**Python计算机编程入门**

**\*\*\*\*\*本堂课知识点\*\*\*\*\***

* 海龟制图库：画笔颜色
* 列表（继续）
* 元组
* 字典
* 求余运算符

**\*\*\*\*\*教学活动\*\*\*\*\***

* **复习 (5 min)**
  1. 海龟制图入门
  2. 生成画笔
  3. 画直线和旋转画笔（三角形、正方形）
  4. for循环
  5. 列表
  6. 【导师笔记】提问：range(10)生成的是哪些数字？range(0,10)呢？range(1,10)呢？range(1,10,2)呢？注意range的初始值是闭区间，结束值是开区间。
* **列表 (10 min)**
  1. 【提问】刚才我们已经再形状上做足了文章，那么出了刚才讲过的那些，我们还可以在什么地方对这个图形进行美化呢？  
       
     颜色！
  2. 【导师笔记】请同学打开[triangle.py](https://drive.google.com/open?id=1H5m4x3yaX-7YYxJ-xTgrmg0PJR92E6Rf)，里面应该是以下代码：

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

* 1. 画笔颜色的添加

|  |
| --- |
| t.pencolor() |

作用是改变我们这支画笔的颜色

* 1. 系统自带颜色：系统为了方便我们去使用一些常用的颜色，对一些基础的颜色进行了储存，这样我们就能直接使用颜色的名字去调用这些颜色，比如“red”“yellow”
  2. 颜色单色填充代码

|  |
| --- |
| t.pencolor("red") |

这句代码加在t = turtle.Pen()后面，然后运行代码：画出来的所有东西都是红彤彤的

* 1. 多种颜色循环上色：如果需要多种颜色循环使用，我们需要用一个列表来储存我们想用到的所有颜色。
  2. 之前我们学习for语句的时候其实就已经用到了**列表，英文是list**，range(3)会生成一个包含0-1这3个数字的这一个列表，而每一次循环，循环变量i就会依次取里面的每一个值。
  3. 【导师笔记】如果有黑板或者白板，最好在上面把[0,1,2]写下来，数字之间间距稍微大一点，再用笔去比划i会依次变成0然后1然后2。也就是用笔依次去指这三个数字，指的时候说：第一轮循环，第二轮循环，第三轮循环
  4. 我们现在需要写一个类似的列表，这样在我们用来画三角形（或者正方形）的for循环里面，我们可以让画笔依次取里面的颜色。
  5. 【导师笔记】在[0,1,2]下面对应地写下[“red”, “blue”, “purple”]
  6. 注意这里的名字全都带引号，因为他们是字符串
  7. 请同学们打开shell，输入

|  |
| --- |
| colours = ["red", "blue", "purple"] |

* 1. 【导师笔记】我一般会先让学生在shell里面尝试一些代码，然后再在文件里面写。shell类似草稿纸
  2. 我们如何获取列表里面的某一个具体的值呢？请学生在shell输入以下代码

|  |
| --- |
| **>>>** colours[0]   'red' **>>>** colours[1]   'blue' **>>>** colours[2]   'purple' |

方括号里面的数字就是具体的index，index为0的数据是”red“

注意列表是从0开始编号的，就像range()如果只有一个数字的话，是默认从0开始取值

* 1. 我们也可以修改列表里面的元素：

|  |
| --- |
| **>>>** colours = ["red", "blue", "purple"]  **>>>** colours[0] = "violet" **>>>** colours  ['violet', 'blue', 'purple'] |

* 1. 再请学生输入colours[3]并按下回车

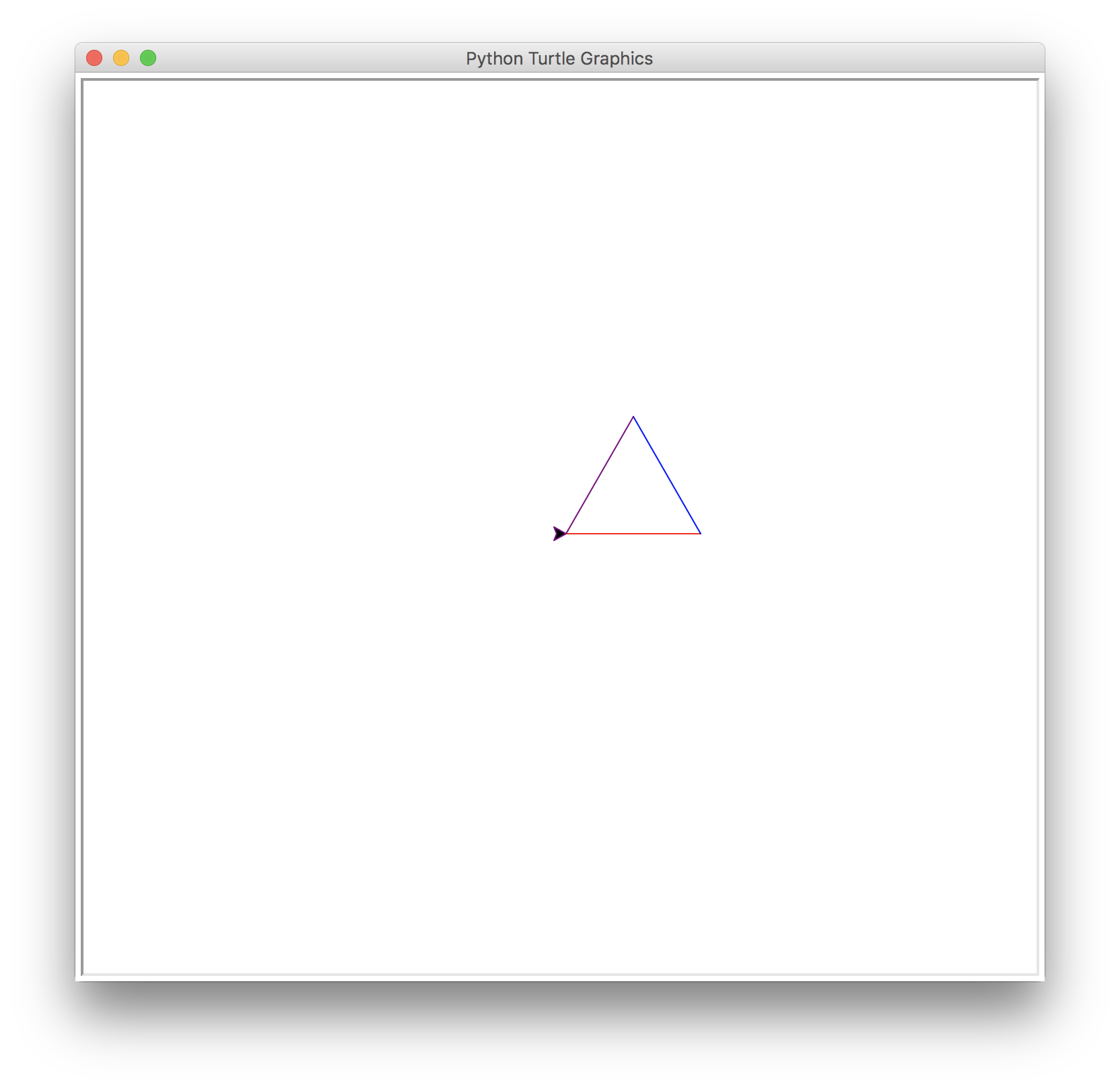
|  |
| --- |
| **>>>** colours[3]   Traceback (most recent call last):  File "<pyshell#67>", line 1, **in** <module>  colours[3] IndexError: list index out of range |

* 1. 这里我们遇到了一个新的Error，叫IndexError，这是很多人都容易犯的错误，具体的解释是list index out of range，意思是我们尝试读取list里面的第3个元素，虽然colours这个列表有3个元素，但是它们是从0开始编号的。
  2. 【导师笔记】可以让学生看黑板上写的相对应的两个列表[0,1,2]和[“red”, ”blue”, ”purple”]
  3. 【导师笔记】程序员冷笑话：程序员最烦恼的事情有两件：第0件，如何起名字（变量名之类的）；第1件，缓存失效（这个超纲了不解释，涉及到计算机底层系统）；第2件，index out of range (一般是off by 1，因为新手总会从1开始数数）
  4. 【导师笔记】这里也可以趁机复习一下之前遇到过的NameError（没有声明变量或者没有引入相应的库）还有SyntaxError（参数的个数不对）
  5. 回到[triangle.py](https://drive.google.com/open?id=1H5m4x3yaX-7YYxJ-xTgrmg0PJR92E6Rf)，把colours = ["red", "blue", "purple"]写在t = turtle.Pen()下面，在for循环里面加上t.pencolor(colours[i])，程序变成[triangle\_list.py](https://drive.google.com/open?id=1aZiEsDyoK0x9dFvIWG7P-lRYK0GpntPA)：

|  |
| --- |
| **import** turtle  t = turtle.Pen() colours = ["red", "blue", "purple"]  **for** i **in** range(3):  t.pencolor(colours[i])  t.forward(100)  t.left(120)  turtle.done() |

让学生先保存并运行一次

* 1. 效果如图：



* 1. 【导师笔记】问一下有没有同学理解了是怎么回事，让ta给其他同学讲解
  2. 回到黑板上的[0,1,2]，第一次循环的时候，i取值0，相当于：

|  |
| --- |
| t.pencolor(colours[0]) |

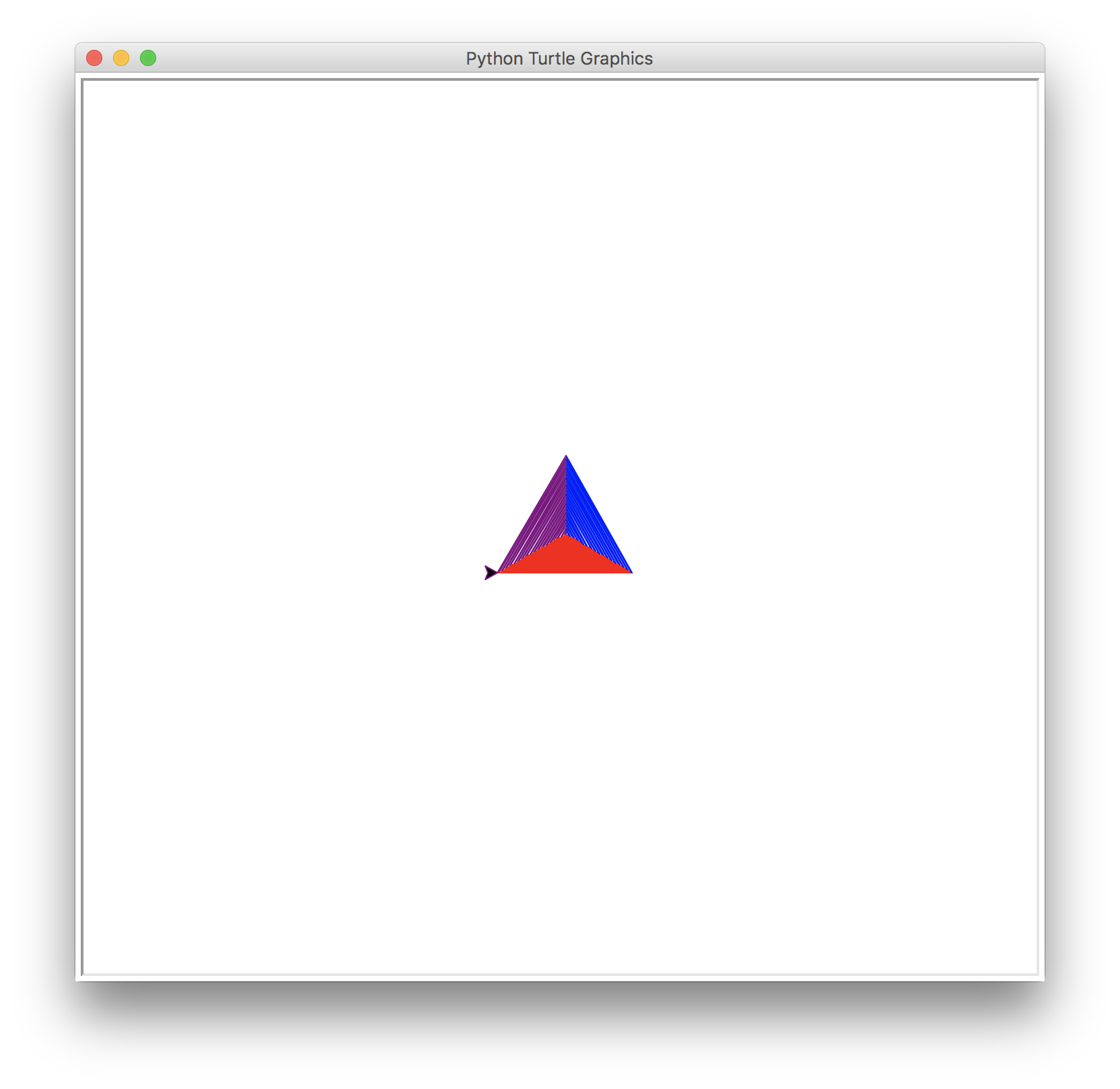
colours[0] = “red”，所以相当于：

|  |
| --- |
| t.pencolor("red") |

* 1. 同理可知蓝色和紫色
  2. 但是我们不满足于只画三条边，我们希望画102条边，颜色依次是红蓝紫红蓝紫红蓝紫，我们需要一个办法可以把大于2的循环变量的值转换到[0,2]的整数区间里。这时候我们可以用一个求余的运算[triangle\_mod.py](https://drive.google.com/open?id=1pPEK6Kvq6Y58eMweBjbSj22awvCZ1n6A)：

|  |
| --- |
| **import** turtle  t = turtle.Pen() colours = ["red", "blue", "purple"]  **for** i **in** range(102):  t.pencolor(colours[i % 3])  t.forward(i)  t.left(120)  turtle.done() |

* 1. 代码格式：注意这个程序分成四个段落：引入库，初始化画笔和颜色列表，画画，结束
  2. 代码格式：注意i % 3之间有空格，这样是为了让程序看起来不会太挤
  3. 效果如下：



* 1. 自定义颜色：如果不满足于系统带的颜色，我们也可以自定义。turtle支持英语单词，也支持6位16进制的颜色代码，比如可以尝试把“red”替换成"#33cc8c"
* **元组 (5 min)**
  1. **RGB颜色**：除此之外，turtle还支持我们输入RGB的值。RGB分别代表的是红Red，绿Green，蓝Blue，光的三原色。他们的取值是从0到255（2^8-1，因为用8位0或者1表示)
  2. 黑色就是(0, 0, 0)，没有光。白色就是(255, 255, 255)，红色是(255, 0, 0)，绿色是(0, 255, 0)，蓝色是(0, 0, 255)
  3. 但是在turtle里面，默认情况下，颜色的范围是0-1，所以我们要把所有的数除以255.0，比如255/255.0 100/255.0，注意被除数必须带小数点，不然就会认为是整数相除，结果永远都是0
  4. **元组**：因为RGB3个值是一起出现的，我们这里用小括号把他们括起来了，这种叫做元组，英文是tuple。它的作用有点类似列表，但是注意这里是小括号。
  5. 请打开shell，输入以下代码：

|  |
| --- |
| **>>>** red = (1, 0, 0) **>>>** red[0] 1 **>>>** red[1]  0 |

* 1. 元组里面元素的获取方式和列表是一样的，但是要注意的是，列表的数据是可以修改的，元组的则不可以：

|  |
| --- |
| **>>>** red[1] = 100   Traceback (most recent call last):  File "<pyshell#80>", line 1, **in** <module>  red[1] = 100 TypeError: 'tuple' object does **not** support item assignment |

* 1. 这里我们又遇到了一个TypeError，这里说的是tuple，也就是元组，这个玩意，不支持item assignment，也就是不允许我们修改里面的值。所以一旦定义了red = (1, 0, 0)就没办法小修小补了！想要修改就必须整个元组重新定义，回炉重造
  2. 回到triangle.py，pencolor支持以下几种语法：

|  |
| --- |
| t.pencolor((1, 0, 0)) *#这是一个元组作为参数，注意两层括号*t.pencolor((255/255.0, 0, 0)) *#注意被除数有小数点* t.pencolor(1, 0, 0) *#这是RGB三个数字作为参数（3个参数）* t.pencolor("red") *#这是字符串作为参数* t.pencolor("#FF0000") *#这是16进制颜色代码作为参数* |

* 1. 【练习】让同学修改colours列表，加上他们喜欢的颜色，继续画三角形，还可以画n边形
  2. 【导师笔记】这个网页挺有用：<http://tool.oschina.net/commons?type=3>
* **字典 (10 min)**
  1. 我们写了RGB的值之后，希望在程序里可以给这个颜色一个名字，这时候我们需要用到字典这种数据结构
  2. **字典，英文是dictionary**，就像一本字典一样，有索引词，根据索引词可以瞬间找到我们想要的数据
  3. 请学生打开shell，输入以下代码，定义colours字典：

|  |
| --- |
| **>>>** colours = {"SkyBlue": "#87CEEB", "Gold": "#FFD700", "Wheat": "#F5DEB3"} |

* 1. 这本字典里面目前有两个词条，用逗号分隔开，每一个词条里面，冒号前的是key，也就是索引词，冒号后面的是value，是具体的数据
  2. 从字典获取值的方法和列表元组都不一样了，输入以下代码 ：

|  |
| --- |
| **>>>** colours[0]   Traceback (most recent call last):  File "<pyshell#82>", line 1, **in** <module>  colours[0] KeyError: 0 |

* 1. 这里我们见到了一种新的错误，KeyError，这是因为字典里面的取值是通过key也就是索引词来实现的，这个错误是说我们字典里面没有key叫做0
  2. 以下代码可以获取具体的值（value）：

|  |
| --- |
| **>>>** colours["Gold"]  "#FFD700" |

* 1. 但是我们也可以用index来获取某一个具体的key:

|  |
| --- |
| >>> keys = list(colours.keys())  >>> keys  ['SkyBlue', 'Gold', 'Wheat'] |

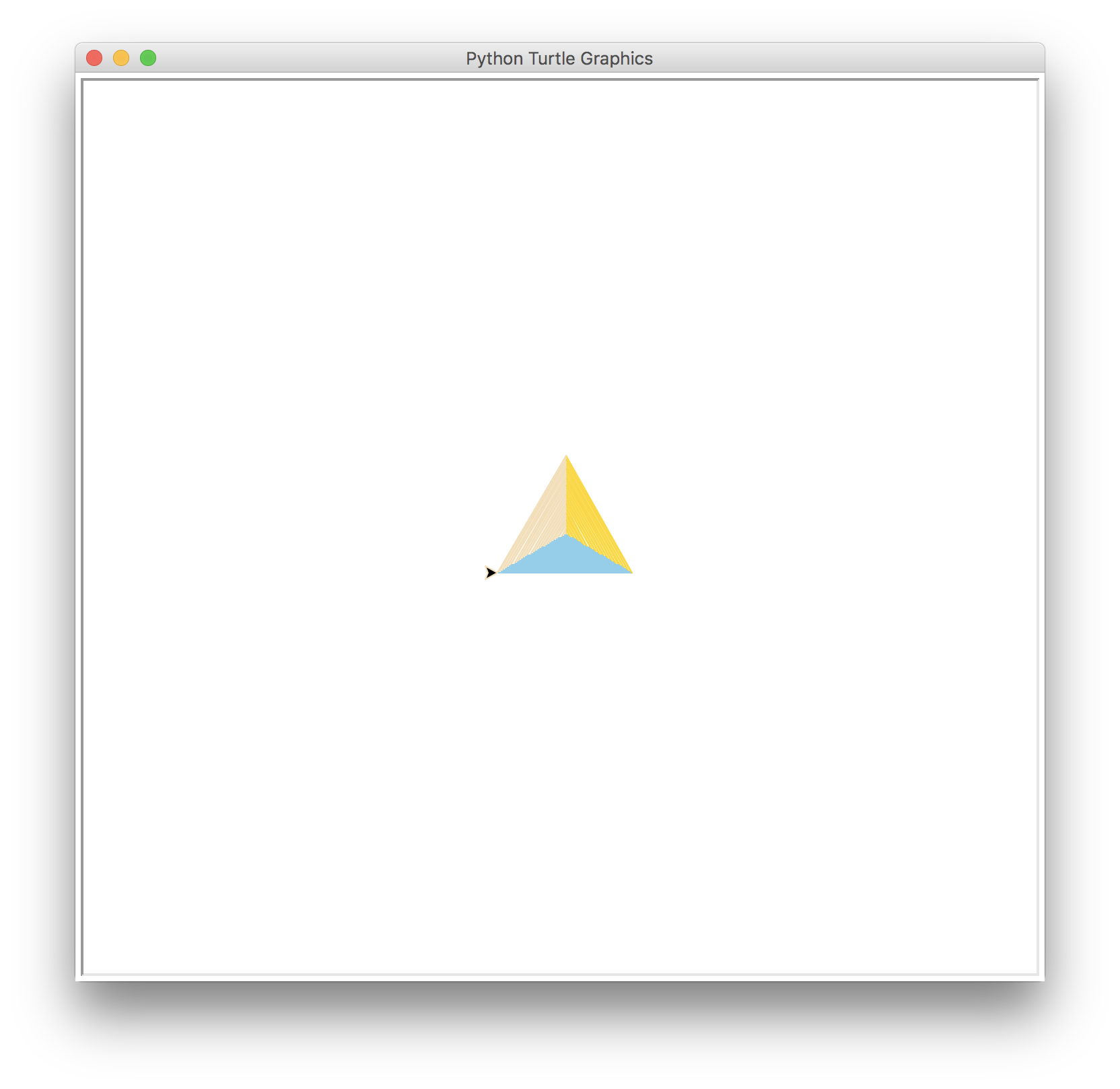
* 1. list()函数把colours字典的keys转化成了一个列表：

|  |
| --- |
| **>>>** keys[0]  'SkyBlue' |

* 1. 需要获取colours里面的值就得colours[keys[0]]
  2. 程序变成这样，[triangle\_dict.py](https://drive.google.com/open?id=1GmynB1H0iQ77bEmhdClLeBmzGzfgRnHo)：

|  |
| --- |
| **import** turtle  t = turtle.Pen() colours = {"SkyBlue": "#87CEEB", "Gold": "#FFD700", "Wheat": "#F5DEB3"} keys = list(colours.keys())  **for** i **in** range(102):  t.pencolor(colours[keys[i % 3]])  t.forward(i)  t.left(120)  turtle.done() |

* 1. 效果如图：



* 1. 【练习】建一个有至少5种颜色的字典，每个颜色用元组或者字符串表示。用不同的颜色画100个边长逐渐变长的正5边形或者正6边形。