

Swampy/Turtle

- <http://tinyurl.com/thinkpython2e>

Exemplo

```
import turtle
bob = turtle.Turtle()
print(bob)
turtle.mainloop()
```

Funções

- fd e bk: mover a tartaruga para a frente e para trás, respectivamente
- lt e rt: mover a tartaruga para a esquerda e para a direita, respectivamente.
- pu e pd: p(en) u(p) e d(own), ou seja, a tartaruga não deixa ou deixa ficar rasto, respectivamente

Exemplo

```
import turtle
bob = turtle.Turtle()
print(bob)
```

```
bob.fd(100)
bob.lt(90)
bob.fd(100)
```

Como é que poderíamos desenhar um quadrado?

Exemplo

```
bob.fd(100)
bob.lt(90)
bob.fd(100)
bob.lt(90)
bob.fd(100)
bob.lt(90)
bob.fd(100)
```

Exemplo

```
for i in range(4):  
    print('Hello!')
```

Exemplo

```
bob.fd(100)  
bob.lt(90)
```

Defina uma função `square` que recebe como parâmetro uma “tartaruga” e utiliza-a para desenhar um quadrado

Exemplo

```
def square(t):  
    for i in range(4):  
        t.fd(100)  
        t.lt(90)  
  
square(bob)
```


Adicione à função `square` um parâmetro `length` que indique o “tamanho” de cada lado do quadrado.

Exemplo

```
def square(t, length):  
    for i in range(4):  
        t.fd(length)  
        t.lt(90)  
  
square(bob, 100)
```

- A partir da função `square` defina uma outra denominada `polygon` que desenhe um polígono regular com n lados.

Dica

Os ângulos externos de um polígono regular de n -lados são de $360/n$ graus.

Exemplo

```
def polygon(t, n, length):  
    angle = 360 / n  
    for i in range(n):  
        t.fd(length)  
        t.lt(angle)  
  
polygon(bob, 7, 70)
```

Escreva uma função para desenhar um círculo que recebe como parâmetro o raio r

Exemplo

```
def circle(t, r):  
    circumference = 2 * math.pi * r  
    n = 50  
    length = circumference / n  
    polygon(t, n, length)  
  
circle(bob, 50)
```

Exemplo

```
def circle(t, r):  
    circumference = 2 * math.pi * r  
    n = int(circumference / 3) + 1  
    length = circumference / n  
    polygon(t, n, length)  
  
circle(bob, 50)
```

Escreva uma versão mais geral `circle` denominada por `arc` que recebe um parâmetro adicional `angle` que determina que fracção do círculo se deve desenhar. Quando `angle` for igual a 360, `arc` deve desenhar um círculo completo.

Exemplo

```
def arc(t, r, angle):  
    arc_length = 2 * math.pi * r * angle / 360  
    n = int(arc_length / 3) + 1  
    step_length = arc_length / n  
    step_angle = angle / n  
    for i in range(n):  
        t.fd(step_length)  
        t.lt(step_angle)  
  
arc(bob, 50, 90)
```

[Mundo das Tartarugas](#)[Mundo das Tartarugas](#)[Encapsulation](#)[Generalização](#)[Interface Design](#)[Refactoring](#)

Exemplo

```
def polyline(t, n, length, angle):  
    for i in range(n):  
        t.fd(length)  
        t.lt(angle)  
  
def polygon(t, n, length):  
    angle = 360.0 / n  
    polyline(t, n, length, angle)  
  
def arc(t, r, angle):  
    arc_length = 2 * math.pi * r * angle / 360  
    n = int(arc_length / 3) + 1  
    step_length = arc_length / n  
    step_angle = float(angle) / n  
    polyline(t, n, step_length, step_angle)  
  
def circle(t, r):  
    arc(t, r, 360)
```