

PYTHON diapo 18

Complexes

Roblet²

dernière MAJ le 08/01/20



CE QU'IL FAUT SAVOIR

Observons :

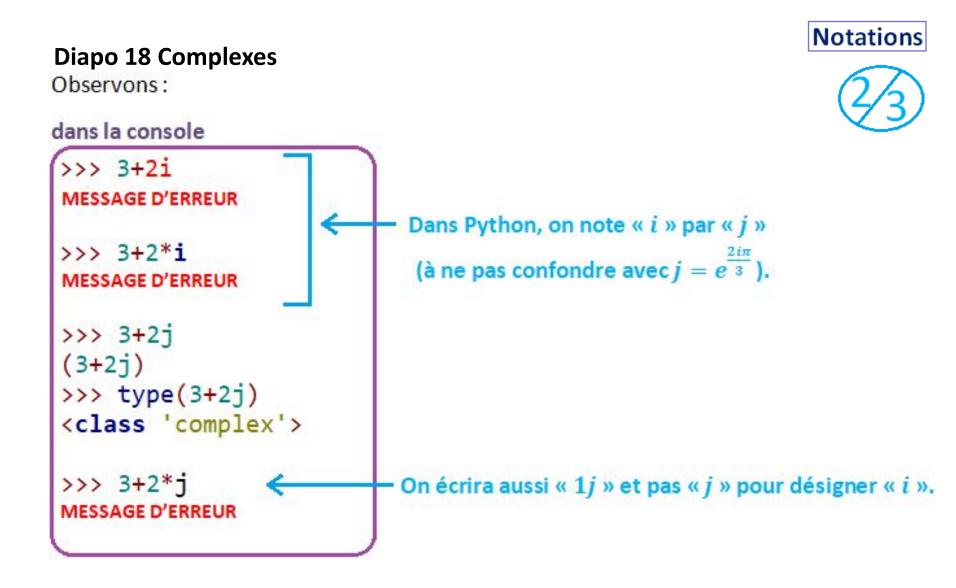
dans la console

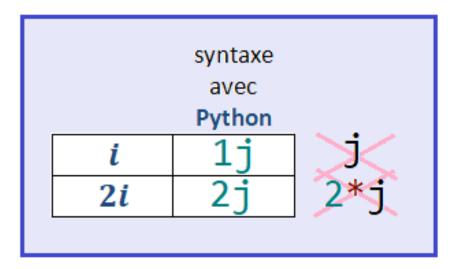
>>> 3+2i MESSAGE D'ERREUR
>>> 3+2*i MESSAGE D'ERREUR
<pre>>>> 3+2j (3+2j) >>> type(3+2j) <class 'complex'=""></class></pre>

>>> 3+2*j MESSAGE D'ERREUR









Remarque 1

Voici une autre façon d'écrire

un nombre complexe :

>>> complex(3,2) (3+2j)

Remarque 2

Si on utilise la bibliothèque Sympy,

on peut <u>aussi</u> noter *i* par I, et dans ce cas,

il faut mettre le signe * : syntaxe

particulière avec

	Sympy
i	Т
21	 2*T
20	∠ ⊥





CE QU'IL FAUT SAVOIR

Depuis la bibliothèque numpy	>>> from numpy import *
conjugué	<pre>>>> conj(3+2j) (3-2j)</pre>
partie réelle	<pre>>>> real(3+2j) 3.0</pre>
partie imaginaire	<pre>>>> imag(3+2j) 2.0</pre>
argument <mark>(en radians)</mark>	<pre>>>> angle(3+2j) 0.5880026035475675</pre>
module	<pre>>>> abs(3+2j) 3.6055512754639896</pre>
Depuis la bibliothèque cmath	>>> from cmath import *
argument (en radians)	<pre>>>> phase(3+2j) 0.5880026035475675</pre>
coordonnées polaires (module, argument)	<pre>>>> polar(3+2j) (3.6055512754639896, 0.5880026035475675)</pre>
forme algébrique	<pre>>>> rect(3,pi/4) (2.121320343559643+2.1213203435596424j)</pre>





Remarque

On aura remarqué que les valeurs retournées sont des valeurs approchées.

Par exemple, on sait qu'un nombre de coordonnées polaires $(2; \pi)$ a pour forme algébrique -2.

Or, voici ce que renvoie Python :

```
>>> rect(2,pi)
(-2+2.4492935982947064e-16j)
erreur de l'ordre de 10<sup>-16</sup>
```