**Python计算机编程入门**

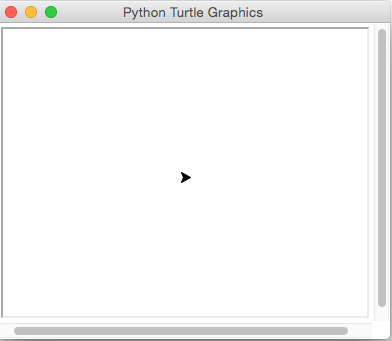
**\*\*\*\*\*本堂课知识点\*\*\*\*\***

* 海龟制图库
* 用海龟制图库画图形
* for循环
* 列表

**\*\*\*\*\*教学活动\*\*\*\*\***

* **复习 (5 min)**
  1. 变量的声明和使用
  2. 如何写一个Python程序文件
  3. 库：math
  4. 函数：print()和math.cos()
  5. 上节课我们讲到了如果需要更多的功能，python有很多库，除了math之外，我们这节课会用到一个新的库，turtle
* **海龟制图库 (35 min)**
  1. 首先我们需要新建一个python程序文件，下面的代码都在新的文件里面运行，保存为my\_turtle\_1.py
  2. 【导师笔记】**注意千万不要保存为turtle.py**，后面会具体讲到，简单给学生解释一下就是当我们import turtle的时候电脑会现在当前文件夹找turtle.py
  3. 【问题】引入turtle这个库该用什么python语句？
  4. 【答案】**import** turtle
  5. python的语句很简洁易懂，import就是引用的意思
  6. 首先我们需要一支画笔，在turtle里面有一个Pen()的函数可以生成一支画笔，具体原理我们后面会学到

|  |
| --- |
| **import** turtle  t = turtle.Pen() |

* 1. 注意，这两行代码之间最好有一行空行。写代码就像写文章一样，是分段落的。import是引入外来的库，下面的生成画笔就已经是在使用具体的函数了，这两句程序的作用很不一样，所以中间用空行隔开
  2. 我们会发现程序运行之后打开了一个新的白色界面，这就是我们的画布。画布上有一  
     个黑色的箭头这就是画笔，当前画笔是朝右的，这是画笔默认的方向。
  3. 我们现在关闭这个窗口。有些同学的窗口可能会卡死，这是因为画布一直在向我们的代码文件请求后续的代码而我们在t = turtle.Pen()之后其实已经没有代码了。所以它就会处于一边一直在发请求，一边一直没有回应的状况，因此程序发生请求超时，从而卡死，无法正常关闭。
  4. 我们需要告诉画布代码到这里已经结束了。在代码的最后加入

|  |
| --- |
| turtle.done() |

这里最好也空一行

* 1. 现在我们所需要的前期配置已经全部完成了，接下来我们就可以开始绘图了。当然我们肯定不可能一步登天一下子把第一节课看到的那种万花筒画出来，我们先从绘制直线开始。注意，下面的代码全都在t = turtle.Pen()和turtle.done()中间写

|  |
| --- |
| t = turtle.Pen()  *#CODE*  turtle.done() |

* 1. 移动画笔：我们想要绘制出一条直线，最直观的方法就是让画笔朝前移动一段距离。Turtle中就有一个这样的让画笔向前移动的函数。

|  |
| --- |
| t.forward(Number) |

* 1. 这个函数的意义是让名为t的画笔往笔头朝向的方向移动Number个单位长度（像素）。
  2. 注意这里.前面的不是turtle了，而是t，也就是我们刚才生成的画笔。这是因为forward是这只画笔的动作。类似于你从turtle文具店买了一只笔，现在拿起笔开始画画。
  3. 现在我们再次运行一下程序，我们看到笔已经向右运动了100个像素，同时留下了笔的运行轨迹：一条长度为100 的直线。
  4. 旋转画笔：控制画笔在当前位置向左侧旋转（逆时针方向）一定的角度

|  |
| --- |
| t.left(angle) |

* 1. 现在我们再次运行一下程序，我们看到笔已经转过了一定角度（注意箭头方向）
  2. 【练习】画一个正方形

|  |
| --- |
| **import** turtle  t = turtle.Pen()  t.forward(100) t.left(90) t.forward(100) t.left(90) t.forward(100) t.left(90) t.forward(100)  turtle.done() |

* 1. 注意这里面程序分4个段落：import外来的库，生成画笔，画画，结束
  2. 【练习】画一个三角形（注意，t.left(angle)不是三角形的内角大小，而是画笔逆时针旋转的角度大小）

|  |
| --- |
| **import** turtle  t = turtle.Pen()  t.forward(100) t.left(120) t.forward(100) t.left(120) t.forward(100)  turtle.done() |

* 1. 用循环画三角形:
  2. 为什么需要循环：在上个代码中我们为了写一个三角形重复了三遍一样的代码，那设想一下，如果我们要或一个一百边形，一千边形，代码是不是就有几百几千行了，这显然是不合理也是没有必要的，那么这边就有一个简化代码的方法，让电脑帮助我们重复地去做同一件事情那么这个过程的名字就是循环。  
     【举例】比如你们每天早上来到这边，上课，然后上完课晚上再回去休息，这几天重复地执行这一系列动作，那么这其实就是一个重复的过程，在电脑看来这就是一个循环。
  3. For循环：for循环是python循环中的一种。请打开**shell**：

|  |
| --- |
| **>>> for** i **in** range(5): |

* 1. For后面跟的是一个**循环变量**和一个**列表**
  2. 首先讲解range(5)。这个语句相当于生成了0-4这5个数字，他们组成了一个**列表**，**英文是list**，列表是python里面的另一种数据结构。在电脑看来，range(5)相当于[0,1,2,3,4]。注意电脑是从0开始数数的。
  3. 【导师笔记】如何区分程序员：让ta数5个数字，说01234而不是12345的就是程序员
  4. i是一个循环变量，每一轮循环的时候，i都会依次从range(5)生成的列表里面取值。
  5. 也就是说，第一轮循环的时候i的值是0
  6. 第二轮循环的时候i的值是1
  7. 【问题】最后一轮循环的时候，i的值是什么？
  8. 【答案】4，因为range(5)生成的列表是[0,1,2,3,4]
  9. 后面要加上一个冒号，代表从这里开始我们要开始写**循环体**，也就是需要循环执行的代码了
  10. 按下回车，会发现光标缩进了，这是因为结尾带有冒号的代码表示接下来我要说一个大段落的内容了，这些内容全都是需要重复的，在python里面就用缩进来把循环体和其他内容分开。
  11. 继续输入print(i)，然后按回车，第一次回车啥都不会发生，第二次回车之后会打印出0-5的数字

|  |
| --- |
| **>>> for** i **in** range(5):  print(i) |

* 1. 第一次回车之后，shell不确定你是不是输入完毕了，也许你还要输入多一行，但是当你再一次回车之后，对于shell来说，你的循环内容就写完了
  2. 上面的for循环相当于下面的代码：（导师一行行输入shell里面）

|  |
| --- |
| i = 0 print(i) i = 1 print(i) i = 2 print(i) i = 3 print(i) i = 4 print(i) |

* 1. 如果不希望从0开始，可以用range(a,b），得到的就是[a, b)里面的整数，注意最后是开区间，b是取不到的。
  2. 【导师笔记】在shell里面示范

|  |
| --- |
| **>>> for** i **in** range(1, 5):  print(i) |

* 1. 如果希望生成等差数列，可以用range(a,b,c)，则可以得到[a, b)的等差数列，第二项和第一项的差是c
  2. 【导师笔记】在shell里面示范：

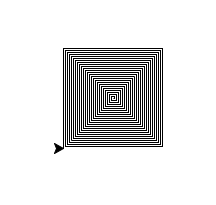
|  |
| --- |
| **>>> for** i **in** range(1, 10, 2):  print(i) |

* 1. 【练习】应用在turtle上，用for语句写一个画正方形的代码和一个画三角形的代码。

|  |
| --- |
| **import** turtle  t = turtle.Pen()  *# square* **for** i **in** range(4):  t.forward(100)  t.left(90)  turtle.done() |

|  |
| --- |
| **import** turtle  t = turtle.Pen()  *# triangle* **for** i **in** range(3):  t.forward(100)  t.left(120)  turtle.done() |

* 1. 【导师笔记】可以顺便讲一下#用来写注释
  2. 【导师笔记】请学生保存下画三角形的代码为triangle.py。接下来的所有练习请学生在新的文件里面编写。
  3. 【练习】如果让边长从1开始递增到100，也就是range(1,101)，每次旋转角度都是t.left(90)，会画出怎样的图形？



* 1. 【练习】用类似的方法画三角形，边长螺旋式地增大，每一条边比上一条长2像素，最短的边为1，画50条边（注意range的初始值和结束值）

|  |
| --- |
| **import** turtle  t = turtle.Pen()  **for** i **in** range(1,101,2):  t.forward(i)  t.left(120)  turtle.done() |