



**NumPy** est une bibliothèque scientifique incluant des tableaux et des fonctions mathématiques opérant sur ces tableaux.



La bibliothèque **NumPy** est intégrée à EduPython.

## Import de la bibliothèque

- `import numpy` ou `import numpy as np`  
Allège le code... Par exemple `numpy.sqrt(2)` devient `np.sqrt(2)`



Dans tout ce document la bibliothèque **numpy** est renommée **np**.

Instruction : `import numpy as np`.

## Fonctions usuelles

Fonctions		Commande	Bibliothèques	syntaxe
Racine carrée	$\sqrt{x}$	<b>sqrt</b>	numpy	<code>np.sqrt(x)</code>
Puissance	$x^n$	<b>**</b>		<code>x**n</code>
Sinus	$\sin(x)$	<b>sin</b>	numpy	<code>np.sin(x)</code>
Cosinus	$\cos(x)$	<b>cos</b>	numpy	<code>np.cos(x)</code>
Tangente	$\tan(x)$	<b>tan</b>	numpy	<code>np.tan(x)</code>
Exponentielle	$e^x$	<b>exp</b>	numpy	<code>np.exp(x)</code>
Logarithme népérien	$\ln(x)$	<b>log</b>	numpy	<code>np.log(x)</code>
Valeur absolue	$ x $	<b>abs</b>		<code>abs(x)</code>

## Constantes utiles

constantes		Bibliothèques	syntaxe
<b>pi</b>	$\pi$	numpy	<code>np.pi</code>
<b>e</b>	$e$	numpy	<code>np.exp(1)</code>
<b>i (complexe)</b>	$i$		<code>1j</code>

## Fonctions sur les complexes

Fonctions		Commande	Bibliothèques	syntaxe
Partie réelle	$Re(z)$	<b>real</b>	numpy	<code>np.real(z)</code>
Partie imaginaire	$Im(z)$	<b>imag</b>	numpy	<code>np.imag(z)</code>
Module	$ z $	<b>abs</b>	numpy	<code>np.abs(z)</code>
Argument (en radian)	$Arg(z)$	<b>angle</b>	numpy	<code>np.angle(z)</code>
Conjugué	$\bar{z}$	<b>conj</b>	numpy	<code>np.conj(z)</code>

## Générateur de nombres aléatoires

Fonctions	Comman de	Bibliothèques	Syntaxe / exemples
Générateur aléatoire sur [0; 1]	rand	numpy.random	<ul style="list-style-type: none"> <li>numpy.random.rand()</li> <li>numpy.random.rand(n) (tableau de nombres aléatoires)</li> </ul> <pre>&gt;&gt;&gt; numpy.random.rand(4) array([0.43343257, 0.67802613, 0.96654483, 0.93035919])</pre>
Générateur aléatoire d'entiers entre deux bornes	randint	numpy.random	<ul style="list-style-type: none"> <li>numpy.random.randint(a,b) (entier <math>x</math> tel que <math>a \leq x &lt; b</math>)</li> <li>numpy.random.randint(a,b,n) (tableau de <math>n</math> entiers)</li> </ul> <pre>&gt;&gt;&gt; numpy.random.randint(1,7,5) array([5, 1, 1, 6, 4])</pre>

Il existe d'autres fonctions de génération de nombres aléatoires suivant des distribution précises (binomiales, exponentielles ou normales...)

## Tableaux NumPy

Création de tableaux NumPy	Commande	Bibliothèques	Syntaxe / exemples
A partir d'une liste de nombres	array	Numpy	<pre>np.array([1,2,3,4]) np.array([[1,0],[0,1]])</pre>
Générateur entre deux bornes	arange	Numpy	<pre>np.arange(a,b,pas)</pre> <p>(a : borne inférieure ; b borne supérieure exclue, pas facultatif)</p> <pre>&gt;&gt;&gt; np.random.randint(1,10,2) array([1, 3, 5, 7, 9])</pre>
Générateur de nombres avec écart fixe	linspace	Numpy	<pre>np.linspace(a,b,n)</pre> <p>(n nombres réels à écart constant entre a et b)</p> <pre>&gt;&gt;&gt; np.linspace(1,2,5) array([1. , 1.25, 1.5 , 1.75, 2. ])</pre>
Générateur de tableaux de zéros	zeros	Numpy	<ul style="list-style-type: none"> <li>np.zeros(n) (n zéros « au format réel »)</li> <li>np.zeros((n,p)) (tableau de zéros de dimension 3x2)</li> <li>np.zeros(n,dtype='int') (n zéros « au format entier »)</li> </ul> <pre>&gt;&gt;&gt; np.zeros(6) array([0., 0., 0., 0., 0., 0.]) &gt;&gt;&gt; np.zeros(6,dtype='int') array([0, 0, 0, 0, 0, 0]) &gt;&gt;&gt; np.zeros((3,2), dtype='int') array([[0, 0], [0, 0], [0, 0]])</pre>