

**Unidade 9: Módulo Turtle em TI-Python**

**Lição 1: Iniciação com polígonos**

Nesta lição, vai instalar o módulo Turtle na calculadora e escrever os primeiros programas para desenhar alguns polígonos, utilizando instruções que permitam à tartaruga fazer movimentos.

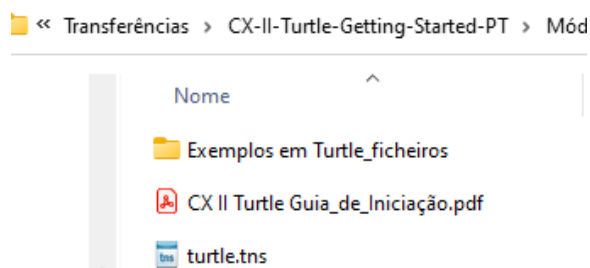
**Objetivos:**

- Instalar o módulo Python
- Desenhar e pintar polígonos

**1. Introdução 1**

Antes de começar, certifique-se de que a sua TI-Nspire CX II-T tem o Sistema Operativo atualizado.

De seguida, caso não tenha o módulo **Turtle** instalado, deve aceder à página web <https://education.ti.com/pt/product-resources/turtle-module/nspire-python> para obter uma pasta compactada, onde encontra o módulo e ainda um guia de iniciação.



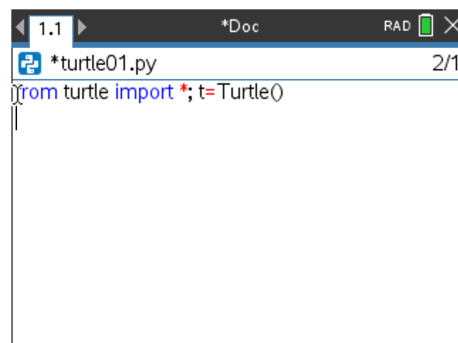
**2. Introdução 2**

Com o módulo **Turtle** instalado pode agora incluir os respetivos comandos em linhas de código num programa em Python.

Para que os comandos tenham efeito, deve começar por importar a biblioteca de Turtle para a primeira linha do programa.

Prima **[Menu] > Mais módulos > Turtle Graphics > from turtle import\***

Surge a instrução **from turtle import\***, mas também, separado por ponto e vírgula (;), a instrução **t=Turtle()** na linha de código, a qual cria o objeto Turtle com o nome t.



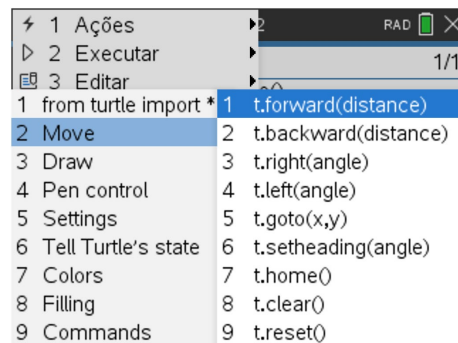
**NOTA:** A utilização do ponto e vírgula (;) é a forma como o Python permite que se escrevam duas instruções na mesma linha.

**3. Introdução 3**

Voltando ao **[Menu]**, a **Mais módulos** e depois novamente a **Turtle Graphics**, encontra oito submenus (2 ... 8): *Move*, *Draw*, *Pen control*, *Settings*, *Tell Turtle's state*, *Colors*, *Filling* e *Commands*.

Cada submenu contém funções específicas da tartaruga (e outras), as quais começam com a variável do objeto **t**, que é o nome padrão da tartaruga (turtle). O nome pode ser alterado e pode também ter-se várias tartarugas no mesmo programa, devendo o utilizador fazer as escolhas que entender.

As funcionalidades do submenu **Move** podem observar-se na figura. Poderá observar as restantes funcionalidades submenu a submenu.





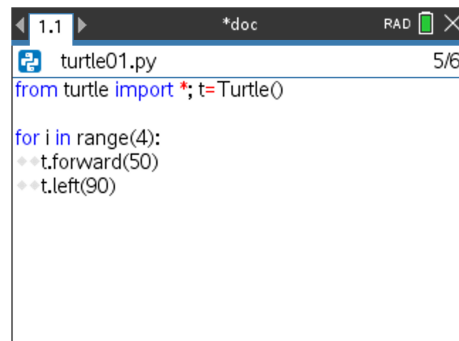
#### 4. Um quadrado – o código

Para fazer com que a tartaruga desenhe um quadrado, escreva um ciclo de repetição com apenas duas instruções:

```

for i in range(4):
    t.forward(50) # pixels
    t.left(90)   # graus

```

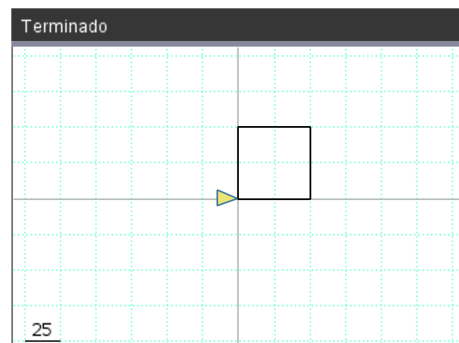


#### 5. Um quadrado – a execução do programa

Depois de introduzir o código, execute o programa (**CTRL + R**) para perceber se a tartaruga realmente faz um quadrado.

A tartaruga começa na origem, virada para a direita. O valor 50, em forward, refere-se ao número de pixels no ecrã. As linhas de grelha estão separadas por 25 pixels, como se pode observar na legenda no canto inferior esquerdo do ecrã. O valor 90, em left (rotação no sentido positivo), representa 90°.

No final, a tartaruga regressa à origem e fica voltada para a direita. O desenho (gráfico) permanece no ecrã até ser premida uma tecla.



#### 6. Um polígono regular com n lados – o desenho

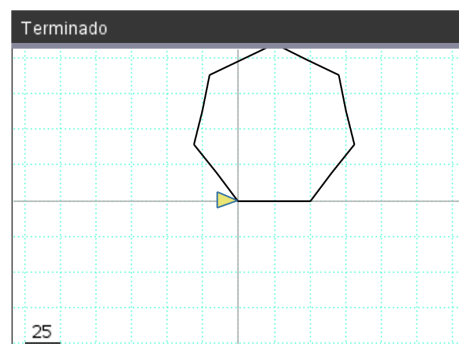
Modifique o programa que permitiu desenhar o quadrado para permitir que o utilizador introduza o número de lados para que a tartaruga desenhe um polígono regular com esse número de lados.

Utilize, antes do ciclo de repetição, a instrução:

```
n=int(input("Quantos lados? "))
```

e altere o código do ciclo de repetição.

**NOTA:** *Um pouco mais de conhecimento matemático será necessário para decidir qual o ângulo de rotação, que passará a ser a divisão de 360° por n. Para desenhar o heptágono da figura o ângulo é 360°/7.*



### 7. Um polígono regular com n lados – o código

Um programa para desenhar o heptágono observado anteriormente ou outro polígono regular à escolha, em função do número de lados, pode ser o que se observa na figura ao lado.

Coloque a instrução **t=Turtle()** após a linha de código para introdução do número de lados para que a pergunta seja feita antes de surgir o ecrã do desenho.

Para alterar a velocidade da tartaruga, utilize **t.speed(0)**, instrução que pode encontrar no submenu **Settings** (Menu > Mais módulos > Turtle Graphics > Settings). O argumento entre parênteses é um valor entre 0 a 10, sendo 0 o que corresponde ao movimento mais rápido.

**NOTA:** *Faça algumas experiências com outros valores de n, de velocidade e de medida do lado do polígono.*

```

1.1 1.2 *doc RAD X
*turtle01.py 3/11
from turtle import *
n=int(input("Quantos lados? "))
t=Turtle()
t.speed(0)
for i in range(n):
    t.forward(50)
    t.left(360/n)
    
```

### 8. Colorir o polígono – introdução

Há dois métodos para atribuir cor a uma figura:

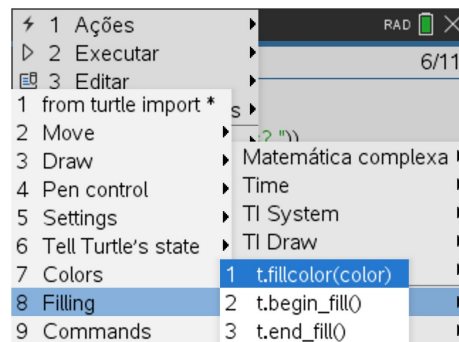
**t.pencolor(r,g,b)** - no submenu **Pen control**

**t.fillcolor(r,g,b)** - no submenu **Filling**

O terno **(r,g,b)** corresponde ao código RGB da cor pretendida.

**t.pencolor** permite colorir linhas, nesta caso os lados do polígono. Se a instrução estiver no ciclo de repetição e o código RGB for automaticamente alterado, pode ter um lado de cada cor.

**t.fillcolor** permite preencher a figura com a cor do código RGB considerado.

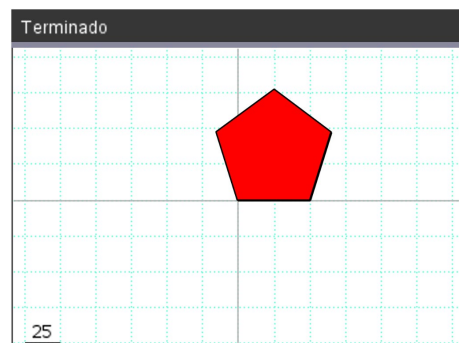


### 9. Um pentágono vermelho

Para obter um pentágono com o interior vermelho, antes de o desenhar tem de escolher a cor, com a instrução **t.fillcolor(255,0,0)** do submenu **Filling**.

Depois, tem de dar ordem para início do preenchimento com a instrução

**t.begin\_fill()**, do mesmo submenu.





Durante o ciclo de repetição, o sistema guarda a lista de vértices “visitados”.

No final, para que se observe o polígono com o interior preenchido tem de colocar a instrução `t.end_fill()`.

```

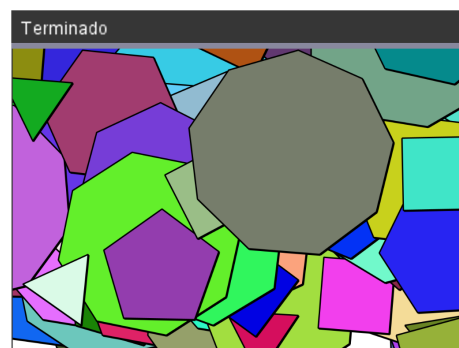
1.1 1.2 *doc RAD 10/10
turtle01.py
from turtle import *
n=int(input("Quantos lados? "))
t=Turtle()
t.speed(0)
t.fillcolor(255,0,0)
t.begin_fill()
for i in range(n):
    t.forward(50)
    t.left(360/n)
t.end_fill()

```

### 10. Arte com polígonos I

Modifique o programa para fazer polígonos de forma aleatória, seja em relação ao número de lados, à cor ou ao local de início do desenho.

- Utilize `t.hideturtle()` e `t.hidegrid()` para melhorar a aparência e a velocidade.
- Utilize `while get_key()!="esc"` no ciclo de repetição principal para parar ao premir a tecla “esc”.
- Utilize `t.penup()`, `t.goto(,)`, `t.setheading()` e `t.pendown()` para colocar a tartaruga num ponto de partida do ecrã aleatório, virada numa direção aleatória, para um comprimento aleatório do lado do polígono.



**NOTA:** O ecrã da tartaruga tem um número de pixels que varia de -159 a 159 na horizontal e de -106 a 106 na vertical e não há problema se sair fora do ecrã. Pode usar qualquer ângulo, em graus, para rotação.

### 11. Arte com polígono II

Estas linhas de código produzem algo ainda um pouco monótono, como pode observar na imagem.

Modifique alguma coisa para tornar o quadro multicolor!

```

1.2 2.1 2.2 *doc RAD 10/23
*turtle2.py
from turtle import *
from random import *
#n=int(input("Quantos lados? "))
t=Turtle()
t.hidegrid();t.hideturtle();t.speed(0)
while get_key()!="esc":
    n=randint(3,9)
    t.penup()
    t.goto(randint(-150,150),randint(-105,105))
    t.pendown()
    t.setheading(randint(-90,90))
    t.fillcolor(255,0,0)
    t.begin_fill()
    for i in range(n):
        t.forward(50)
        t.left(360/n)
    t.end_fill()

```

