

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

**Lição 2: Cor e movimento**

Nesta lição, vai continuar a descobrir o módulo Turtle, utilizando as opções desenho e as opções de movimento.

**Objetivos:**

- Desenhar um triângulo isósceles e colorir o interior
- Colorir um círculo
- Integrar estes dois objetos para fazer um desenho

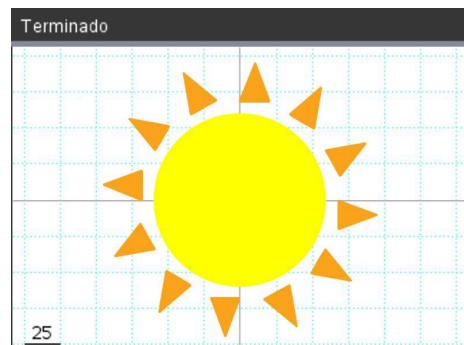
**0. Introdução**

Como desenhar a figura ao lado utilizando um programa em Python e o módulo Turtle?

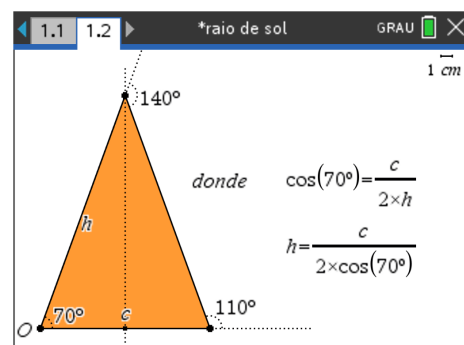
Parece bastante simples. Mas será realmente simples?

Observe que a figura é constituída por:

- um círculo amarelo centrado no ecrã
- um conjunto de triângulos isósceles iguais colocados à volta do círculo.



Para desenhar o sol importa perceber a construção de cada elemento em separado: o círculo e o triângulo.



**1. Desenhar um triângulo isósceles 1**

Vejam as etapas a seguir para construirmos um triângulo isósceles:

- Inicie um novo programa num editor de Python e nomeie-o “Triangulo”.
- Importe a biblioteca de Turtle e ainda a biblioteca de funções matemáticas (math) para os cálculos trigonométricos.
- Crie a variável c, correspondente à medida da base do triângulo com 60 pixéis (c=60).
- Crie a variável b, exprimindo em radianos um ângulo de 70°.
- Crie a variável h, medida de cada um dos lados iguais.

```
*Triangulo.py 5/5
from turtle import *; t=Turtle()
from math import *
c=60
b=pi*70/180
h=c/(2*cos(b))
```

## 2. Desenhar um triângulo isósceles 2

Mais algumas configurações para o nosso triângulo:

- Poderá fixar a velocidade da tartaruga com **t.speed()**, escolhendo como argumento um número natural ente 0 e10.
- No menu **8: Filling** pode escolher a cor no sistema RGB para colorir o interior do polígono.

**Nota:** Há, neste sistema, tantas cores quantos os códigos que se podem formar com um terço em que o primeiro elemento é a intensidade de vermelho, de 0 a 255, e cada um dos seguintes, também no mesmo intervalo, respetivamente verde e azul.

- Os comandos **t.begin\_fill()** e **t.end\_fill()** são utilizados para estabelecer os momentos inicial e final do preenchimento do polígono.

```

1.1 *Doc RAD
*Triangulo.py 3/11
from turtle import *; t=Turtle()
from math import *
t.hideturtle()
c=60
b=pi*70/180
h=c/(2*cos(b))
t.pencolor(251,163,26)
t.fillcolor(251,163,26)
t.begin_fill()

t.done()

```

## 3. Desenhar um triângulo isósceles 3

Continuando a completar o código para o nosso triângulo:

- No menu **2: Move**, use os comandos **t.forward()** e **t.left()** para completar a construção do triângulo.
- Por uma questão estética esconda a tartaruga com o comando **t.hideturtle()** localizado no menu **5: Settings**.
- Não se esqueça do comando **t.done()** no final para exibir o desenho no final da execução do programa.

```

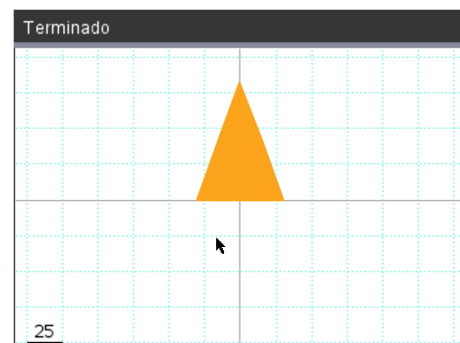
1.1 Doc RAD
Triangulo.py 13/18
t.fillcolor(251,163,26)
t.begin_fill()
t.forward(c/2)
t.left(110)
t.forward(h)
t.left(140)
t.forward(h)
t.left(110)
t.forward(c/2)
t.end_fill()
t.done()

```

**Nota:** O triângulo a desenhado podia ter sido simplificado. Foi construído desta forma para serem melhor compreendidos os cálculos, salientando os dois triângulos retângulos e a escolha do ponto de partida.

- Execute o programa e observe o triângulo obtido.
- Faça várias experiências, modificando a cor de fundo e do contorno, as medidas, ...

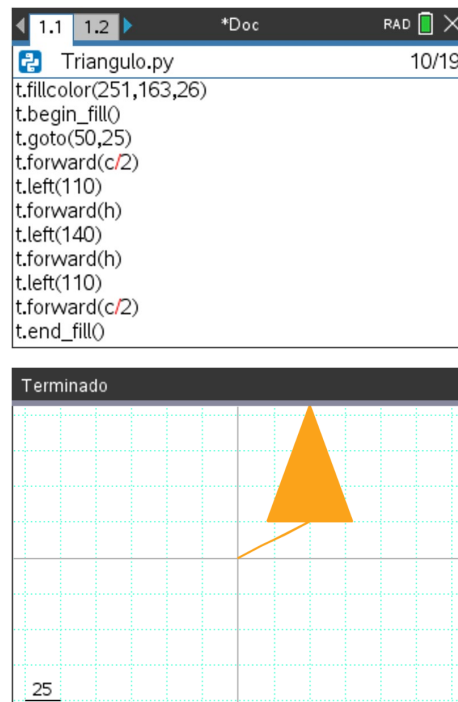
**Nota:** Para construir o sol com os raios deverá ser capaz de modificar a orientação do triângulo, assim como a sua posição no referencial.



#### 4. Mudar o sítio do triângulo

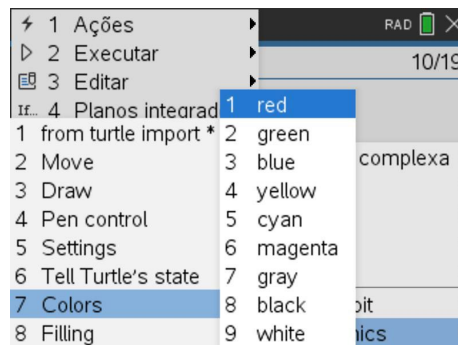
Vajamos, agora, como mover o triângulo para outra região do nosso referencial:

- No menu **2: Move**, escolha o comando **t.goto(x,y)**. Este comando permite indicar o ponto, através da suas coordenadas (x,y), para onde a tartaruga se deve mover.
- O local da linha de código com este comando é muito importante. Neste exemplo será antes do desenho do triângulo. Coloque **t.goto(50,25)**.
- Execute o programa. Foi realmente feita uma translação para o ponto (50,25), com a desvantagem de ter ficado o rasto dessa mudança, um segmento de reta do ponto (0,0) ao ponto (50,20).



#### 5. As cores no Turtle

**Nota:** As cores podem ser escolhidas, também, no menu **7: Colors**, para além do uso do código RGB.

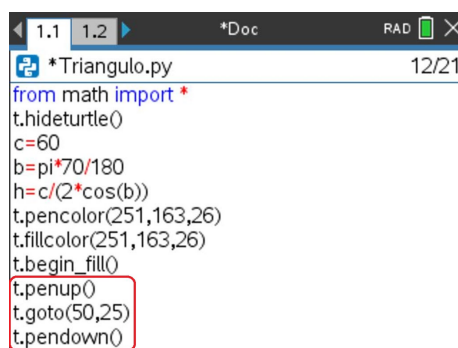


#### 6. Eliminar o rasto do deslocamento

Vejamos como eliminar o segmento que traduz a deslocação do triângulo:

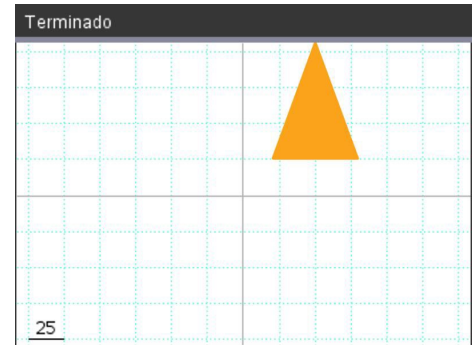
- Para eliminar o segmento de reta definido pelos pontos de coordenadas (0,0) e (50,25), vá ao menu **4: Pen control** e utilize os comandos **2: t.penup()** e **3: t.pendown()**.

**Nota:** Naturalmente, o comando para deslocar a tartaruga deverá ser incluindo entre estes.





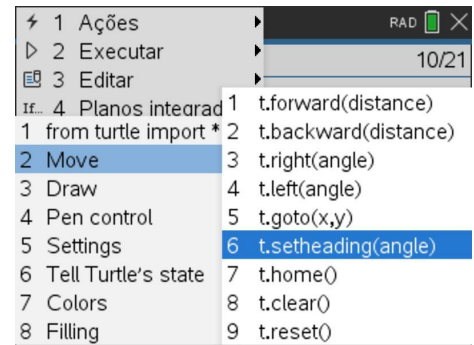
- Execute o programa e observe o efeito produzido por estas três linhas de código.



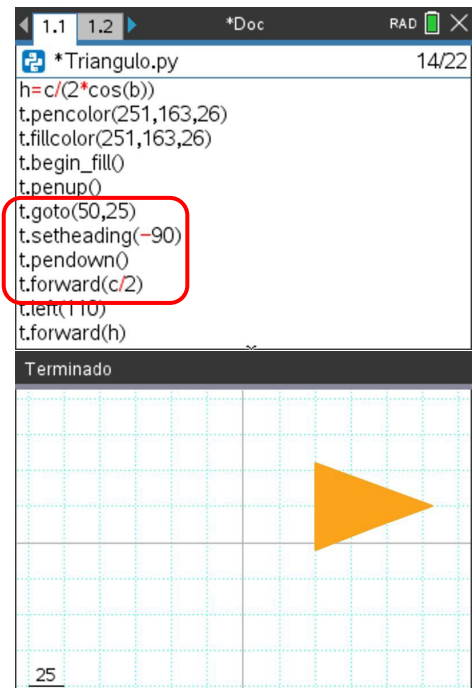
## 7. Mudar a orientação do triângulo

- No menu **1: Move**, selecione o comando **t.setheading()**, esta instrução permitirá rodar a seta (cabeça) do triângulo que representa a tartaruga.
- Coloque esta instrução antes dos comandos para desenhar o triângulo

**Nota:** *t.setheading(graus)* define a medida do ângulo de rotação da tartaruga, em graus. Se os valores da medida forem positivos, a rotação faz-se no sentido anti-horário e a rotação será no sentido horário para valores negativos.



- Em comparação com a orientação anterior, o triângulo deve rodar 90° no sentido horário, portanto terá que se inserir a instrução **t.setheading(-90)**.



- Execute de novo o programa e observe o resultado obtido
- Para desenhar o sol raiado, será assim possível criar a coroa de raios.

## 8. Construção do círculo que representa o Sol

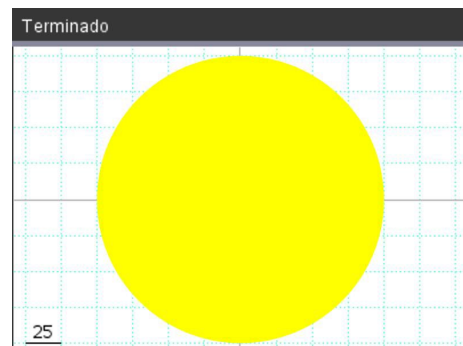
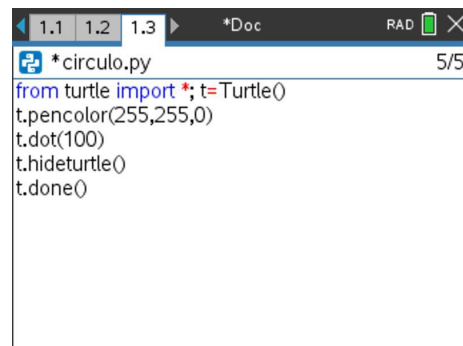
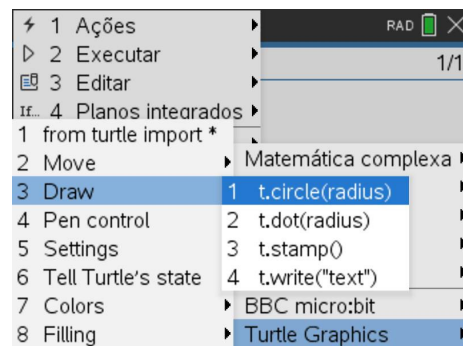
Os comandos do tipo **t.fillcolor(r,g,b)**, associados a **t.begin\_fill()** e **t.end\_fill()** apenas funcionam para polígonos e não para círculos.

A instrução **t.circle()** apenas desenha uma circunferência e não o seu interior.

Deverá então ser utilizado o comando **t.dot(diâmetro)** que se encontra no menu **3: Draw**. A ideia subjacente é o desenho de um “ponto” com um determinado diâmetro, definido em pixels.

O comando **t.pencolor(r,g,b)** permite a escolha da cor do lápis para desenhar esse ponto.

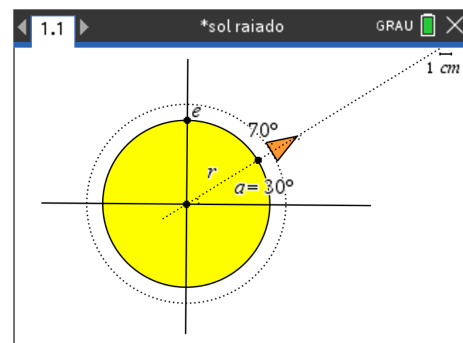
- Crie um novo programa e designe-o por círculo.
- Escreva os comandos que observa na figura para desenhar um círculo amarelo.
- Teste o programa, executando-o.



### 9. Código para o programa do Sol raiado

Vamos terminar este projeto usando os programas anteriores para construir o nosso sol raiado.

- Modifique o programa triangulo e crie agora um programa para desenhar o Sol com os raios à volta.
- A representação geométrica ao lado fornece os nomes das variáveis para utilizar no programa.
- Ao redor do círculo central do Sol temos, a cada 30°, um triângulo isósceles de base c e lado h (medida de cada um dos lados iguais). Estes triângulos são representados numa circunferência de raio r+e.





- Para orientar corretamente cada triângulo que representa cada raio de sol, utilize o comando `t.setheading(-90+30*i)`, em que `i` é número de incrementos de 30°.
- O programa completo obtém-se com todos os passos que foram observados até aqui nesta lição, e que se encontram na imagem ao lado. Abaixo coloca-se o código relativo à construção dos raios:

```
#raios
h=c/(2*cos(70*pi/180))
for g in range(0,360,30):
    t.penup()
    t.goto((r+e)*cos(g*pi/180),(r+e)*sin(g*pi/180))
    t.setheading(-90+p*i)
    t.pendown()
    t.pencolor(251,163,26)
    t.begin_fill()
    t.fillcolor(251,163,26)
    t.forward(c/2)
    t.pendown()
    t.forward(c/2)
    t.left(110)
    t.forward(h)
    t.left(140)
    t.forward(h)
    t.left(110)
    t.forward(c/2)
    i+=1
    t.end_fill()
```

```
*Triangulo.py 1/37
from turtle import *; t=Turtle()
from math import *
#sol
r=60
c=20
e=8
p=30
i=0
t.hideturtle()
#raios
h=c/(2*cos(70*pi/180))
for g in range(0,360,30):
    t.penup()
    t.goto((r+e)*cos(g*pi/180),(r+e)*sin(g*pi/180))
    t.setheading(-90+p*i)
    t.pendown()
    t.pencolor(251,163,26)
    t.begin_fill()
    t.fillcolor(251,163,26)
    t.forward(c/2)
    t.pendown()
    t.forward(c/2)
    t.left(110)
    t.forward(h)
    t.left(140)
    t.forward(h)
    t.left(110)
    t.forward(c/2)
    i+=1
    t.end_fill()
#sol raiado
t.home()
t.pencolor(255,255,0)
t.dot(r)
t.hideturtle()
t.done()
```

## 10. O Sol Raiado

Execute o programa, **CTRL + R**, e veja o produto final do seu projeto.

