



#### Unité 7 : Utiliser la bibliothèque cmath

#### Compétence 1 : Les fonctions de la bibliothèque


#### cmath

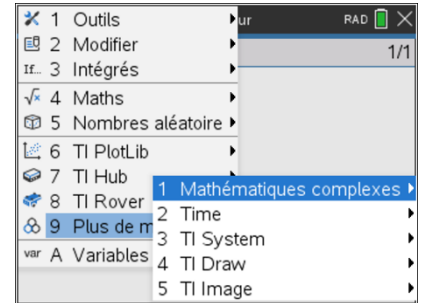
Dans cette première leçon de l'unité 7, vous allez découvrir comment utiliser la bibliothèque **cmath** pour effectuer des calculs simples sur les nombres complexes.

#### Objectifs :

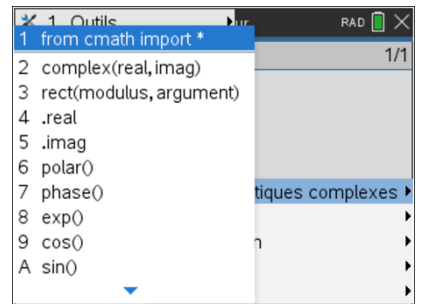
- Découvrir la bibliothèque **cmath**.
- Utiliser les fonctions de la bibliothèque **cmath**.

#### 1. Utiliser le module **cmath**.

- Insérer une nouvelle application et choisir le menu **A Ajouter Python**.
- Dans cette leçon, nous allons essentiellement travailler en utilisant la console (shell), afin de découvrir les instructions de la bibliothèque **cmath**.
- Dans la fenêtre qui s'ouvre, choisir l'option **3 Shell**.
- La touche  donne accès à **9 Plus de modules**, puis **1 Mathématiques complexes**.



- Importer la bibliothèque **cmath**.
- Choisir l'option **2 complex(real,imag)** et affecter ce nombre à une variable **z**.

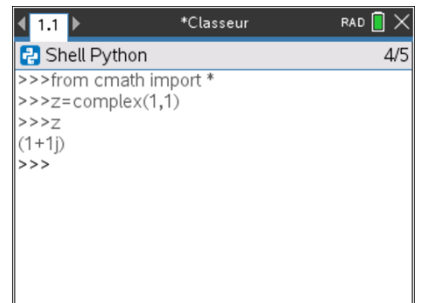


- Demander l'affichage de **z**.
- La calculatrice utilise **j** pour désigner le nombre imaginaire pur.

On remarquera que le nombre complexe est édité entre parenthèse sous la forme

$$z = a + bj$$

- De la même façon, définir un nombre complexe sans passer par l'instruction **complex(real, imag)**, mais directement en utilisant la même syntaxe, soit par exemple  $z2 = (2 - 3j)$



**Conseil à l'enseignant** : l'oubli de parenthèses lors de l'écriture d'un nombre complexe dans la console entraîne l'édition d'un message d'erreur.





# 10 Minutes de Code

TI - NSPIRE™ CX II & TI - PYTHON

## UNITE 7 : COMPETENCE 1

### NOTES DU PROFESSEUR

- Importer également dans la console, la bibliothèque de calculs mathématiques.
- Dans la bibliothèque **cmath**, choisir l'instruction **3 rect(module, argument)**.
- Compléter l'instruction `rect(sqrt(2),pi/4)`, soit la demande en coordonnées rectangulaires d'un nombre complexe de module  $\sqrt{2}$  et d'argument  $\frac{\pi}{4}$ .

```

1.1 *Classeur RAD 4/4
Shell Python
>>>from math import *
>>>rect(sqrt(2),pi/4)
(1+1j)
>>>|

```

- L'affichage des parties réelles et imaginaires d'un nombre complexe s'effectue par l'intermédiaire des méthodes **.real** et **.imag** précédées du nom de la variable complexe.

```

1.1 *Classeur RAD 8/8
Shell Python
>>>from math import *
>>>rect(sqrt(2),pi/4)
(1+1j)
>>>z2.real
2.0
>>>z2.imag
-3.0
>>>|

```

- L'instruction **6 polar()** ne doit comporter comme seul argument que le nom de la variable complexe afin de renvoyer un tuple dont le premier élément sera le module du nombre complexe et le second un argument.

```

1.1 *Classeur RAD 10/10
Shell Python
>>>type(z)
<class 'complex'>
>>>polar(z)
(1.414213562373095, 0.7853981633974483)
>>>u=polar(z)
>>>u[0]
1.414213562373095
>>>u[1]
0.7853981633974483
>>>|

```

- L'instruction **7 phase()** donne l'argument du nombre complexe exprimé en radians.

```

1.1 *Classeur RAD 3/3
Shell Python
>>>phase(z)
0.7853981633974483
>>>|

```





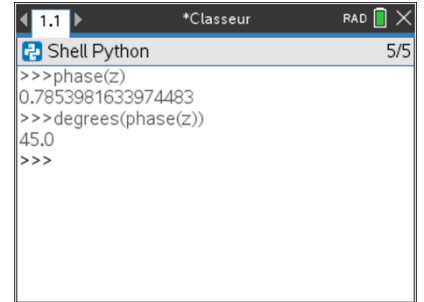
## 10 Minutes de Code

TI - NSPIRE™ CX II & TI - PYTHON

## UNITE 7 : COMPETENCE 1

### NOTES DU PROFESSEUR

- Pour exprimer un argument d'un nombre complexe en degrés, utiliser l'instruction **degrees** issue de la bibliothèque de fonctions mathématiques.

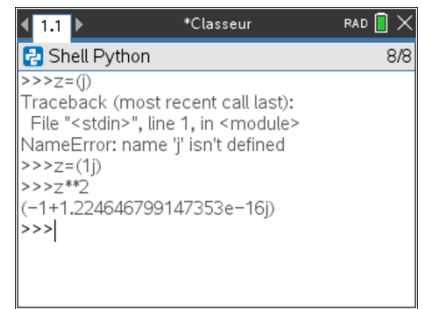


```
1.1 *Classeur RAD 5/5
Shell Python
>>>phase(z)
0.7853981633974483
>>>degrees(phase(z))
45.0
>>>
```

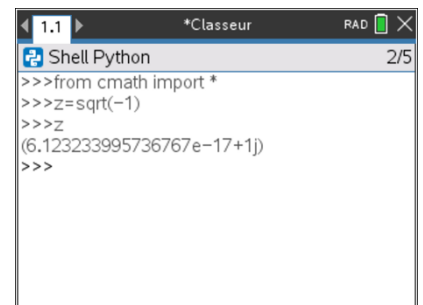
**Conseil à l'enseignant :** Le module d'un nombre complexe peut également être obtenu en utilisant l'instruction `abs(z)`.

### Carré d'un imaginaire pur.

- Créer le nombre complexe  $z = j$  (attention à l'instruction à fournir à la calculatrice).
- Calculer  $z^2$
- On obtient bien le résultat attendu, avec la précaution d'usage de bien conserver à l'esprit la façon dont les nombres décimaux sont exprimés en langage python.
- On pourra écrire dans un script une fonction permettant d'affecter 0 à la partie réelle ou imaginaire d'un nombre complexe, lorsque sa valeur n'excède pas  $10^{-n}$ , la valeur de  $n$  étant à préciser.



```
1.1 *Classeur RAD 8/8
Shell Python
>>>z=(j)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'j' isn't defined
>>>z=(1j)
>>>z**2
(-1+1.224646799147353e-16j)
>>>|
```



```
1.1 *Classeur RAD 2/5
Shell Python
>>>from cmath import *
>>>z=sqrt(-1)
>>>z
(6.123233995736767e-17+1j)
>>>
```





2. Quelques calculs élémentaires.

- a) Définir le nombre complexe de partie réelle 3 et de partie imaginaire 2. Vérifier que le module de ce nombre est  $\sqrt{13}$ .

```

1.1 *Classeur RAD 4/4
Shell Python
>>>z=complex(3,2)
>>>abs(z)==sqrt(13)
True
>>>|

```

- b) Nombre conjugué d'un nombre complexe.

Le conjugué d'un nombre complexe n'est pas implémenté dans la calculatrice TI-Nspire™. On se propose donc d'écrire pour terminer cette leçon, une fonction qui renvoie le conjugué d'un nombre complexe donné.

- Insérer une nouvelle application Python permettant d'écrire un script et le nommer U7SB1.
- Importer les bibliothèques de calculs mathématiques et le module **cmath**.
- Extraire les parties réelle et imaginaire du complexe passé en argument de la fonction.
- Afficher les deux nombres  $z$  et  $\bar{z}$ .

```

1.1 1.2 1.3 *Classeur RAD 10/10
*U7SB1.py
# Math Calculations
#=====
from math import *
from cmath import *
#=====
def conj(z):
    a=z.real
    b=z.imag
    conjugue=complex(a,-b)
    return z,"a pour conjugue",conjugue

```

Exemple : Soit  $z = 2 + 3j$ , déterminer à l'aide de la fonction **conj(z)** le nombre complexe conjugué.

```

1.1 1.2 1.3 *Classeur RAD 4/4
Shell Python
>>>z=complex(2,3)
>>>conj(z)
((2+3j), 'a pour conjugue', (2-3j))
>>>|

```

