NumPy

Command	Structure	Example	
Import	**import** numpy **as** np		
Array	[array] = np.**array(** [lista] **)**	vetor = np.**array(** [1, 2, 3, 4, 5, 6, 7, 8] **)**	
		matriz = np.**array(** [[1, 2, 3] , [4, 5, 6]] **)**	
Matrix	[matriz] = np.**matrix(** [lista] **)**		
Access Item	[array] **[** [index] **]**	vetor**[**2**]**	
	[array] **[** [start] **:** [stop] **:** [step] **]**	vetor**[**1**:**9**:**3**]**	vetor = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] retorna [2, 5, 8]
	[matriz] **[** [linha] , [coluna] **]**	matriz **[**1, 2**]**	
Replace Item	[array] **[** [index] **]** = "[item]"	vetor**[**2**]** = 44	
	[matriz] **[** [linha] , [coluna] **]** = "[item]"	matriz**[**1,2**]** = 66	
Arrange	[array] = np.**arange(** [tamanho] **)**	vetor2 = np.**arange(** 3 **)**	vetor2 = [0, 1, 2]
	[array] = np.**arange(** [start] , [stop] , [step] **)**	vetor2 = np.**arange(** 0, 4.5, 0.5 **)**	vetor2 = [0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5]
Zeros	[array] = np.**zeros(** [tamanho] **)**	vetor3 = np.**zeros(** 8 **)**	Cria vetor de zeros com 8 posições
	[array] = np.**zeros(** ([linhas] , [colunas]) **)**	matriz3 = np.**zeros(** (2, 3) **)**	Cria matriz de zeros com 2 linhas e 3 colunas
Ones	[array] = np.**ones(** [tamanho] **)**	vetor3 = np.**ones(** 8 **)**	
	[array] = np.**ones(** ([linhas] , [colunas]) **)**	matriz3 = np.**ones(** (2, 3) **)**	
Eye	[matriz] = np.**eye(** [tamanho] **)**	matriz2 = np.**eye(** 3 **)**	Cria **matriz quadrada** com 1 na diagonal principal
Diag	[matriz] = np.**diag(** [lista] **)**	matriz3 = np.**diag(** [1, 2, 3, 4] **)**	Cria **matriz quadrada** com a diagonal principal definida
Linspace	[array] = np.**linspace(** [start] , [stop] , [num_elementos] **)**	vetor4 = np.**linspace(** 0, 10, 15 **)**	Cria vetor com 15 valores igualmente distribuídos entre 0 e 10

Methods

Command	Structure		Obs
Random.Rand	[variable] = np.**random.rand(** [tamanho] **)**	vetor = np.**random.rand**(5)	Retorna vetor com valores aleatório entre 0 e 1
	[variable] = np.**random.rand(** ([linhas] , [colunas]) **)**	vetor = np.**random.rand**(5,4)	
Random.Randn	[variable] = np.**random.randn(** [tamanho] **)**	vetor = np.**random.randn**(5)	Retorna vetor com valores aleatórios normalizados
	[variable] = np.**random.randn(** ([linhas] , [colunas]) **)**	vetor = np.**random.randn**(5,4)	
Mean	[variable] = np.**mean(** [array] **)**	media = np.**mean**(vetor)	Calcula a **media**
	[variable] = [array].**mean()**	media = vetor.**mean()**	
	[variable] = [array].**mean(** [eixo] **)**	media = vetor.**mean**(0)	0 : media de cada coluna 1: media de cada linha
Median	[variable] = np.**median(** [array] **)**	mediana = np.**median**(vetor)	Calcula a **mediana**
Quantile	[variable] = np.**quantile(** [array] , [quantis] **)**	quantis = np.**quantile**(vetor , [0, 0.25, 0.5, 0.5, 1])	Calcula os **quantis**
Std	[variable] = np.**std(** [array] **)**	desvio = np.**std**(vetor)	Calcula **desvio padrao**
	[variable] = [array].**std()**	desvio = vetor.**std()**	
Var	[variable] = np.**var(** [array] **)**	variancia = np.**var**(vetor)	Calcula a **variancia**
	[variable] = [array].**var()**	variancia = vetor.**var()**	
Sum	[variable] = np.**sum(** [array] **)**	soma = np.**sum**(vetor)	Soma os valores do vetor
	[variable] = [array].**sum()**	soma = vetor.**sum()**	
Cumsum	[variable].**cusum()**	vetor.**cusum()**	vetor = [1, 2, 3, 4, 5, 6, 7, 8] retorna [1, 3, 6, 10, 15, 21, 28, 36]
Min	[variable] = [array].**min()**	minimo = vetor.**min()**	
Max	[variable] = [array].**max()**	maximo = vetor.**max()**	
array_equal	np.**array_equal(** [array1] , [array2] **)**	np.**array_equal**(vetor1, vetor2)	Compara se 2 vetores sao iguais
Around	np.**around(** [array] , [num_decimais] **)**	np.**around**(vetor, 3)	Arredonda valores para 3 casas decimais
Repeat	np.**repeat(** [lista] , [num_vezes] **)**	np.**repeat(** [1,2,3,4], 3 **)**	Retorna [1,1,1, 2,2,2, 3,3,3, 4,4,4]
Tile	np.**tile(** [lista] , [num_vezes] **)**	np.**tile(** [1,2,3,4], 3 **)**	Retorna [1,2,3,4, 1,2,3,4, 1,2,3,4]
Concatenate	np.**concatenate(** ([array1] , [array2] **), axis=** [num] **)**	np.**concatenate(** (vetor1, vetor2**), axis=**0**)**	0 : add linha pra cada coluna 1: add coluna pra cada linha

Atributos

Command	Structure	Example	Obs
Shape	[variable].**shape**	**print(** vetor.**shape** **)**	retorna as dimensoes do array
dtype	[variable].**dtype**	**print(** vetor.**dtype** **)**	retorna o tipo de dado que compoem o array