

[◀ VOLTAR](#)

Funções de preenchimento

Apresentar o objeto turtle que trabalha com funções de preenchimento.

NESTE TÓPICO

> TRABALHANDO COM FUNÇÕES DENOMINADAS

> Dê uma olhada nos links abaixo para saber mais sobre a linguagem Python:



Marcar
tópico



Olá alunos,

Vamos ver uma parte lúdica na linguagem Python. Para quem já trabalhou ou conhece o LOGO, a tartaruga, criada pelo Prof. Seymour Papert na década de 60, o LOGO era uma espécie de linguagem de programação para crianças, quando os recursos computacionais eram ainda bem precários na época, a ideia do professor Papert era ajudar a construção do conhecimento (o construcionismo) com auxílio da computação, “o aprender fazendo”, era o seu lema.

A linguagem Python já traz implantada em sua biblioteca a classe **turtle** (tartaruga) com muitas funções. Vamos ver algumas delas, mas incentivo a explorar mais os recursos dessa classe.

Vamos ver um exemplo:

```
1. from turtle import *
2. # movimentando a tartaruga
3. shape("turtle") # coloca o formato da tartaruga
4. forward(200) # anda para frente 100 pixels
5. right(90) # vira 90 graus para direita
6. forward(200)
7. right(90)
8. forward(200)
9. right(90)
10. forward(200)
```

Na linha **1**, pegamos todas as funções do objeto **turtle**, lembre-se que o asterisco é um coringa que representa tudo, no caso, todas as funções da tartaruga.

Na linha **3**, a função **shape** com o argumento **turtle** coloca o desenho da tartaruga que será movimentada.

Nas sequencias das próximas linhas, fizemos a tartaruga andar 200 pixels para frente (**forward**) e cada vez que se movimenta, a tartaruga deixa um rastro (traço) mas, podemos também desabilitar esse rastro. Depois, pedimos para a tartaruga virar 90 graus para direita (linha **5**). Depois vai 200 pixels para frente, 90 graus à direita e assim até a linha **10**.

Este código desenha um quadrado.

Veja o vídeo abaixo:

./videos/370377864.mp4



Agora, e se quiséssemos desenhar um retângulo?

```
1. from turtle import *
2. # movimentando a tartaruga
3. shape("turtle") # coloca o formato da tartaruga
4. delay(50)
5. forward(100) # anda para frente 100 pixels
6. right(90) # vira 90 graus para direita
7. forward(200)
8. right(90)
9. forward(100)
10. right(90)
11. forward(200)
```

Veja no vídeo abaixo, a tartaruga desenhando um retângulo::

./videos/370394381.mp4



Mudamos os valores de duas funções **forward(100)** para desenhar um retângulo. Também adicionamos na linha **4** a função **delay(50)** (atrasar 50 milissegundos) para a tartaruga desenhar mais devagar.

```
1. from turtle import *
2. # movimentando a tartaruga
3. title("TARTARUGA TOUCHE!") # coloca o título na janela
4. bgcolor("lightyellow") # coloca a cor de fundo da janela em amarelo claro
5. pensize(4) # define a espessura do traço
6. color("blue") # cor da tartaruga e do traço
7. shape("turtle") # coloca o formato da tartaruga
8. delay(50)
9. for cont in range(4):
10.     forward(200) # anda para frente 100 pixels
11.     right(90) # vira 90 graus para direita
```

Veja agora no vídeo abaixo, como ficou:

./videos/370225197.mp4



Acrescenta mais algumas funções da biblioteca: **title**, **bgcolor**, **pensize** e **color**. Observe os comentários após a declaração das funções para você ver para que servem cada uma delas. Colocamos um bloco **for** (linha 9) para simplificar o código na hora de desenhar um quadrado, o loop vai realizar quatro vezes os movimentos da tartaruga.

Agora vamos trabalhar com o código, criando nossas funções:

```
1. from turtle import *
2. # movimentando a tartaruga
3.
4. def quadrado(graus,tamanho):
5.     for cont in range(4):
6.         forward(tamanho)
7.         right(graus)
8.
9. def triangulo(graus,tamanho):
10.    for i in ["red","violet","orange"]:
11.        color(i)
12.        left(graus)
13.        forward(tamanho)
14.
15. def touche():
16.    title("TARTARUGA TOUCHE!") # coloca o título na janela
17.    bgcolor("lightyellow") # coloca a cor de fundo da janela em amarelo claro
18.    pensize(4) # define a espessura do traço
19.    color("blue") # cor da tartaruga e do traço
20.    shape("turtle") # coloca o formato da tartaruga
21.    delay(50)
22.    quadrado(90,200)
23.    triangulo(120,150)
24.
25. touche() # Chama a função
26. exitonclick() # aguarda o clique do mouse para encerrar
```

TRABALHANDO COM FUNÇÕES DENOMINADAS

No exemplo acima, aplicamos vários recursos que já aprendemos, agora dividimos o código e colocamos tudo em funções. Criamos as funções: **touche**, **quadrado** e **triangulo**.

Vamos criar um script com o nome **tartaruga.py**, ao executar este script, a função **touche()** define o título, cor de fundo, etc,

Depois é chamada a função **quadrado** (linha **22**) e é executada, depois a função **triangulo** é chamada (linha **23**) e executada a partir da função **touche**.

Na função **triangulo**, a função **touche** passa dois argumentos: 120 graus e o tamanho de 150 pixels de lado, o comando **for** (linha **10**) tem um contador **i** que irá receber uma lista com três valores: red, violet, orange. Na linha **11**, a função **color** recebe como valor o contador **i**, isto significa que o loop será realizado três vezes e a cada vez a função **color** mudará de cor (vermelho, violeta e laranja), que irá trocando a cor do lado do triângulo.

Note que a execução das instruções é sequencial, mas com a divisão do código em funções, isto já não acontece. Quando executamos o script **tartaruga.py**, chamamos para executar a função **touche()**, o código começa a ser executado a partir da linha **25**. E depois a função **touche()** chama a função **quadrado** e a função **triangulo**.

Depois da tartaruga desenhar o quadrado e o triângulo, a função **exitonclick()**, linha **26**, aguardará o clique do mouse para fechar a aplicação.

Recomendo que vejam as demais aplicações da turtle pelo Windows: Python 3.n, Python 3.n Manuals, na busca digite: turtle.

Veja no vídeo abaixo, o resultado final:

./videos/370208724.mp4



SAIBA MAIS...

Dê uma olhada nos links abaixo para saber mais sobre a linguagem Python:

<https://www.python.org/doc/> (<https://www.python.org/doc/>)

<https://wiki.python.org/moin/PythonBooks>
(<https://wiki.python.org/moin/PythonBooks>)

Neste tópico utilizamos o objeto turtle do Python para iniciar com a parte gráfica e principalmente para trabalharmos com a criação e a instanciação de funções denominadas por nós.

Quiz

Exercício Final

Funções de preenchimento

INICIAR ➤

Referências

SUMMERFIELD, M. *Programação em Python 3: Uma introdução completa à linguagem Python*. Rio de Janeiro Alta Books, 2012. 495 p.

MENEZES, N. N. C. *Introdução à programação com Python: algoritmos e lógica de programação para iniciantes*. 2. ed. São Paulo: Novatec, 2014. 328 p.

SWEIGART, AL. *Automatize tarefas maçantes com Python: programação prática para verdadeiros iniciantes*. São Paulo: Novatec, 2015. 568 p.

PYTHON, doc. Disponível em: <<https://www.python.org/doc/>>. Acesso em: Junho/2018.

PYTHON, books. Disponível em: <<https://wiki.python.org/moin/PythonBooks>>. Acesso em: Junho/2018.



Avalie este tópico



ANTERIOR

Funções personalizadas

Biblioteca

(<https://www.uninove.br/conhec-a-uninove/biblioteca/sobre-a-biblioteca/apresentacao/>)

Portal Uninove

(<http://www.uninove.br>)

Mapa do Site

Índice

Ajuda?

PRÓXIMO

(<https://ava.uninove.br/cursos/>)

Fundamentos da rede em Python

© Todos os direitos reservados