SEED

- **Ponto inicial de um algoritmo**
- **Garante reprodutibilidade**

```
np.random.seed(0)
```

```
np.random.randint(1,7)
```

```
5
```


Simulações em Python

PROBABILIDADE

- Abstrata
- Concreta

USANDO LAÇOS

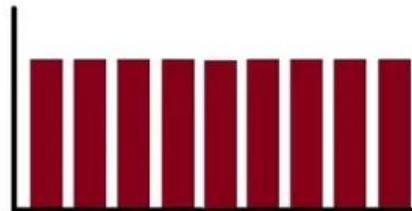
```
for c in range(100):  
    print(np.random.randint(1,7))
```

```
for c in range(10000):  
    print(np.random.randint(1,7))
```

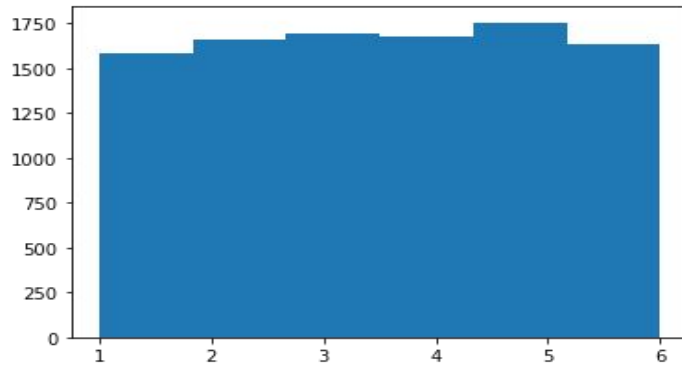
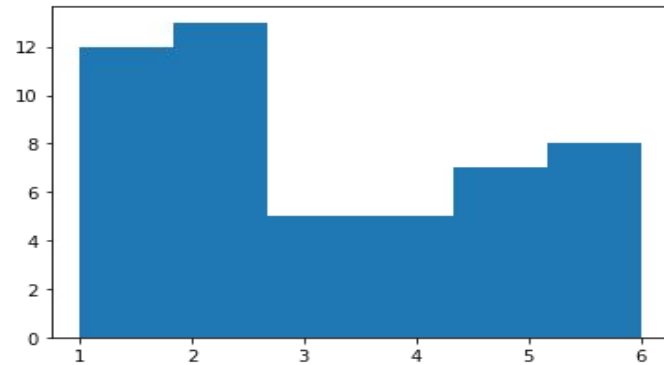
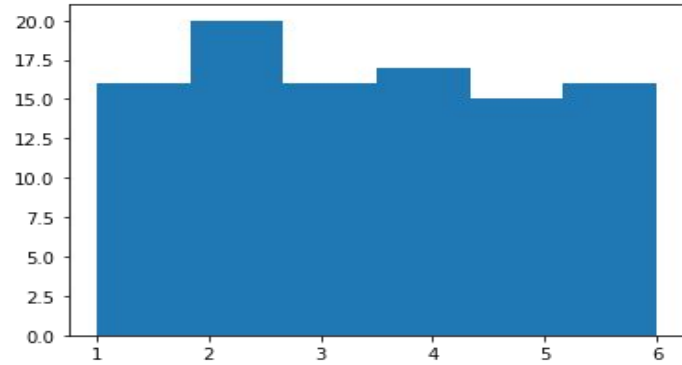
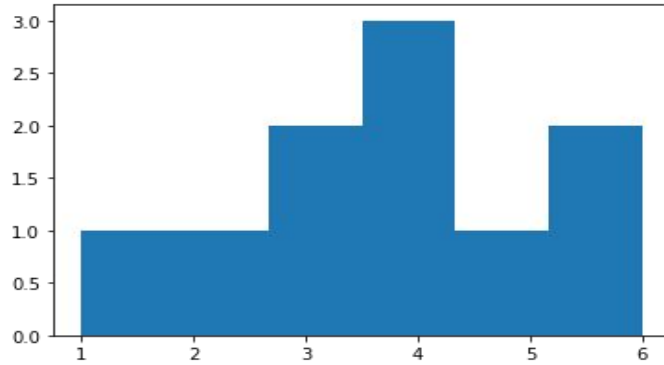
VISUALIZAÇÃO

```
x = []  
for c in range(10000):  
    x.append(np.random.randint(1,7))
```

```
x == [5, 3, 5, 2, 5, ... ]
```



Simulações em Python



Pandas

IMPORTANDO

```
import pandas as pd
```

SERIES

- Uma coluna

```
pd.Series()
```

```
pd.Series(x)
```

EXEMPLO:

```
import pandas as pd  
x = [1, 2, 3]  
print(pd.Series(x))
```

```
0  1  
1  2  
2  3
```

MUDANDO INDEX:

```
import pandas as pd  
  
x = [1, 2, 3]  
print(pd.Series(x, index = ['a',  
                             'b', 'c']))
```

```
a  1  
b  2  
c  3
```

Pandas

DATAFRAME

- Dados em tabelas

```
pd.DataFrame()
```

DICIONÁRIOS E DATAFRAMES

- chaves são os rótulos das colunas
- valores são as colunas

INDEX

```
dicionario.index = ['a1', 'a2', 'a3']
```

EXEMPLO:

```
import pandas as pd

dicionario = {"alunos": [50, 85, 73], "faltas": [3, 7, 8]}

dados = pd.DataFrame(dicionario)
dados.index = ['a1', 'a2', 'a3']

print(dados)
```

	alunos	faltas
a1	50	3
a2	85	7
a3	73	8

Pandas

ARQUIVOS CSV

```
dados = pd.read_csv('arquivo.csv')
```

- colocar caminho até arquivo dentro da função
- caso a primeira coluna seja o título de linha, adicionar dentro da função:

```
, index_col = 0)
```

```
dados = pd.read_csv('arquivo.csv', index_col = 0)
```

LOC:

- Seleciona linha dos dados

```
dicionario.loc[['a2']]
```

- Seleciona linha e coluna

```
dicionario.loc[['a2']][['faltas']]
```

ILOC:

```
dicionario.iloc[[1]]
```

	alunos	faltas
a1	50	3
a2	85	7
a3	73	8