

**Unidade 7: Utilização da biblioteca Matemática Complexa**

**Lição 1: As funções da biblioteca cmath**

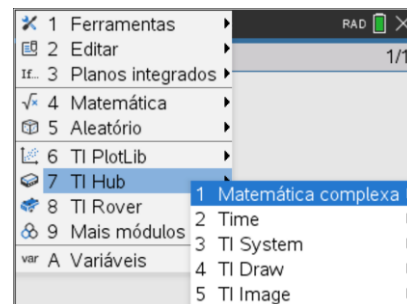
Nesta primeira lição da Unidade 7, pode aprender como utilizar a biblioteca **cmath (Matemática Complexa)** para efetuar cálculos simples com números complexos.

**Objetivos:**

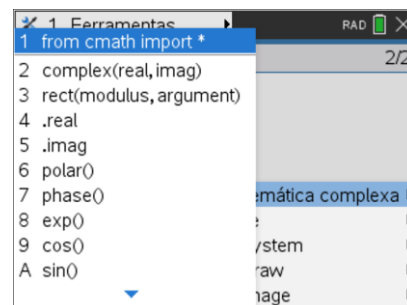
- Descobrir a biblioteca **cmath**.
- Utilizar as funcionalidades da biblioteca **cmath**.

**1. Utilizar o módulo cmath.**

- Inserir uma nova aplicação e escolher o menu **A: Adicione Python**.
- Nesta lição, vamos trabalhar essencialmente com o interpretador (Shell) para tratar as instruções da biblioteca **cmath**.
- Na janela que se abre, escolher a opção **3: Shell**.
- A tecla **menu** dá acesso a **9: Mais módulos**, e depois a aceder a **1: Matemática complexa**.



- Importar a biblioteca **cmath**.
- Escolher a opção **2: complex(real, imag)** e atribuir este número a uma variável **z**.

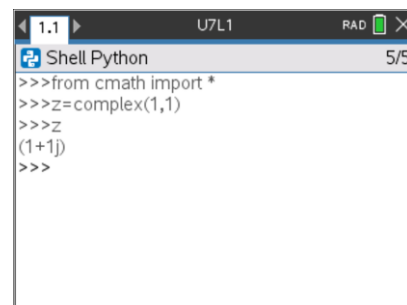


- Chamar a variável **z**.
- A calculadora utiliza **j** para designar a unidade imaginária.

Observe que o número complexo é apresentado entre parênteses na forma

$$z = a + bj$$

- Da mesma forma, definir um número complexo sem usar a instrução **complex(real, imag)**, mas diretamente utilizando a sintaxe, como por exemplo  $z2 = (2 - 3j)$



**SUGESTÃO:**

Não colocar parênteses na escrita de um número complexos na Shell pode levar a uma mensagem de erro.





- Importar também na Shell a biblioteca de cálculos matemáticos.
- Na biblioteca **cmath**, escolher a instrução **3: rect(modulus, argument)**.
- Completar com a instrução **rect(sqrt(2), pi/4)**, que dá origem à escrita em coordenadas cartesianas do número complexo de módulo  $\sqrt{2}$  e argumento  $\frac{\pi}{4}$ .
- A exibição da parte real e do coeficiente da parte imaginária de um número complexo efetuam-se com **.real** e **.imag** precedido do nome da variável complexa.
- A instrução **6: polar()** deve conter como único argumento o nome da variável complexa para retomar um par ordenado cujo primeiro elemento será o módulo e o segundo o argumento do número complexo.
- A instrução **7: phase()** dá o argumento, em radianos, do número complexo.

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

```
1.1 U7L1 RAD 10/10
Shell Python
>>>from cmath import *
>>>z=complex(1,1)
>>>z
(1+1j)
>>>z2=(2-3j)
>>>z2.real
2.0
>>>z2.imag
-3.0
>>>
```

```
1.1 U7L1 RAD 20/20
Shell Python
>>>type(z)
<class 'complex'>
>>>polar(z)
(1.414213562373095, 0.7853981633974483)
>>>u=polar(z)
>>>u[0]
1.414213562373095
>>>u[1]
0.7853981633974483
>>>
```

```
1.1 U7L1 RAD 3/3
Shell Python
>>>phase(z)
0.7853981633974483
>>>
```





- Para obter o argumento de um número complexo em graus, utilizar a função **degrees()**, que está na biblioteca de funções matemáticas.

```
1.1 U7L1 RAD 7/7
Shell Python
>>>z=(1+1j)
>>>phase(z)
0.7853981633974483
>>>from math import *
>>>degrees(phase(z))
45.0
>>>|
```

**SUGESTÃO:**

O módulo de um número complexo  $z$  pode também ser obtido com a instrução **abs(z)**.

**Quadrado de um imaginário puro.**

- Criar o número complexo  $z = j$  (preste atenção às instruções a dar à calculadora).
- Calcular  $z^2$
- Obtém-se o resultado esperado, tendo em consideração a forma como os decimais são expressos em linguagem Python.
- Pode escrever-se num programa uma função que atribua 0 à parte real ou imaginária de um número complexo quando o valor não ultrapassar  $10^{-n}$ , sendo  $n$  a precisão.

```
1.1 U7L1 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 U7L1 RAD 5/5
Shell Python
>>>from cmath import *
>>>z=sqrt(-1)
>>>z
(6.123233995736767e-17+1j)
>>>
```





**2. Alguns cálculos elementares**

**a) Condições com números complexos.**

Definir o número complexo com parte real 3 e coeficiente da parte imaginária 2. Verificar que o módulo deste número é  $\sqrt{13}$ .

```

1.1 U7L1 RAD 4/4
Shell Python
>>>z=complex(3,2)
>>>abs(z)==sqrt(13)
True
>>>
    
```

**b) Conjugado de um número complexo.**

O conjugado de um número complexo não é implementado na calculadora TI-Nspire™. Assim, propõe-se, para finalizar esta lição, a escrita de um programa que permita obter o conjugado de um número complexo dado.

- Inserir uma nova página com a aplicação TI-Python para escrita do programa e nomeie-se por U7L1.
- Importar as bibliotecas de cálculo matemático e o módulo **cmath**.
- Extrair as partes real e imaginária do complexo que é argumento da função definida.
- Exibir os dois números  $z$  e  $\bar{z}$ .

```

1.1 1.2 *U7L1 RAD 10/10
*U7L1.py
# Cálculos matemáticos
# =====
from math import *
from cmath import *
# =====
def conj(z):
    a=z.real
    b=z.imag
    conjugado=complex(a,-b)
    return z, "tem como conjugado",conjugado
    
```

**Exemplo:**

Sendo  $z = 2 + 3j$ , determinar o seu conjugado com a função **conj(z)**.

```

1.1 1.2 1.3 *U7L1 RAD 4/4
Shell Python
>>>z=complex(2,3)
>>>conj(z)
((2+3j), 'tem como conjugado', (2-3j))
>>>
    
```

