

NumPy

Introdução

Parte 3

Números complexos em NumPy são representados por `j`.

```
In [52]: arr = np.array([1.j + 3, 4.j + 2])
```

```
In [53]: arr
```

```
Out[53]: array([3.+1.j, 2.+4.j])
```

```
In [54]: arr.real
```

```
Out[54]: array([3., 2.])
```

```
In [55]: arr.imag
```

```
Out[55]: array([1., 4.])
```

```
In [56]: arr.dtype
```

```
Out[56]: dtype('complex128')
```

`real`: Retorna a parte real do array ou o próprio array se somente possuir números reais.

`imag`: Contém a parte imaginária do array.

Iterando nos itens do array com flat

- O atributo flat retorna um objeto numpy.flatiter (esta é a única maneira de obter um flatiter porque não temos acesso a um construtor flatiter). O flat iterator permite percorrer um array como se fosse um array “plano”. Podemos acessar um ou mais elementos diretamente passando o índice. Podemos também definir o valor de todos os elementos ou específicos.

```
In [62]: arr = np.array([[ 0, 1, 2, 3], [4, 5, 6, 7], [8, 9, 10, 11]])
```

```
In [63]: itens = arr.flat
```

```
In [64]: for item in itens: print(item)
```

```
0  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11
```

```
In [68]: arr = np.array([[ 1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]])
```

```
In [69]: arr.flat[6]
```

```
Out[69]: 7
```

```
In [70]: arr.flat[[0,3]]
```

```
Out[70]: array([1, 4])
```

```
In [71]: arr.flat = 8
```

```
In [72]: arr
```

```
Out[72]:  
array([[8, 8, 8, 8],  
       [8, 8, 8, 8],  
       [8, 8, 8, 8]])
```

```
In [73]: arr.flat[[0, 1, 7]] = 3
```

```
In [74]: arr
```

```
Out[74]:  
array([[3, 3, 8, 8],  
       [8, 8, 8, 3],  
       [8, 8, 8, 8]])
```

Convertendo array em lista

Podemos usar o método `tolist()` para converter um array em lista.

```
In [81]: arr = np.array([[ 1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]])

In [82]: lista = arr.tolist()

In [83]: lista
Out[83]: [[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]]

In [84]: arr
Out[84]:
array([[ 1,  2,  3,  4],
       [ 5,  6,  7,  8],
       [ 9, 10, 11, 12]])

In [85]:
```

FIM