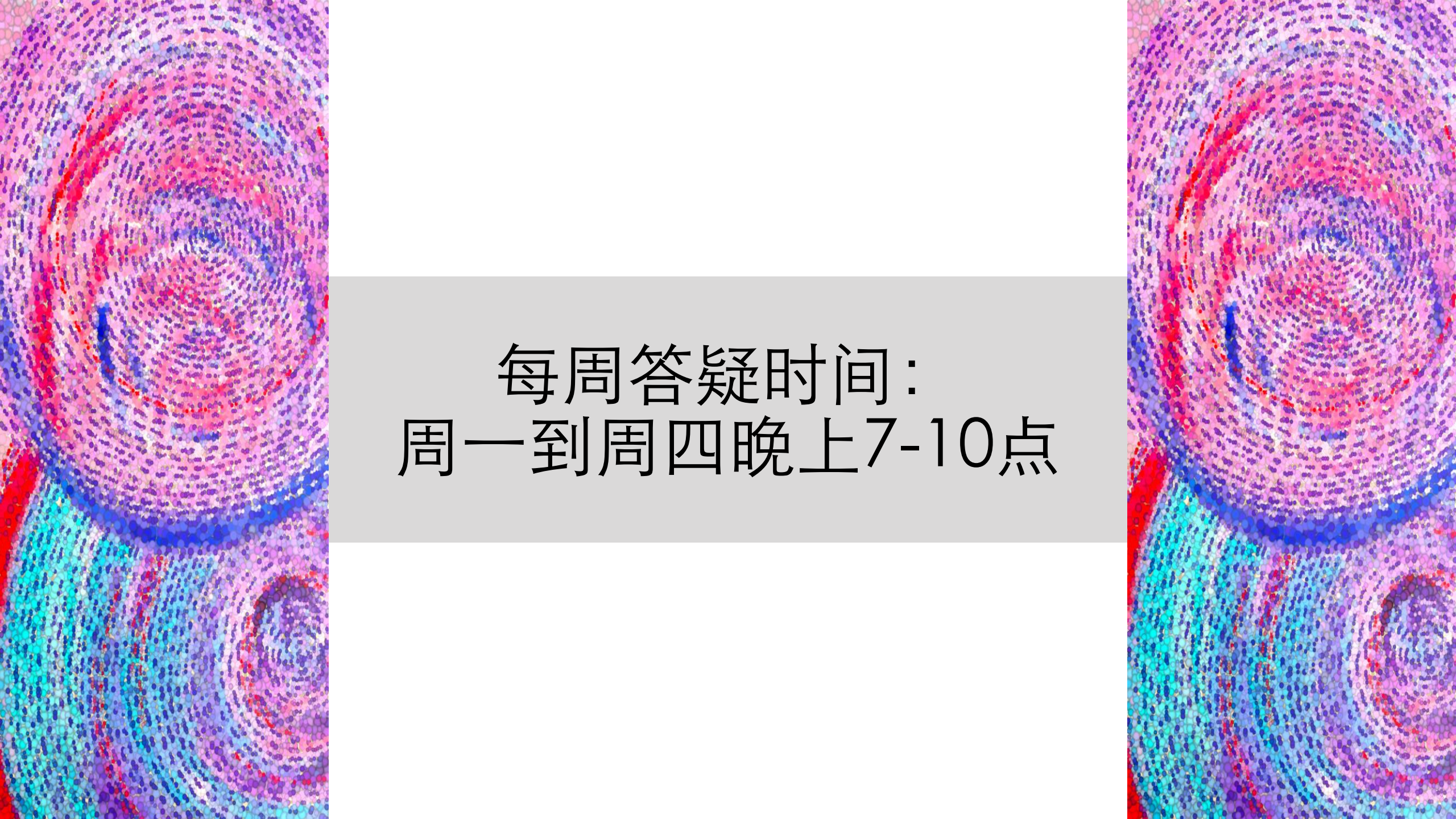




CS001 编程零基础 Python 语言入门

第三讲



每周答疑时间：
周一到周四晚上7-10点



作业回顾



命令行与.py文件区别

```
>>>3
```

```
3
```

```
>>>print(3)
```

```
3
```

```
>>> type(5.5)
```

```
<class 'float'>
```

```
>>>print(type(5.5))
```

```
<class 'float'>
```

Shell命令行会直接显示结果，不需要用任何输出语句，可以省去`print()`操作

.py文件的程序写作过程中，必须通过输出语句来显示结果，例如用标准输出`print()`

四舍五入

解法一：利用int()可以取整的效果

#solution1.py

```
s=input("Please input a number:")
```

```
number=float(s)
```

注意：input()输入的是字符串，
需要转换为数字才能计算

```
answer=int(number+0.5)
```

```
print(answer)
```

解法二：利用内置函数round()直接完成四舍五入

#solution2.py

```
print(round(float(input("Please input a  
number:"))))
```

温度转换

```
#temperature.py
```

```
c=float(input("Please input a Celsius degree:"))
```

```
f=9/5*c+32
```

```
print(f)
```

注意：input()输入的是字符串，需要转换为数字才能计算。解答中使用float()强制转换为浮点类型。

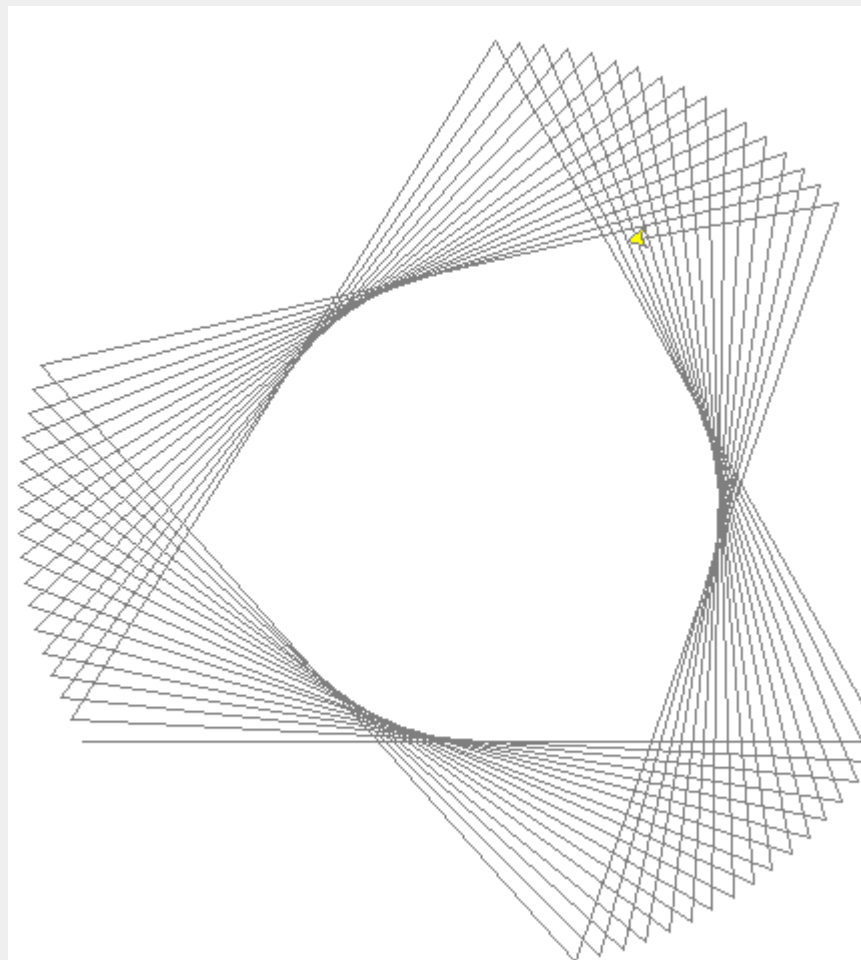
附加题：输入控制作图

```
from turtle import *  
size=int(input("size:"))  
angle=int(input("angle:"))
```

答案文件名draw2.py

```
colormode(255)  
color((127, 127, 127),(255,255,0))  
begin_fill()  
while True:  
    forward(size)  
    left(180-angle)  
    if abs(pos()) < 1:  
        break  
end_fill()  
done()
```

思考：left(180-angle) 和
right(angle)有什么区别？



变量(variable) 与 赋值(assignment)

变量：可以变化的量

变量的名字：必须以字母或下划线开头，区分大小写

赋值“=”：将某一数值赋给某个变量的过程

其实更严格的理解应该是：
“x=y” 是让x指向y这个对象

```
>>> __mike=666
```

```
>>> __MIKE=888
```

```
>>> print(__mike)
```

```
666
```

```
>>> print(__MIKE)
```

```
888
```

```
>>> x=__mike
```

```
>>> print(x)
```

```
666
```

```
>>> x=x+1
```

```
>>> print(x)
```

```
667
```

```
>>> x=x-2
```

```
>>> print(x)
```

```
665
```

```
>>> x=x*2
```

```
>>> print(x)
```

```
1330
```


变量(variable) 与 赋值(assignment)

```
>>> x+=1
>>> print(x)
1331
>>> x-=10
>>> print(x)
1321
>>> x*=2
>>> print(x)
2642
>>> x/=10
>>> print(x)
264.2
```

```
>>> a=[1,2,3]
>>> b=a
>>> b
[1, 2, 3]
>>> a[0]=66
>>> a
[66, 2, 3]
>>> b
[66, 2, 3]
>>> a=a+[4]
>>> a
[66, 2, 3, 4]
>>> b
[66, 2, 3]
```

a, b指向同一个列表对象

a, b不再指向同一个列表对象了

```
>>> c=a
>>> c
[66, 2, 3, 4]
>>> c[0]=1
>>> a
[1, 2, 3, 4]
>>> a=[5,6]
>>> a
[5, 6]
>>> c
[1, 2, 3, 4]
```

a, c指向同一个列表对象

a, c不再指向同一个列表对象了

```
>>> s="abc"
>>> t=s
>>> s
'abc'
>>> t
'abc'
>>> s=s+"d"
>>> s
'abcd'
>>> t
'abc'
```

多变量赋值技巧

```
>>> x,y=66,88
```

```
>>> print(x,y)
```

```
66 88
```

```
>>> print(x)
```

```
66
```

```
>>> print(y)
```

```
88
```

x, y 互换技巧

```
>>> x,y=y,x
```

```
>>> print(x,y)
```

```
88 66
```

```
>>> i,j,k=1,2,3
```

```
>>> i,j,k=j,k,i
```

```
>>> print(i,j,k)
```

```
2 3 1
```

```
>>> fibo,nacci=1,2
```

```
>>> fibo,nacci=nacci,fibo+nacci
```

```
>>> print(fibo,nacci)
```

```
2 3
```

```
>>> fibo,nacci=nacci,fibo+nacci
```

```
>>> print(fibo,nacci)
```

```
3 5
```

```
>>> fibo,nacci=nacci,fibo+nacci
```

```
>>> print(fibo,nacci)
```

```
5 8
```

斐波那契数列

列表类型补充 - list

```
>>> a=[1,"a",4.4,"xxx",[8,9,10],True]
```

```
>>> print(a)
```

```
[1, 'a', 4.4, 'xxx', [8, 9, 10], True]
```

```
>>> print(matrix)
```

```
[[0, 1, 2], [3, 4, 5], [6, 7, 8]]
```

```
>>> matrix[2][1]
```

```
7
```

```
>>> x=[5,4,8,9,10,12,88,3,2,1]
```

```
>>> sorted(x)
```

```
[1, 2, 3, 4, 5, 8, 9, 10, 12, 88]
```

```
>>> print(x)
```

```
[5, 4, 8, 9, 10, 12, 88, 3, 2, 1]
```

```
>>> x.sort()
```

```
>>> print(x)
```

```
[1, 2, 3, 4, 5, 8, 9, 10, 12, 88]
```


循环语句 - for

```
>>> things=["a pen","an apple","a pineapple"]
>>> for item in things:
    print("I have",item)
I have a pen
I have an apple
I have a pineapple
>>> s="hello"
>>> for letter in s:
    print(ord(letter))

104
101
108
108
111
```

```
>>> for i in range(10,0,-2):
    print(i)
>>> tot=0
>>> for i in range(10):
    tot+=i
>>> print(tot)
45
>>> tot=0
>>> for i in range(10):
    tot+=i*i
>>> print(tot)
285
>>> money=100
>>> for year in range(0,10):
    money=money*1.05
    print(money)
```

```
>>> a,b=1,1
>>> for i in range(10):
    print(b)
    a,b=b,a+b
1
2
3
5
8
13
21
34
55
89
```

斐波那契数列

多重循环 – for

```
>>> for i in range(3):
    for j in range(3):
        for k in range(3):
            print(i,j,k)

>>> for i in range(5):
    for j in range(i+1,5):
        for k in range(j+1,5):
            print(i,j,k)

>>> for i in range(1,4):
    for j in range(i+1,4):
        for k in range(j+1,4):
            print(i,j,k)
```

1 2 3

```
>>> t=[[1]]
>>> for i in range(10):
    print(t[i])
    t.append([1])
    for j in range(0,i):
        t[i+1].append(t[i][j]+t[i][j+1])
    t[i+1].append(1)
```

```
[1]
[1, 1]
[1, 2, 1]
[1, 3, 3, 1]
[1, 4, 6, 4, 1]
[1, 5, 10, 10, 5, 1]
[1, 6, 15, 20, 15, 6, 1]
[1, 7, 21, 35, 35, 21, 7, 1]
[1, 8, 28, 56, 70, 56, 28, 8, 1]
[1, 9, 36, 84, 126, 126, 84, 36, 9, 1]
```

杨辉三角形

循环语句的应用：解方程

固定点迭代法 `fixedpoint.py`

Fixed Point Iteration

求解方程 $x^3 - x - 1 = 0$

第一步：移项 $x = (x+1)^{1/3}$

第二步：初始化 $x=0$

第三步：计算 $y=(x+1)^{1/3}$

第四步：赋值使 $x=y$

迭代：重复第三步和第四步
直到原方程的解足够精确

请打开该程序 `fixedpoint.py`

运行并查看结果，试图对其做修改来解其他方程

```
#solve  $x^3 - x - 1 = 0$  by fixed point iteration
```

```
#let  $x \leftarrow (x+1)^{1/3}$ 
```

```
x=0.0
```

```
for i in range(12):
```

```
    y=pow(x+1,1/3)
```

```
    print("#",i," iteration: x=%.8f and  $x^3 - x - 1 = %.8f$ "%(x,y-x))
```

```
    x=y
```


导入模块/库 - import

每个模块/库 (module/library) 里汇聚了很多解决问题的方法和工具，就像是一个个的工具箱里面有很多各种各样的工具。

通过import语句可以把特定模块导入你的程序，帮助你高效地解决问题。

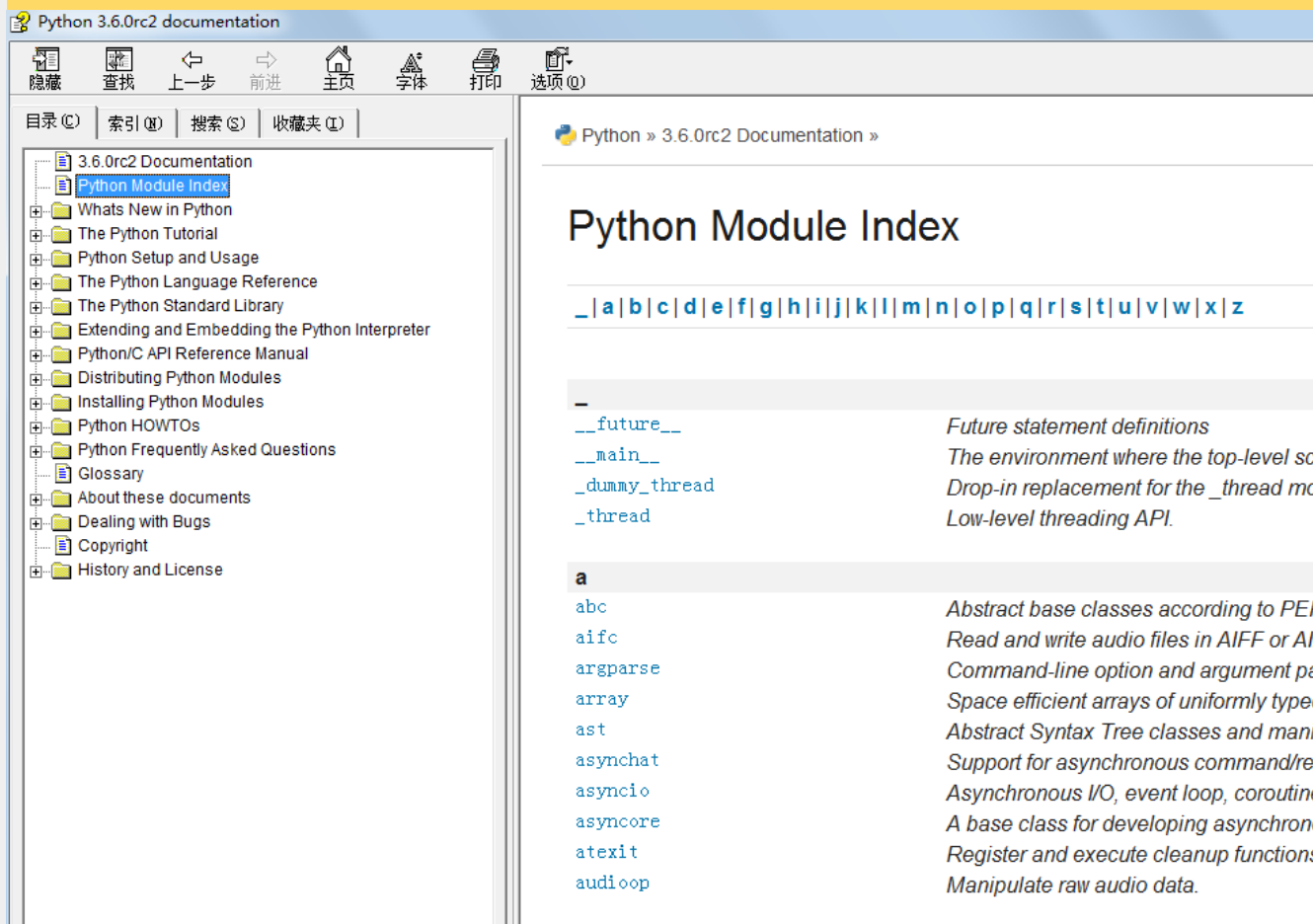
import turtle 导入海龟作图工具模块

import time 导入时间工具模块

import random 导入随机工具模块

import math 导入数学工具模块

各类模块的汇总，在帮助中可以找到



The screenshot shows the Python 3.6.0rc2 documentation website. The left sidebar contains a table of contents with links to various documentation sections. The main content area displays the 'Python Module Index' with a alphabetical navigation bar and a list of modules under the letter 'a'.

Python 3.6.0rc2 documentation

隐藏 查找 上一步 前进 主页 字体 打印 选项 (Q)

目录 (C) | 索引 (I) | 搜索 (S) | 收藏夹 (U)

- 3.6.0rc2 Documentation
- Python Module Index**
- Whats New in Python
- The Python Tutorial
- Python Setup and Usage
- The Python Language Reference
- The Python Standard Library
- Extending and Embedding the Python Interpreter
- Python/C API Reference Manual
- Distributing Python Modules
- Installing Python Modules
- Python HOWTOs
- Python Frequently Asked Questions
- Glossary
- About these documents
- Dealing with Bugs
- Copyright
- History and License

Python » 3.6.0rc2 Documentation »

Python Module Index

[_](#) [a](#) [b](#) [c](#) [d](#) [e](#) [f](#) [g](#) [h](#) [i](#) [j](#) [k](#) [l](#) [m](#) [n](#) [o](#) [p](#) [q](#) [r](#) [s](#) [t](#) [u](#) [v](#) [w](#) [x](#) [z](#)

<code>__future__</code>	Future statement definitions
<code>__main__</code>	The environment where the top-level script is executed
<code>_dummy_thread</code>	Drop-in replacement for the <code>_thread</code> module
<code>_thread</code>	Low-level threading API.

a

<code>abc</code>	Abstract base classes according to PEP 311
<code>aifc</code>	Read and write audio files in AIFF or AU format
<code>argparse</code>	Command-line option and argument parsing
<code>array</code>	Space efficient arrays of uniformly typed data
<code>ast</code>	Abstract Syntax Tree classes and methods
<code>asynchat</code>	Support for asynchronous command-line interaction
<code>asyncio</code>	Asynchronous I/O, event loop, coroutine
<code>asyncore</code>	A base class for developing asynchronous network applications
<code>atexit</code>	Register and execute cleanup functions
<code>audioop</code>	Manipulate raw audio data.

时间工具模块 – import time

循环计时程序looptime.py

```
import time

for i in range(9):
    LIMIT=10**i
    start=time.time()    记录开始时间
    for j in range(LIMIT):
        x=1
    end=time.time()      记录结束时间
    print(str(LIMIT)+" loops take "+str(end-start)+" seconds")
```

```
1 loops take 0.0 seconds
10 loops take 0.0 seconds
100 loops take 0.0 seconds
1000 loops take 0.0 seconds
10000 loops take 0.0010001659393310547 seconds
100000 loops take 0.010000228881835938 seconds
1000000 loops take 0.09800553321838379 seconds
10000000 loops take 0.9490542411804199 seconds
100000000 loops take 9.421539068222046 seconds
>>> |
```

时间工具模块 – import time

打字计时程序typingtime.py

```
import time
```

```
#TEXT="Hello everybody! I have to tell you I love programming."  
TEXT="abcdefghijklmnopqrstuvwxyz"
```

```
start=time.time()
```

记录开始时间

```
s=input("Please type:"+TEXT+"\n")
```

```
end=time.time()
```

记录结束时间

```
print("You took "+str(end-start)+" seconds to type the above.")
```

```
print("The matching result is "+ str(s==TEXT))
```


时间工具模块 和海龟作图模块 – time, turtle, 循环

倒计时程序countdown.py

```
import turtle
import time
p=turtle.Turtle()
for i in range(10,0,-1):
    p.write(i,font=("Arial", 50, "normal"))
    time.sleep(1)
    p.clear()
p.write("BLAST OFF!!",font=("Arial", 50, "normal"))
```

现代艺术画自动生成程序I – 利用turtle, random, 循环

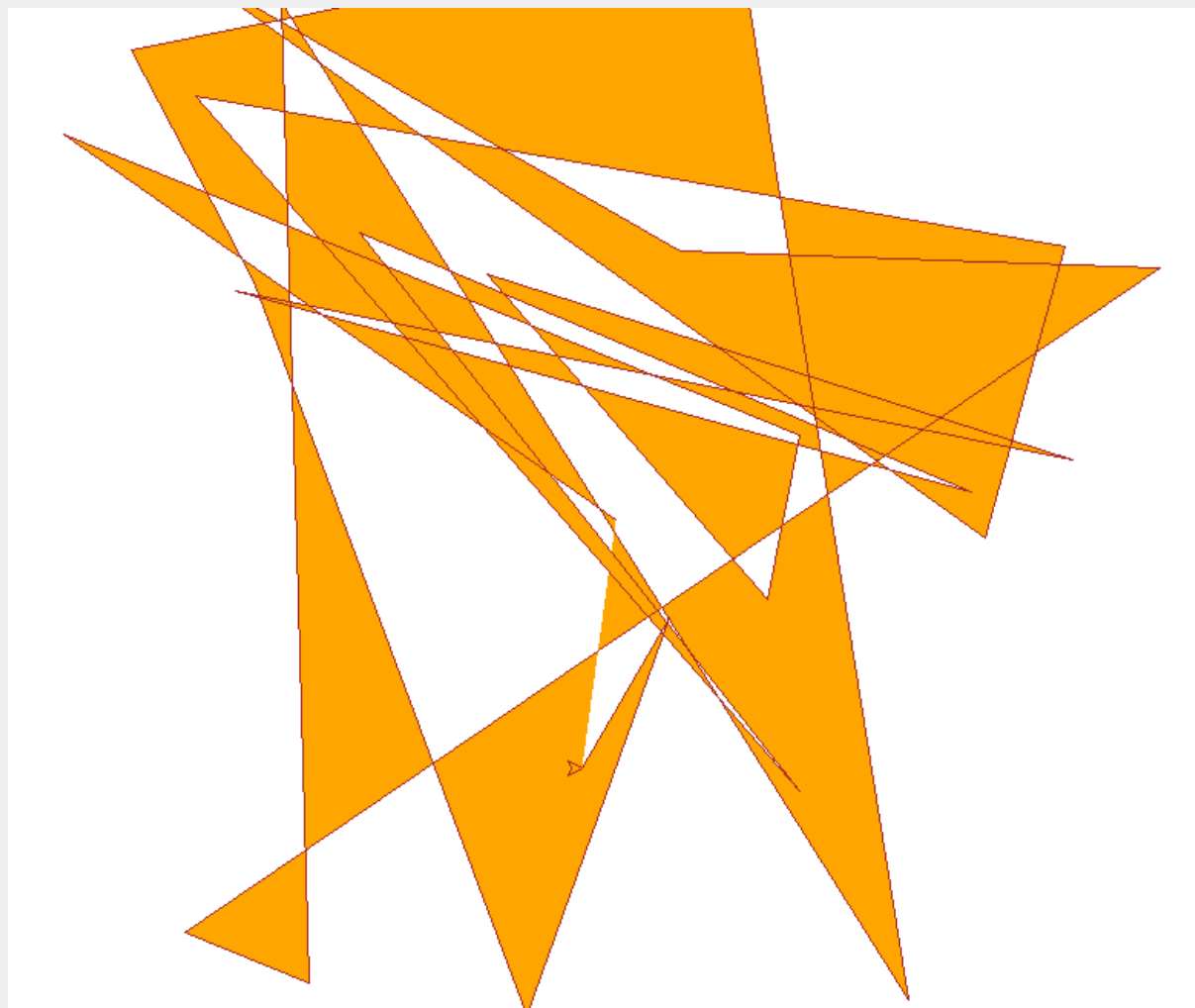
程序文件的文件名为randomlines.py
可以用IDLE打开并且运行

```
import turtle  
import random
```

```
NUM=25  
XMAX=400  
YMAX=400
```

```
for i in range(NUM):  
    x=random.randrange(-XMAX,XMAX)  
    y=random.randrange(-YMAX,YMAX)  
    turtle.goto(x,y)
```

随机位置

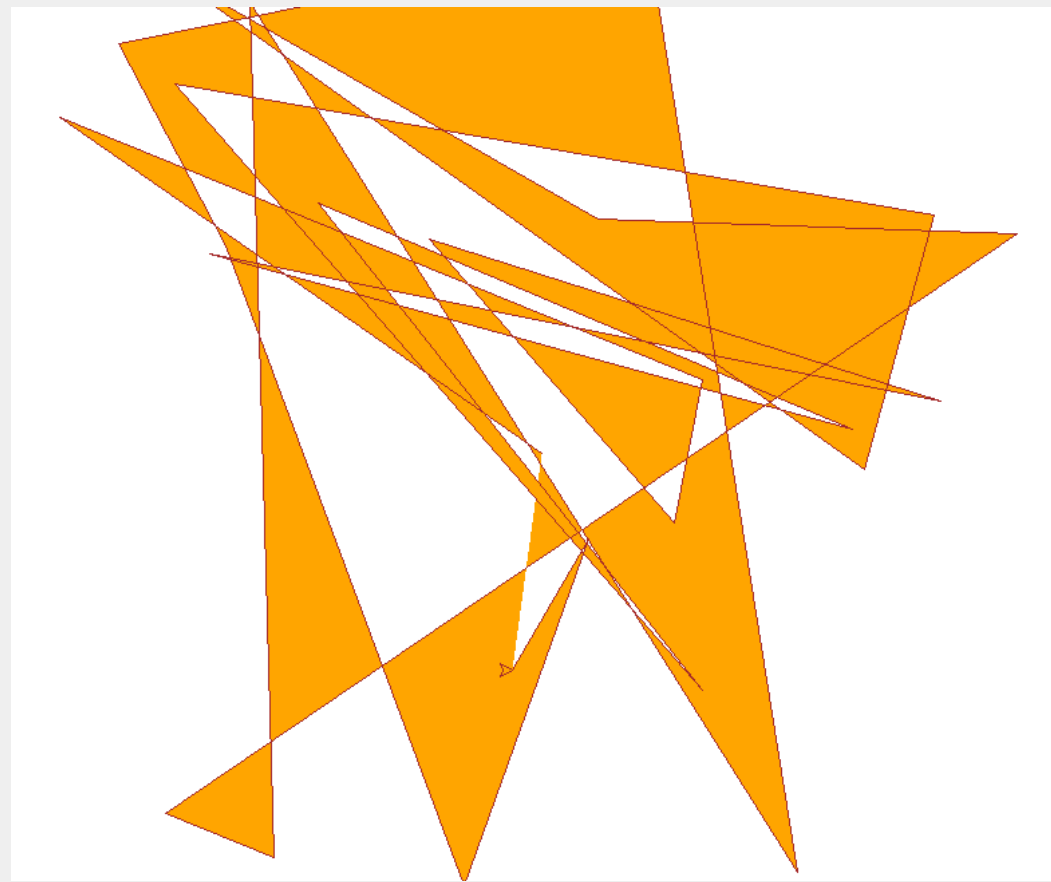


现代艺术画自动生成程序I – 利用turtle, random, 循环

```
import turtle
import random
NUM=25
XMAX=400
YMAX=400
colors=["grey","brown","orange","yellow","blue","red","yellow"]
rcolor=random.choices(colors,k=2)
turtle.color(rcolor[0],rcolor[1])
turtle.begin_fill()
for i in range(NUM):
    x=random.randrange(-XMAX,XMAX)
    y=random.randrange(-YMAX,YMAX)
    turtle.goto(x,y)
turtle.end_fill()
```

随机挑选颜色

随机位置



现代艺术画自动生成程序II – 利用turtle, random, 循环

程序文件的文件名为modernart.py
可以用IDLE打开并且运行

关键代码如下：

```
for i in range(NUM):
```

```
    p=turtle.Turtle()
```

```
    p.hideturtle()
```

```
    p.speed("fastest")
```

随机挑选颜色

```
    rcolor=random.choice(["grey","blue","yellow"])
```

```
    p.pen(pencolor=rcolor,pensize=3)
```

```
    x1=random.randrange(-XMAX,XMAX)
```

```
    y1=random.randrange(-YMAX,YMAX)
```

随机位置

```
    x2=random.randrange(-XMAX,XMAX)
```

```
    y2=random.randrange(-YMAX,YMAX)
```

