1. Slice no Numpy

O Numpy oferece algumas síntaxes para realizar *slices* em um *array* Numpy.

- *[:j]*: retorna todos os valores anteriores ao *j*-ésimo elemento.
- *[i:]*: retorna todos os valores a partir do *i*-ésimo elemento.
- *[i:j]*: retorna todos os valores a partir do *i*-ésimo elemento até o *(j-1)*-ésimo elemento.
- *[::k]*: retorna todos os valores a cada *k* passos.
- *[:j:k]*: retorna todos os valores anteriores ao *j*-ésimo elemento a cada *k* passos.
- *[i::k]*: retorna todos os valores a partir do *i*-ésimo elemento a cada *k* passos.
- *[i:j:k]*: retorna todos os valores a partir do *i*-ésimo elemento até o *(j - 1)*-ésimo elemento a cada *k* passos.

Note que geral, a síntaxe para a realização de *slices* pode ser resumida em *i:j:k* considerando que se pode omitir *i*, *j*, ou *k*. Abaixo é mostrado as síntaxes apresentadas acima em exemplos.

In [1]:
```
import numpy as np
```

In [2]:
```
x = np.arange(15)
x
```

Out[2]:
```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14])
```

In [5]:
```
a = x[:j]
a
```

Out[5]:
```
array([0, 1, 2, 3, 4, 5, 6, 7])
```

In [8]:
```
i, j, k = 4, 8, 2

print(f"[:j]: {x[:j]}")
print(f"[i:]: {x[i:]}")
print(f"[i:j]: {x[i:j]}")
print(f"[::k]: {x[::k]}")
print(f"[:j:k]: {x[:j:k]}")
print(f"[i::k]: {x[i::k]}")
print(f"[i:j:k]: {x[i:j:k]}")
```

```
[:j]: [0 1 2 3 4 5 6 7]
[i:]: [ 4  5  6  7  8  9 10 11 12 13 14]
[i:j]: [4 5 6 7]
[::k]: [ 0  2  4  6  8 10 12 14]
[:j:k]: [0 2 4 6]
[i::k]: [ 4  6  8 10 12 14]
[i:j:k]: [4 6]
```

2. Link para Revisar

Link: https://numpy.org/doc/stable/user/absolute_beginners.html (olhar a seção ***Indexing and Slicing***).

No link acima é mostrado como usar operações *booleanas* para realizar *slices* em *arrays*.