

Numpy

Advanced Institute for Artificial Intelligence

<https://advancedinstitute.ai>

- ☐ Introdução
- ☐ Estruturas
- ☐ Atributos
- ☐ Operações

- ☐ Biblioteca de computação científica
- ☐ Especializada para trabalhar com matrizes
- ☐ Funções de Álgebra Linear

Numpy manipula classes do tipo: ndarray

- ☐ Representa a estrutura fornecida pelo Numpy
- ☐ Estrutura parecida com as Listas
- ☐ Classe possui diversos atributos e funções

Alguns atributos:

- ❑ `ndarray.ndim` - Numero de dimensões do array
- ❑ `ndarray.shape` - Tupla com o tamanho dos array que representa cada dimensão
- ❑ `ndarray.size` - Quantidade de elementos no array
- ❑ `ndarray.dtype` - Tipo de dados que estão armazenados no array

Funções para criação de matrizes:

- ☐ `numpy.arange` - Cria um array com valores sequenciais (similar a função `range`)
- ☐ `numpy.array` - Convert outro objeto (`list`, `tupla`, etc) para `ndarray`
- ☐ `numpy.identity` - cria matriz identidade de tamanho `n`
- ☐ `numpy.ones` - cria matriz de tamanho `n` apenas com valores `1`
- ☐ `numpy.zeros` - cria matriz de tamanho `n` apenas com valores `0`

Algumas operações:

- ❑ `numpy.sum` - Soma os valores de um array para uma determinada dimensão
- ❑ `numpy.prod` - Multiplica os valores de um array para uma determinada dimensão
- ❑ `numpy.cumsum` - Soma cumulativa dos valores de um array para uma determinada dimensão
- ❑ `numpy.cumprod` - Multiplicação cumulativa dos valores de um array para uma determinada dimensão

Algumas operações:

- `numpy.transpose` - retorna a matriz transposta
- `numpy.dot` - multiplica duas matrizes
- `numpy.reshape` - alterar o formato da matriz, dentro de um mesmo conjunto de elementos
 - ex: alterar o formato de 3x3 para 1x9, 4x4 para 2x8

axis são definidos para matrizes com mais de uma dimensão.

- Uma matriz bidimensional possui dois eixos correspondentes:
 - O primeiro correndo verticalmente para baixo nas linhas (eixo 0)
 - O segundo correndo horizontalmente nas colunas (eixo 1)

Exemplo:

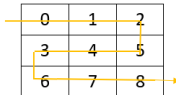
- somar cada linha de uma matriz
 - usamos a operação `sum` com opcional `axis=0`
- somar cada coluna de uma matriz
 - usamos a operação `sum` com opcional `axis=1`

Matriz original

0	1	2
3	4	5
6	7	8

Sum(axis=None)

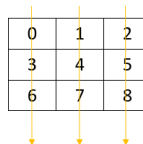
0	1	2
3	4	5
6	7	8



Retorno: 36

Sum(axis=0)

0	1	2
3	4	5
6	7	8

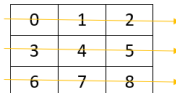


Retorno:

9	12	15
---	----	----

Sum(axis=1)

0	1	2
3	4	5
6	7	8



Retorno:

13
12
21