

Programação em Python

Módulo turtle

2023

Departamento de Ciência de Computadores

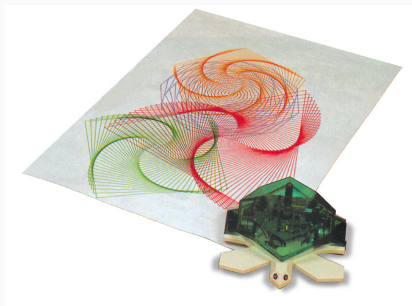


Módulo turtle

Módulo turtle

Vamos criar programas que fazem desenhos usando o módulo *turtle*:

- o programa controla um *robot* virtual (tartaruga);
- desloca-se para frente, para trás e roda sobre si próprio;
- usa uma caneta para deixar um rasto;
- muito simples, mas permite fazer desenhos impressionantes.



Primeiros passos

Devemos começar por importar o módulo:

```
>>> import turtle
```

Os comandos têm a forma `turtle.comando(...)`:

```
>>> turtle.clear()           # limpar a janela
>>> turtle.forward(100)      # avançar 100 pixels
>>> turtle.left(90)          # rodar 90 graus a esquerda
>>> turtle.forward(200)      # avançar 200 pixels
```

Primeiros passos (cont.)

Em alternativa, podemos usar

```
>>> from turtle import *
```

e omitir o nome do módulo:

```
>>> clear()
>>> forward(100)
>>> left(90)
>>> forward(200)
```

Comando principais

forward(n) avançar n pixels

backward(n) retroceder n pixels

left(α) rodar α graus à esquerda

right(α) rodar α graus à direita

color(c) mudar a cor do traço

pensize(n) mudar a largura do traço

penup() levantar a caneta

pendown() baixar a caneta

speed(n) mudar a velocidade da tartaruga

clear() limpar a janela

reset() limpar a janela e re-inicializar a tartaruga

Desenhar um quadrado

Vamos definir uma função sem argumentos para desenhar um quadrado com 100 pixels de lado.

- Desenhar quatro lados, rodando 90° para a esquerda após cada lado
- Alternativa: também poderíamos rodar para a direita

Desenhar um quadrado (cont.)

```
def quadrado():  
    forward(100)    # primeiro lado  
    left(90)  
    forward(100)    # segundo lado  
    left(90)  
    forward(100)    # terceiro lado  
    left(90)  
    forward(100)    # quarto lado  
    left(90)        # terminar na orientação original
```

Evitando repetições

- Repetimos quatro vezes as instruções:

```
forward(100)  
left(90)
```

- Podemos evitar a repetição usando um **ciclo** *for*

Evitando repetições (cont.)

```
def quadrado():  
    for lado in [1,2,3,4]:    # repetir 4 vezes  
        forward(100)  
        left(90)
```

- A variável *lado* não é usada dentro do ciclo
- Poderíamos usar qualquer outra lista de quatro valores

Alternativas

```
def quadrado():  
    for i in [1,2,3,4]:  
        forward(100)  
        left(90)
```

```
def quadrado():  
    for i in range(4): # 0, 1, 2, 3  
        forward(100)  
        left(90)
```

```
def quadrado():  
    for c in ['red','green','blue','black']:  
        color(c)  
        forward(100)  
        left(90)
```

Vamos agora generalizar a função para desenhar um quadrado de qualquer lado.

Basta tomar a medida do lado como um parâmetro da função.

Generalizando (cont.)

```
def quadrado(lado):  
    'Desenhar um quadrado dado o comprimento do lado.'  
    for i in range(4):  
        forward(lado)  
        left(90)
```

Exemplo final

espiral.py

```
from turtle import *

def quadrado(lado):
    for c in ['red', 'blue', 'green', 'black']:
        color(c)
        forward(lado)
        left(90)

reset()
speed(10)
for i in range(36):
    quadrado(50+i*5)
    left(10)
```
