

python

The Python logo, consisting of two interlocking snakes, one blue and one yellow, is positioned below the word "python".

```
import turtle
turtle.setup(650,350,200,200)
turtle.penup()
turtle.fd(-250)
turtle.pendown()
turtle.pensize(25)
turtle.pencolor("purple")
for i in range(4):
    turtle.circle(40, 80)
    turtle.circle(-40, 80)
    turtle.circle(40, 80/2)
    turtle.fd(40)
    turtle.circle(16, 180)
    turtle.fd(40 * 2/3)
```

Python语言程序设计

turtle程序语法元素分析



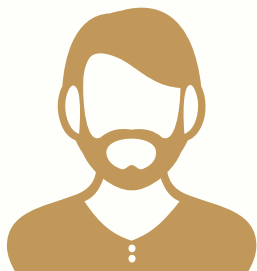
嵩 天
北京理工大学





单元开篇

turtle程序语法元素分析



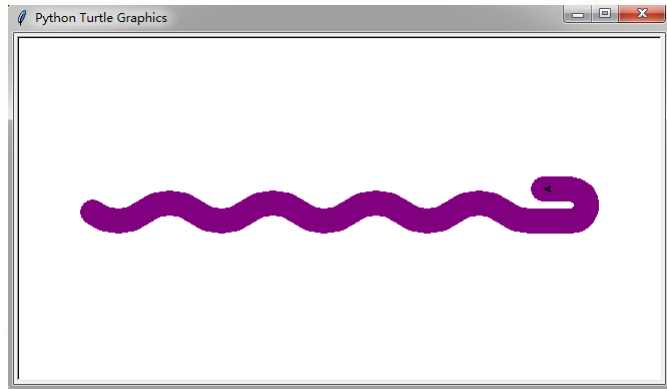
- 库引用与import
- turtle画笔控制函数
- turtle运动控制函数
- turtle方向控制函数
- 基本循环语句
- "Python蟒蛇绘制"代码分析





库引用与import

```
import turtle
turtle.setup(650, 350, 200, 200)
turtle.penup()
turtle.fd(-250)
turtle.pendown()
turtle.pensize(25)
turtle.pencolor("purple")
turtle.seth(-40)
for i in range(4):
    turtle.circle(40, 80)
    turtle.circle(-40, 80)
turtle.circle(40, 80/2)
turtle.fd(40)
turtle.circle(16, 180)
turtle.fd(40 * 2/3)
turtle.done()
```



<a>.()的编码风格

库引用

扩充Python程序功能的方式

- 使用**import**保留字完成，采用<a>.()编码风格

import <库名>

<库名>.<函数名>(<函数参数>)

```
import turtle
turtle.setup(650, 350, 200, 200)
turtle.penup()
turtle.fd(-250)
turtle.pendown()
turtle.pensize(25)
turtle.pencolor("purple")
turtle.seth(-40)
for i in range(4):
    turtle.circle(40, 80)
    turtle.circle(-40, 80)
turtle.circle(40, 80/2)
turtle.fd(40)
turtle.circle(16, 180)
turtle.fd(40 * 2/3)
turtle.done()
```

引入turtle库

使用turtle库函数

完成功能

可是可是, 好多turtle, 很繁琐嘛...

import更多用法

使用from和import保留字共同完成

from <库名> import <函数名>

from <库名> import *

<函数名>(<函数参数>)

```
import turtle
turtle.setup(650, 350, 200, 200)
turtle.penup()
turtle.fd(-250)
turtle.pendown()
turtle.pensize(25)
turtle.pencolor("purple")
turtle.seth(-40)
for i in range(4):
    turtle.circle(40, 80)
    turtle.circle(-40, 80)
turtle.circle(40, 80/2)
turtle.fd(40)
turtle.circle(16, 180)
turtle.fd(40 * 2/3)
turtle.done()
```



```
from turtle import *
setup(650, 350, 200, 200)
penup()
fd(-250)
pendown()
pensize(25)
pencolor("purple")
seth(-40)
for i in range(4):
    circle(40, 80)
    circle(-40, 80)
circle(40, 80/2)
fd(40)
circle(16, 180)
fd(40 * 2/3)
done()
```

老师老师, 这么好的方法为何不早说...

import更多用法

两种方法比较

import <库名>

<库名>.<函数名>(<函数参数>)

from <库名> **import** <函数名>

from <库名> **import** *

<函数名>(<函数参数>)

第一种方法不会出现函数重名问题，第二种方法则会出现

import更多用法

使用import和as保留字共同完成

import <库名> **as** <库别名>

<库别名>.<函数名>(<函数参数>)

给调用的外部库关联一个更短、更适合自己的名字

```
import turtle
turtle.setup(650, 350, 200, 200)
turtle.penup()
turtle.fd(-250)
turtle.pendown()
turtle.pensize(25)
turtle.pencolor("purple")
turtle.seth(-40)
for i in range(4):
    turtle.circle(40, 80)
    turtle.circle(-40, 80)
turtle.circle(40, 80/2)
turtle.fd(40)
turtle.circle(16, 180)
turtle.fd(40 * 2/3)
turtle.done()
```



```
import turtle as t
t.setup(650, 350, 200, 200)
t.penup()
t.fd(-250)
t.pendown()
t.pensize(25)
t.pencolor("purple")
t.seth(-40)
for i in range(4):
    t.circle(40, 80)
    t.circle(-40, 80)
t.circle(40, 80/2)
t.fd(40)
t.circle(16, 180)
t.fd(40 * 2/3)
t.done()
```

这个方法好!



turtle画笔控制函数

```
import turtle
turtle.setup(650, 350, 200, 200)
turtle.penup()
turtle.fd(-250)
turtle.pendown()
turtle.pensize(25)
turtle.pencolor("purple")
turtle.seth(-40)
for i in range(4):
    turtle.circle(40, 80)
    turtle.circle(-40, 80)
turtle.circle(40, 80/2)
turtle.fd(40)
turtle.circle(16, 180)
turtle.fd(40 * 2/3)
turtle.done()
```

penup(), pendown()

pensize(), pencolor()

画笔控制函数

画笔操作后一直有效，一般成对出现

- **turtle.penup()** 别名 **turtle.pu()**

抬起画笔，海龟在飞行

- **turtle.pendown()** 别名 **turtle.pd()**

落下画笔，海龟在爬行

画笔控制函数

画笔设置后一直有效，直至下次重新设置

- `turtle.pensize(width)` 别名 `turtle.width(width)`

画笔宽度，海龟的腰围

- `turtle.pencolor(color)` `color`为颜色字符串或r,g,b值

画笔颜色，海龟在涂装

画笔控制函数

pencolor(color)的color参与可以有三种形式

- 颜色字符串 : `turtle.pencolor("purple")`
- RGB的小数值: `turtle.pencolor(0.63, 0.13, 0.94)`
- RGB的元组值: `turtle.pencolor((0.63,0.13,0.94))`

```
import turtle
turtle.setup(650, 350, 200, 200)
turtle.penup()
turtle.fd(-250)
turtle.pendown()
turtle.pensize(25)
turtle.pencolor("purple")
turtle.seth(-40)
for i in range(4):
    turtle.circle(40, 80)
    turtle.circle(-40, 80)
turtle.circle(40, 80/2)
turtle.fd(40)
turtle.circle(16, 180)
turtle.fd(40 * 2/3)
turtle.done()
```

penup()

pendown()

pensize(width)

pencolor(colorstring)

pencolor(r,g,b)

pencolor((r,g,b))





turtle运动控制函数

```
import turtle
turtle.setup(650, 350, 200, 200)
turtle.penup()
turtle.fd(-250)
turtle.pendown()
turtle.pensize(25)
turtle.pencolor("purple")
turtle.seth(-40)
for i in range(4):
    turtle.circle(40, 80)
    turtle.circle(-40, 80)
turtle.circle(40, 80/2)
turtle.fd(40)
turtle.circle(16, 180)
turtle.fd(40 * 2/3)
turtle.done()
```

fd()

circle()

运动控制函数

控制海龟行进：走直线 & 走曲线

- `turtle.forward(d)` 别名 `turtle.fd(d)`

向前行进，海龟走直线

- `d`: 行进距离，可以为负数

运动控制函数

控制海龟行进：走直线 & 走曲线

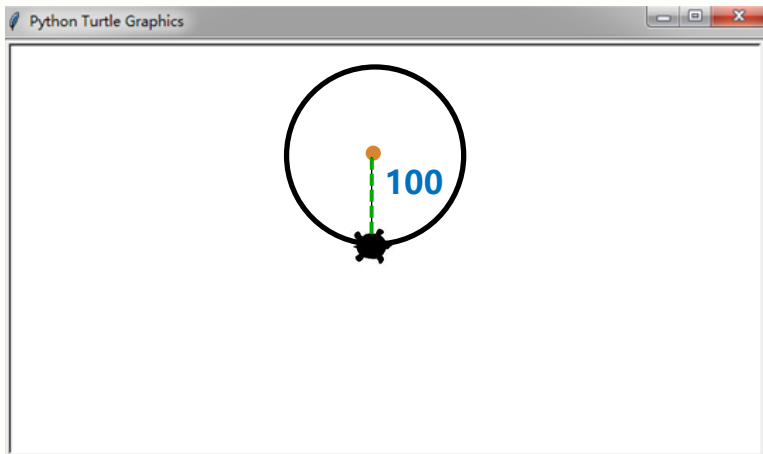
- `turtle.circle(r, extent=None)`

根据半径`r`绘制`extent`角度的弧形

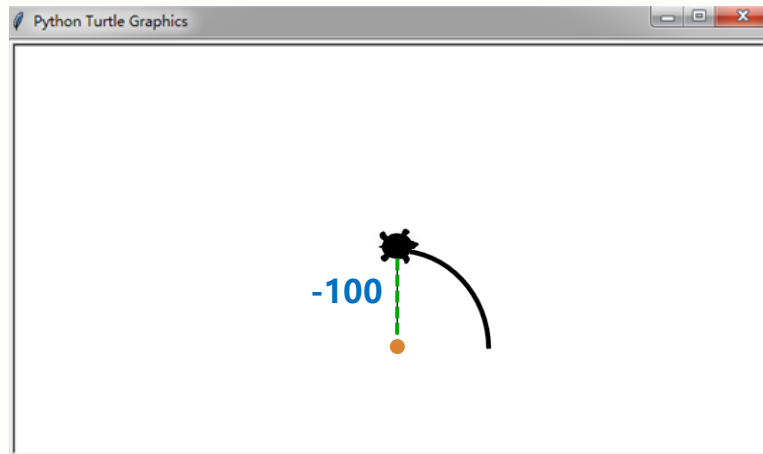
- `r`: 默认圆心在海龟左侧`r`距离的位置
- `extent`: 绘制角度，默认是360度整圆

运动控制函数

```
turtle.circle(100)
```



```
turtle.circle(-100,90)
```




```
import turtle
turtle.setup(650, 350, 200, 200)
turtle.penup()
turtle.fd(-250)
turtle.pendown()
turtle.pensize(25)
turtle.pencolor("purple")
turtle.seth(-40)
for i in range(4):
    turtle.circle(40, 80)
    turtle.circle(-40, 80)
turtle.circle(40, 80/2)
turtle.fd(40)
turtle.circle(16, 180)
turtle.fd(40 * 2/3)
turtle.done()
```

fd(d)

circle(r, extent=None)

运动控制函数

画笔设置后一直有效，直至下次重新设置

- `turtle.forward(d)` 别名 `turtle.fd(d)`

向前行进，海龟走直线

- `d`: 行进距离，可以为负数



turtle方向控制函数

```
import turtle
turtle.setup(650, 350, 200, 200)
turtle.penup()
turtle.fd(-250)
turtle.pendown()
turtle.pensize(25)
turtle.pencolor("purple")
turtle.seth(-40)
for i in range(4):
    turtle.circle(40, 80)
    turtle.circle(-40, 80)
turtle.circle(40, 80/2)
turtle.fd(40)
turtle.circle(16, 180)
turtle.fd(40 * 2/3)
turtle.done()
```

seth()

方向控制函数

控制海龟面对方向: 绝对角度 & 海龟角度

- `turtle.setheading(angle)` 别名 `turtle.seth(angle)`

改变行进方向，海龟走角度

- **angle**: 行进方向的绝对角度

方向控制函数

`turtle.seth(45)`



`turtle.seth(-135)`



方向控制函数

控制海龟面对方向: 绝对角度 & 海龟角度

- `turtle.left(angle)` 海龟向左转
- `turtle.right(angle)` 海龟向右转
- **angle**: 在海龟当前行进方向上旋转的角度

```
import turtle
turtle.setup(650, 350, 200, 200)
turtle.penup()
turtle.fd(-250)
turtle.pendown()
turtle.pensize(25)
turtle.pencolor("purple")
turtle.seth(-40)
for i in range(4):
    turtle.circle(40, 80)
    turtle.circle(-40, 80)
turtle.circle(40, 80/2)
turtle.fd(40)
turtle.circle(16, 180)
turtle.fd(40 * 2/3)
turtle.done()
```

seth(angle)



循环语句与range()函数

```
import turtle
turtle.setup(650, 350, 200, 200)
turtle.penup()
turtle.fd(-250)
turtle.pendown()
turtle.pensize(25)
turtle.pencolor("purple")
turtle.seth(-40)
for i in range(4):
    turtle.circle(40, 80)
    turtle.circle(-40, 80)
turtle.circle(40, 80/2)
turtle.fd(40)
turtle.circle(16, 180)
turtle.fd(40 * 2/3)
turtle.done()
```

for 和 in 保留字

range()

循环语句

按照一定次数循环执行一组语句

for <变量> **in** range(<次数>):

<被循环执行的语句>

- <变量>表示每次循环的计数，0到<次数>-1

循环语句

```
>>> for i in range(5):  
    print(i)
```

0
1
2
3
4

```
>>> for i in range(5):  
    print("Hello:",i)
```

Hello: 0
Hello: 1
Hello: 2
Hello: 3
Hello: 4

range()函数

产生循环计数序列

- range(N)

产生 0 到 N-1的整数序列，共N个

range(5)

0, 1, 2, 3, 4

- range(M, N)

产生 M 到 N-1的整数序列，共N-M个

range(2, 5)

2, 3, 4

```
import turtle
turtle.setup(650, 350, 200, 200)
turtle.penup()
turtle.fd(-250)
turtle.pendown()
turtle.pensize(25)
turtle.pencolor("purple")
turtle.seth(-40)
for i in range(4):
    turtle.circle(40, 80)
    turtle.circle(-40, 80)
turtle.circle(40, 80/2)
turtle.fd(40)
turtle.circle(16, 180)
turtle.fd(40 * 2/3)
turtle.done()
```

for i *in* range(N):

range(N)

range(M, N)

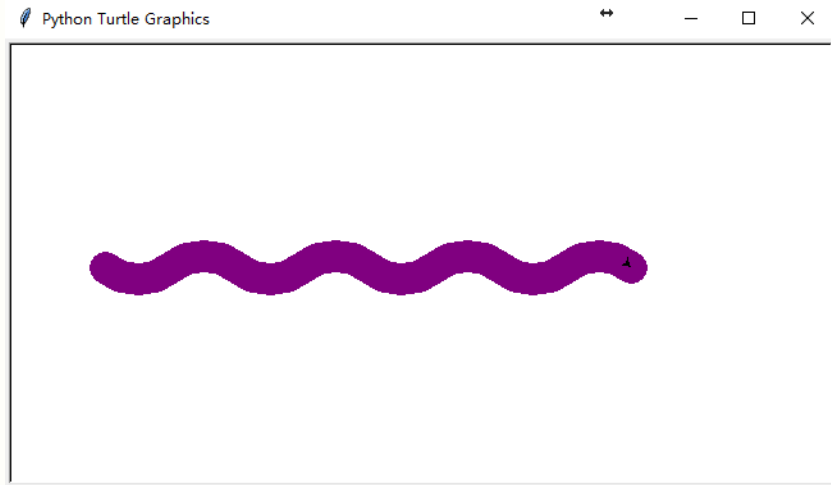


"Python蟒蛇绘制"代码分析

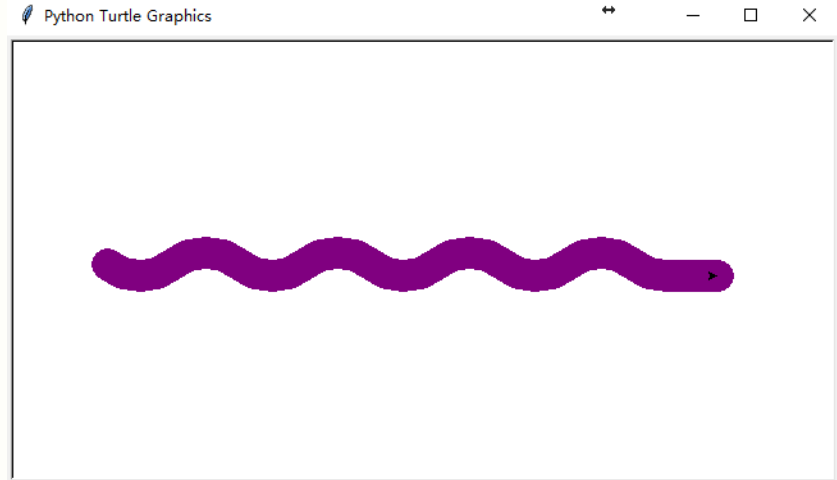
```
import turtle
turtle.setup(650, 350, 200, 200)
turtle.penup()
turtle.fd(-250)
turtle.pendown()
turtle.pensize(25)
turtle.pencolor("purple")
turtle.seth(-40)
for i in range(4):
    turtle.circle(40, 80)
    turtle.circle(-40, 80)
turtle.circle(40, 80/2)
turtle.fd(40)
turtle.circle(16, 180)
turtle.fd(40 * 2/3)
turtle.done()
```



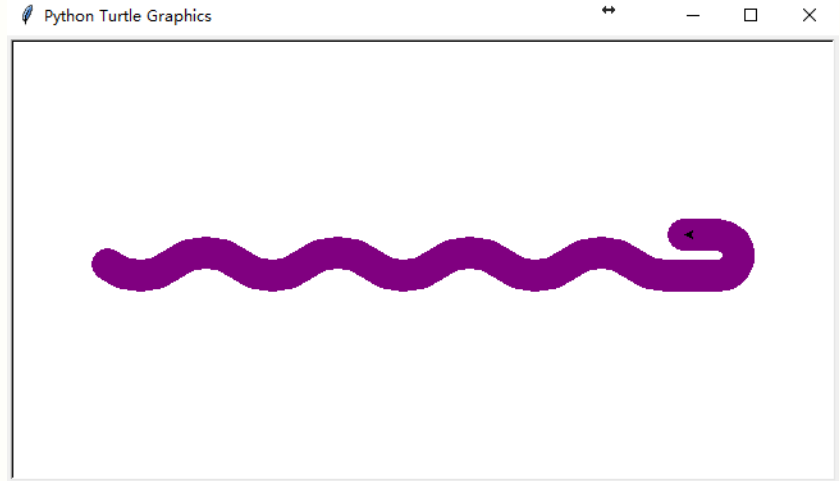

```
import turtle
turtle.setup(650, 350, 200, 200)
turtle.penup()
turtle.fd(-250)
turtle.pendown()
turtle.pensize(25)
turtle.pencolor("purple")
turtle.seth(-40)
for i in range(4):
    turtle.circle(40, 80)
    turtle.circle(-40, 80)
turtle.circle(40, 80/2)
turtle.fd(40)
turtle.circle(16, 180)
turtle.fd(40 * 2/3)
turtle.done()
```



```
import turtle
turtle.setup(650, 350, 200, 200)
turtle.penup()
turtle.fd(-250)
turtle.pendown()
turtle.pensize(25)
turtle.pencolor("purple")
turtle.seth(-40)
for i in range(4):
    turtle.circle(40, 80)
    turtle.circle(-40, 80)
turtle.circle(40, 80/2)
turtle.fd(40)
turtle.circle(16, 180)
turtle.fd(40 * 2/3)
turtle.done()
```



```
import turtle
turtle.setup(650, 350, 200, 200)
turtle.penup()
turtle.fd(-250)
turtle.pendown()
turtle.pensize(25)
turtle.pencolor("purple")
turtle.seth(-40)
for i in range(4):
    turtle.circle(40, 80)
    turtle.circle(-40, 80)
turtle.circle(40, 80/2)
turtle.fd(40)
turtle.circle(16, 180)
turtle.fd(40 * 2/3)
turtle.done()
```





单元小结

turtle程序语法元素分析

- 库引用: `import`、`from...import`、`import...as...`
- `penup()`、`pendown()`、`pensize()`、`pencolor()`
- `fd()`、`circle()`、`seth()`
- 循环语句: `for`和`in`、`range()`函数





小花絮

版权说明

- 本课程所有教学资料(课件)受CC BY-NC-SA 4.0知识产权协议保护
 - 未经授权不能商业使用、不能修改使用、不能格式转换
 - 非商业使用时(如教学)必须以恰当且明确方式声明原作者信息
 - 必须保持原有形式和内容进行使用和传播
- 保护知识产权、分享发展成果、建立良性创新机制

