



高级语言程序设计 Python

教 案

任课教师： 刘冰

学期：2023-2024 第 2 学期

第 1 课 (4 学时)	1
第 2 课 (4 学时)	6
第 3 课 (4 学时)	13
第 4 课 (4 学时)	17
第 5 课 (4 学时)	20
第 6 课 (4 学时)	24
第 7 课 (4 学时)	28
第 8 课 (4 学时)	31
第 9 课 (4 学时)	43
第 10 课 (4 学时)	50
第 11 课 (4 学时)	55
第 12 课 (4 学时)	61
第 13 课 (4 学时)	65
第 14 课 (4 学时)	69

第 1 课（4 学时）

教材章节：

第 1 章 Python 语言概述

教学目的和要求：

- 1、了解程序设计语言的基本概念
- 2、熟悉 Python 开发环境的下载和安装
- 3、熟悉简单的 Python 程序的开发过程

教学重点

- 1、集成开发环境的下载和安装。
- 2、Python 程序设计的一般步骤。
- 3、简单程序的开发过程。

教学难点

- 1、Python 程序的保存、装入和运行。
- 2、Python 语言的基本语法。

教学方法与手段

使用超星学习通线上直播授课，理论与实践相结合，qq 群辅助教学。

教学内容和教学过程

实践课：

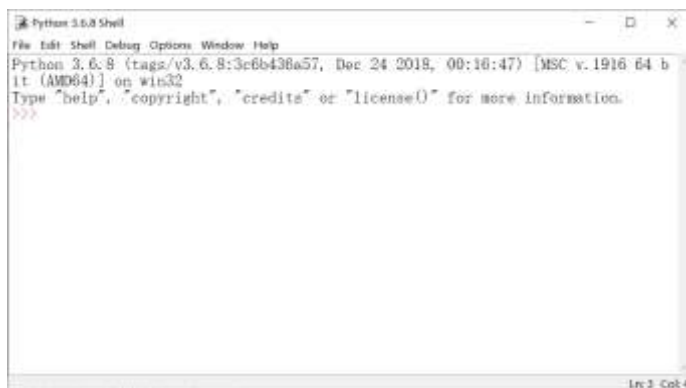
第一课 课程介绍和认识 python

1, 2 节 实践课

一 安排座位

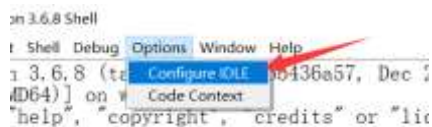
二 python 语言的 idle 编程环境及简单命令。

1 启动 idle(单击开始按钮—所有程序—python3.x—Idle)，出现下方编辑器界面。



****IDLE: Integrated Development and Learning Environment（集成开发与学习环境）即程序编辑器。**

默认编辑器的字号比较小，可以通过设置编辑器中的字体字号等，以方便观看。在 Idle 中，单击“Options—Configure Idle”，出现设置 Idle 使用环境窗口





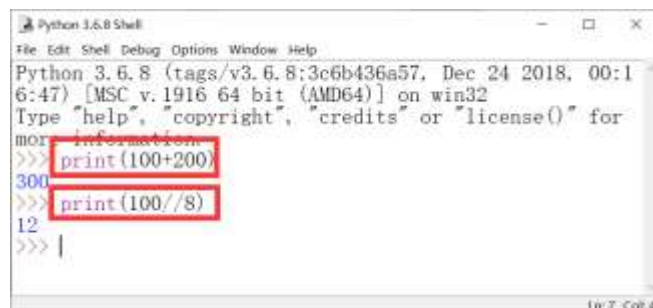
2 在交互环境学习数学运算的几个符号

1) $+$ $-$ $*$ $/$ $\%$ (求余数) $//$ (求商或整除) $**$ (幂运算)
在提示符`>>>`输入运算式，回车确定。会在下方出现运算结果，如下图所示。



2) print 的使用

****注意：**在 Idle 中，输入一个计算式是可以直接看到运算结果的，因为 Idle 可以解析你的运算。但实际上是省略了 `print()` 函数的，完整的表达应该是：
`print(运算式)`，例如：



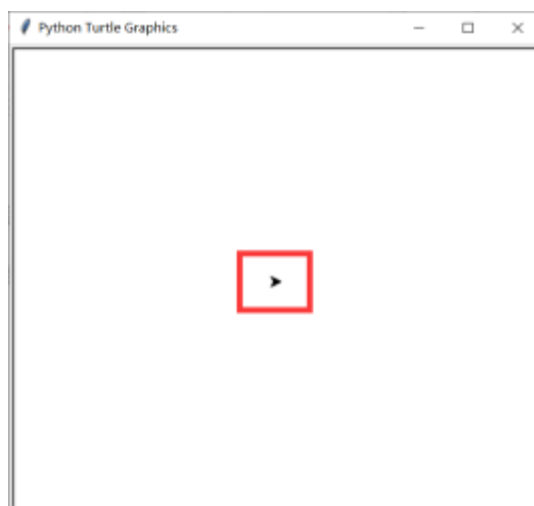
有时，需要将数字赋予一个变量，以便让它多次运算。比如
`>>>r = 5` #本行没有运行结果，在 `print(xxx)` 时才会看到
`>>>pi = 3.14159`

```
>>>print(r*r*pi)
>>>print(2*r*pi)
>>>print('Hello World!')    #需要输出一句话时，使用英文的单引号或双引号括住输出内容。
>>>print("世界，你好！")
>>>sum = 99999 * 99999
>>>print(sum)
```

****小技巧：**可以将光标放在 Idle 中执行过的命令行上，回车即可将该命令放在最新的待执行位置，可以修改命令或者直接回车再次执行。

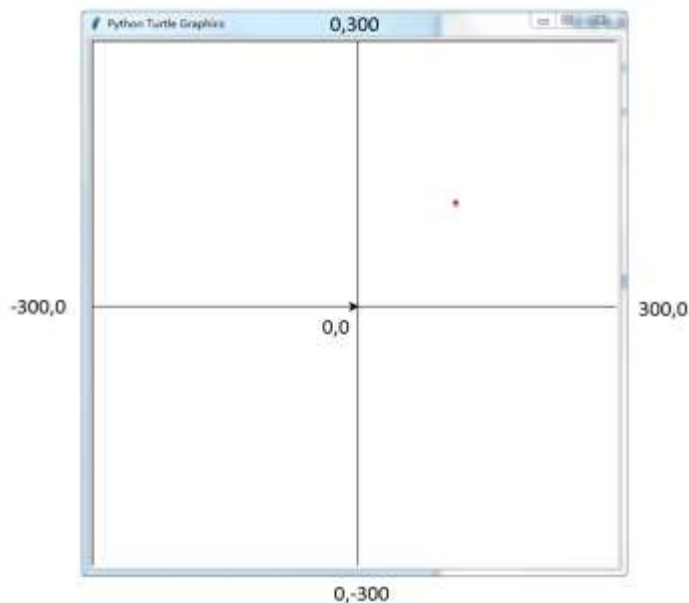
三 实验三 turtle 库的使用（实验教材 P13）

turtle 是 Python 重要的内置库之一，能够进行基本的图形绘制。简单来讲是在一个如下图的画布上，用 Python 命令模拟一个小海龟（注意到下图中间的箭头了么，好吧，这就是海龟）在爬行，海龟爬行的轨迹就形成了绘制的图形。



1 简单直线绘图

- 1) 导入海龟绘图库 `import turtle`。
 - 2) 控制海龟画笔行进命令 `turtle.forward(距离)`, `turtle.backward(距离)`
 - 3) 相对角度旋转命令 `turtle.left(角度)`和 `turtle.right(角度)`
 - 4) 撤销一步命令 `turtle.undo()`、清屏命令 `turtle.clear()`、重置海龟画布命令 `turtle.reset()`
 - 5) **课堂练习 1：请画出一个边长 100 的正方形、六边形(旋转 60)、和五角星(旋转 144)。**
- **绘制不连续图形的时候，需要使用的命令 `turtle.penup()` 和 `turtle.pendown()` 和 `turtle.goto(x,y)`命令。x,y 为坐标数字，与平面直角坐标系一致，参见下图。**



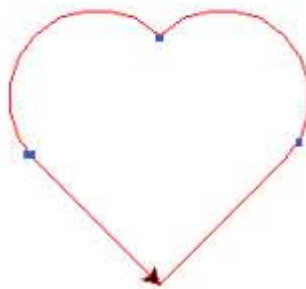
- 6) 指针回到原点 `turtle.home()`、绝对旋转角度 `turtle.setheading(角度)`。`setheading` 为绝对旋转，不受箭头原先方向影响。

2 课堂小作业

- 1) 设置画布大小命令 `turtle.setup(窗口宽度, 窗口高度)`
- 2) 设置背景色命令 `turtle.bgcolor("英文颜色单词")`，画笔颜色命令 `turtle.pencolor("英文颜色单词")`
- 3) **课堂练习 2: 尝试在画布上用线条命令绘制“中国”两个字(需要简单的在平面直角坐标系计算坐标知识)。**

3 画圆命令

- 4) `turtle.circle(半径)` 半径正数，圆在海龟的左侧，半径负数，圆在海龟的右侧
 - 5) `turtle.circle(半径, 角度)` 画弧线，也就是圆的一部分。
 - 6) `turtle.circle(半径, steps=n)` 画内切于圆的正 n 边形，半径也可以为正或负。
 - 7) **课堂练习 3: 做出如下图所示心形图案。**
- **提示: 分为 4 笔，线段、半圆、半圆、线段。**



4 笔画宽度和填充颜色

- 1 画笔默认宽度是 1，可以通过 `turtle.pensize(数字)` 命令，使画笔加宽。
- 2 当需要做一个带有颜色的图形块时，
 - 1) 绘制图形前先用 `turtle.fillcolor(颜色单词)` 命令设置填充颜色
 - 2) 在画图前使用 `turtle.begin_fill()` 命令标记
 - 3) 在画图结束后使用 `turtle.end_fill()` 命令进行填充颜色

3 红色心形的绘制。



课堂作业，3月12日是我国的植树节，请利用以上学过的 turtle 命令，绘制1颗树。并提交学习通作业。

理论课：

详见讲义第一、二章.pptx

思考与练习：

观看学习通本节相关视频，下载安装 python，做好国内库下载镜像

阅读文献：

《Python 语言程序设计》 第1章 第2章

教学后记：

第 2 课（4 学时）

教材章节：

第三方库函数 Turtle

教学目的和要求：

导入 turtle 库的 3 种方法

turtle 库常用函数的使用

教学重点

turtle 库常用函数

导入 turtle 库的 3 种方法

教学难点

turtle 库常用函数。

教学方法与手段

使用超星学习通线上直播授课，理论与实践相结合，qq 群辅助教学。

教学内容和教学过程

实践课：

实验目的：了解循环概念并做练习
掌握填充颜色的方法


第二课 turtle 库实验 2

实践课

实验目的：了解循环概念并做练习
掌握填充颜色的方法

在做上周作业时，很多同学使用的命令方式进行画树，如果某个语句错误，进行修改就比较麻烦，应该使用程序方式进行绘制。请进行以下实验：

实验一 实验教材 P15(4) 利用循环语句绘制一个边长为 200 的黄色五角星。

请启动 idle(开始菜单— IDLE (Python 3.4)), 并使用“file—new file”命令建立一个新的程序文件，将下面代码复制到新的程序中并补充问号处填写的内容，运行程序看到效果图。

```
import turtle
turtle. —? — ("red")    #将画布改为红色
turtle. —? — ("yellow")  #将画笔改为黄色
for i in range(—? —):
    turtle.forward(200)
    turtle.left(—? —)
```

实验二 在 idle 中新建程序文件，将 P16 例题（2）键入保存为 e31.py，并运行（多用复制粘贴 快捷键 ctrl+C ctrl+V 很有用）。

补充一点 turtle 命令，可在本试验中尝试使用。

- 绝对旋转角度命令 setheading(seth)

海龟旋转角度命令除了 left 和 right，还有 setheading 命令，简写为 seth，称为绝

对旋转。Left 和 right 是以当前箭头指向为初始值，向左或右进行旋转。比如当前指向正上方 ▲，left(45)后，箭头会指向左上方 ▼。

seth 以正右方（x 轴正方向）为 0 度，跟下达命令时海龟的当前朝向无关。比如当前指向正上方 ▲，left(45)后，箭头会按照正右方向左旋转 45 度，箭头指向右上方 ▼。

- 其它命令

- 1 shape() — 括号中可设为“arrow”、“turtle”、“circle”、“square”、“triangle”、“classic”，使箭头显示为不同类型。

- 2 shapesize(数字) — 可以设置指针大小，默认为 1，使用 2,3...适当放大。

- 3 showturtle() 和 hideturtle() — 显示或者隐藏箭头，一般画完一个图像后，可以使用隐藏箭头命令后截图。

- 4 color(笔颜色, 填充颜色) — 可以同时设置画笔和填充的颜色。

实验三 p17 实验 3 绘制彩色同心圆程序（重点）需要理解的知识点：

- ✓ 什么是列表；
- ✓ 列表中项目的选取；
- ✓ 循环中 range() 函数的范围；
- ✓ 循环中 i 的取值变化。

1 列表是 Python 的一种重要数据类型，是用中括号[]括起来的若干数据的序列。如 [1,2,3,4,5] 、 ['a' , ' b' , ' c' , ' d'] 等，第四章将详细学习。

2 选择列表中的某一个数据，可以用序号方式，又称索引方式。例如

```
>>> x=["a","b","c","d","e"]
>>> x[0]
'a'
>>> x[4]
'e'
```

以上命令为建立一个列表类型的变量 x，然后用索引号取里列表中的某一个数据。注意，列表索引是从 0 号开始排列。

教材中 c=["red","yellow","blue","green","purple"] 这样语句，如果想取到蓝色，那么应该为 c[??]

但是使用列表有个问题，就是索引号不能超过列表中最大的编号，比如上例中，就不用 c[5]，语句会出错。

3 for i in range(10): 这样语句代表着循环 10 次，但是在循环的过程中，变量 i 的值不是一成不变的，而是从 0~9 在变化。例如：

```
>>> for i in range(10):
        i
结果会是 0~9
```

你明白以下代码的意思么：

```
import turtle as t
c=["red","yellow","blue","green","purple"]
for i in range(5):
    t.pencolor(c[i])
```

```
t.circle(100)
```

重点：比如 i 的取值为 0—9， $i\%5$ 的取值序列为：

i	0	1	2	3	4	5	6	7	8	9	...
$i\%5$	0	1	2	3	4	0	1	2	3	4	...

4 画同心圆

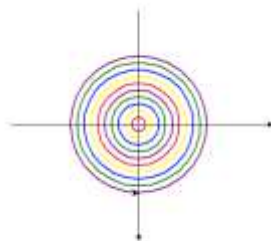
首先画不同心的圆。

```
import turtle as t
for i in range(5):
    t.circle(100+10*i)
```

以上代码是以坐标系原点为起点向上（左）画圆，圆的半径分别为

i 值	0	1	2	3	4
半径 $100+10*i$	100	110	120	130	140

如果每次画圆前，都将起笔要向下移动 10 个单位（在 y 轴上依次向下移动），那么画的就是同心圆了。

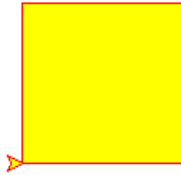


i 值	0	1	2	3	4
画圆的起点 $\text{goto}(0, -10*i)$	0, 0	0, -10	0, -20	0, -30	0, -40

实验四 turtle 中颜色的填充

在 turtle 中，画布颜色 `bgcolor` 和笔的颜色之外，还有一个非常重的跟颜色相关的功能——填充颜色。跟填充有关的命令有三个 1 设置填充的颜色 `fillcolor`（“英文颜色”）；2 开始填充点 `begin_fill()` 和完成填充点 `end_fill()`

使用顺序一般是先设置填充色，再定义起始填充点——绘图——完成填充。例如：



```
>>> t.pencolor("red")
>>> t.fillcolor("yellow")
>>> t.begin_fill()
>>> t.fd(100)
>>> t.left(90)
>>> t.fd(100)
>>> t.left(90)
>>> t.fd(100)
>>> t.left(90)
>>> t.fd(100)
>>> t.left(90)
>>> t.end_fill()
```

5 课堂小练习



['snow', 'ghost white', 'white smoke', 'gainsboro', 'floral white', 'old lace',
 'linen', 'antique white', 'papaya whip', 'blanched almond', 'bisque', 'peach
 puff',
 'navajo white', 'lemon chiffon', 'mint cream', 'azure', 'alice blue',
 'lavender',
 'lavender blush', 'misty rose', 'dark slate gray', 'dim gray', 'slate gray',
 'light slate gray', 'gray', 'light grey', 'midnight blue', 'navy', 'cornflower
 blue', 'dark slate blue',
 'slate blue', 'medium slate blue', 'light slate blue', 'medium blue', 'royal
 blue', 'blue',
 'dodger blue', 'deep sky blue', 'sky blue', 'light sky blue', 'steel blue',
 'light steel blue',
 'light blue', 'powder blue', 'pale turquoise', 'dark turquoise', 'medium
 turquoise', 'turquoise',
 'cyan', 'light cyan', 'cadet blue', 'medium aquamarine', 'aquamarine', 'dark
 green', 'dark olive green',
 'dark sea green', 'sea green', 'medium sea green', 'light sea green', 'pale
 green', 'spring green',
 'lawn green', 'medium spring green', 'green yellow', 'lime green', 'yellow
 green',
 'forest green', 'olive drab', 'dark khaki', 'khaki', 'pale goldenrod', 'light
 goldenrod yellow',
 'light yellow', 'yellow', 'gold', 'light goldenrod', 'goldenrod', 'dark
 goldenrod', 'rosy brown',
 'indian red', 'saddle brown', 'sandy brown',
 'dark salmon', 'salmon', 'light salmon', 'orange', 'dark orange',
 'coral', 'light coral', 'tomato', 'orange red', 'red', 'hot pink', 'deep pink',
 'pink', 'light pink',
 'pale violet red', 'maroon', 'medium violet red', 'violet red',
 'medium orchid', 'dark orchid', 'dark violet', 'blue violet', 'purple', 'medium
 purple',
 'thistle', 'snow2', 'snow3',
 'snow4', 'seashell2', 'seashell3', 'seashell4', 'AntiqueWhite1',
 'AntiqueWhite2',
 'AntiqueWhite3', 'AntiqueWhite4', 'bisque2', 'bisque3', 'bisque4', 'PeachPuff2',
 'PeachPuff3', 'PeachPuff4', 'NavajoWhite2', 'NavajoWhite3', 'NavajoWhite4',
 'LemonChiffon2', 'LemonChiffon3', 'LemonChiffon4', 'cornsilk2', 'cornsilk3',
 'cornsilk4', 'ivory2', 'ivory3', 'ivory4', 'honeydew2', 'honeydew3',
 'honeydew4',
 'LavenderBlush2', 'LavenderBlush3', 'LavenderBlush4', 'MistyRose2',
 'MistyRose3',
 'MistyRose4', 'azure2', 'azure3', 'azure4', 'SlateBlue1', 'SlateBlue2',
 'SlateBlue3',
 'SlateBlue4', 'RoyalBlue1', 'RoyalBlue2', 'RoyalBlue3', 'RoyalBlue4', 'blue2',
 'blue4',

'DodgerBlue2', 'DodgerBlue3', 'DodgerBlue4', 'SteelBlue1', 'SteelBlue2',
 'SteelBlue3', 'SteelBlue4', 'DeepSkyBlue2', 'DeepSkyBlue3', 'DeepSkyBlue4',
 'SkyBlue1', 'SkyBlue2', 'SkyBlue3', 'SkyBlue4', 'LightSkyBlue1',
 'LightSkyBlue2',
 'LightSkyBlue3', 'LightSkyBlue4', 'SlateGray1', 'SlateGray2', 'SlateGray3',
 'SlateGray4', 'LightSteelBlue1', 'LightSteelBlue2', 'LightSteelBlue3',
 'LightSteelBlue4', 'LightBlue1', 'LightBlue2', 'LightBlue3', 'LightBlue4',
 'LightCyan2', 'LightCyan3', 'LightCyan4', 'PaleTurquoise1', 'PaleTurquoise2',
 'PaleTurquoise3', 'PaleTurquoise4', 'CadetBlue1', 'CadetBlue2', 'CadetBlue3',
 'CadetBlue4', 'turquoise1', 'turquoise2', 'turquoise3', 'turquoise4', 'cyan2',
 'cyan3',
 'cyan4', 'DarkSlateGray1', 'DarkSlateGray2', 'DarkSlateGray3', 'DarkSlateGray4',
 'aquamarine2', 'aquamarine4', 'DarkSeaGreen1', 'DarkSeaGreen2', 'DarkSeaGreen3',
 'DarkSeaGreen4', 'SeaGreen1', 'SeaGreen2', 'SeaGreen3', 'PaleGreen1',
 'PaleGreen2',
 'PaleGreen3', 'PaleGreen4', 'SpringGreen2', 'SpringGreen3', 'SpringGreen4',
 'green2', 'green3', 'green4', 'chartreuse2', 'chartreuse3', 'chartreuse4',
 'OliveDrab1', 'OliveDrab2', 'OliveDrab4', 'DarkOliveGreen1', 'DarkOliveGreen2',
 'DarkOliveGreen3', 'DarkOliveGreen4', 'khaki1', 'khaki2', 'khaki3', 'khaki4',
 'LightGoldenrod1', 'LightGoldenrod2', 'LightGoldenrod3', 'LightGoldenrod4',
 'LightYellow2', 'LightYellow3', 'LightYellow4', 'yellow2', 'yellow3', 'yellow4',
 'gold2', 'gold3', 'gold4', 'goldenrod1', 'goldenrod2', 'goldenrod3',
 'goldenrod4',
 'DarkGoldenrod1', 'DarkGoldenrod2', 'DarkGoldenrod3', 'DarkGoldenrod4',
 'RosyBrown1', 'RosyBrown2', 'RosyBrown3', 'RosyBrown4', 'IndianRed1',
 'IndianRed2',
 'IndianRed3', 'IndianRed4', 'sienna1', 'sienna2', 'sienna3', 'sienna4',
 'burlywood1',
 'burlywood2', 'burlywood3', 'burlywood4', 'wheat1', 'wheat2', 'wheat3',
 'wheat4', 'tan1',
 'tan2', 'tan4', 'chocolate1', 'chocolate2', 'chocolate3', 'firebrick1',
 'firebrick2',
 'firebrick3', 'firebrick4', 'brown1', 'brown2', 'brown3', 'brown4', 'salmon1',
 'salmon2',
 'salmon3', 'salmon4', 'LightSalmon2', 'LightSalmon3', 'LightSalmon4', 'orange2',
 'orange3', 'orange4', 'DarkOrange1', 'DarkOrange2', 'DarkOrange3',
 'DarkOrange4',
 'coral1', 'coral2', 'coral3', 'coral4', 'tomato2', 'tomato3', 'tomato4',
 'OrangeRed2',
 'OrangeRed3', 'OrangeRed4', 'red2', 'red3', 'red4', 'DeepPink2', 'DeepPink3',
 'DeepPink4',
 'HotPink1', 'HotPink2', 'HotPink3', 'HotPink4', 'pink1', 'pink2', 'pink3',
 'pink4',
 'LightPink1', 'LightPink2', 'LightPink3', 'LightPink4', 'PaleVioletRed1',
 'PaleVioletRed2', 'PaleVioletRed3', 'PaleVioletRed4', 'maroon1', 'maroon2',

```

'maroon3', 'maroon4', 'VioletRed1', 'VioletRed2', 'VioletRed3', 'VioletRed4',
'magenta2', 'magenta3', 'magenta4', 'orchid1', 'orchid2', 'orchid3', 'orchid4',
'plum1',
'plum2', 'plum3', 'plum4', 'MediumOrchid1', 'MediumOrchid2', 'MediumOrchid3',
'MediumOrchid4', 'DarkOrchid1', 'DarkOrchid2', 'DarkOrchid3', 'DarkOrchid4',
'purple1', 'purple2', 'purple3', 'purple4', 'MediumPurple1', 'MediumPurple2',
'MediumPurple3', 'MediumPurple4', 'thistle1', 'thistle2', 'thistle3',
'thistle4',
'gray1', 'gray2', 'gray3', 'gray4', 'gray5', 'gray6', 'gray7', 'gray8', 'gray9',
'gray10',
'gray11', 'gray12', 'gray13', 'gray14', 'gray15', 'gray16', 'gray17', 'gray18',
'gray19',
'gray20', 'gray21', 'gray22', 'gray23', 'gray24', 'gray25', 'gray26', 'gray27',
'gray28',
'gray29', 'gray30', 'gray31', 'gray32', 'gray33', 'gray34', 'gray35', 'gray36',
'gray37',
'gray38', 'gray39', 'gray40', 'gray42', 'gray43', 'gray44', 'gray45', 'gray46',
'gray47',
'gray48', 'gray49', 'gray50', 'gray51', 'gray52', 'gray53', 'gray54', 'gray55',
'gray56',
'gray57', 'gray58', 'gray59', 'gray60', 'gray61', 'gray62', 'gray63', 'gray64',
'gray65',
'gray66', 'gray67', 'gray68', 'gray69', 'gray70', 'gray71', 'gray72', 'gray73',
'gray74',
'gray75', 'gray76', 'gray77', 'gray78', 'gray79', 'gray80', 'gray81', 'gray82',
'gray83',
'gray84', 'gray85', 'gray86', 'gray87', 'gray88', 'gray89', 'gray90', 'gray91',
'gray92',
'gray93', 'gray94', 'gray95', 'gray97', 'gray98', 'gray99']

```

理论课:

详见讲义

阅读文献:

《Python 语言程序设计》 第 2 章 2.5 节

练习与思考:

修改本节中的实例，绘制 2020 如何添加代码？

熟悉 python 的 turtle 库中可以使用的颜色常量

教学反思:

第3课（4学时）

教材章节：

2.5turtle 库的使用

第3章 基本数据类型与函数

教学目的和要求：turtle 库常用函数的使用

理解程序设计的方法 IPO

教学重点和难点

IPO 程序设计方法

教学方法与手段

使用超星学习通线上直播授课，理论与实践相结合，qq 群辅助教学。

教学内容和教学过程

实践课：

第三课 turtle 库实验 3

实践课

实验目的：掌握循环概念并做练习
程序的 IPO 结构

一 作业讲解 1——填充效果同心圆

```
for i in range(10, 0, -1)
```

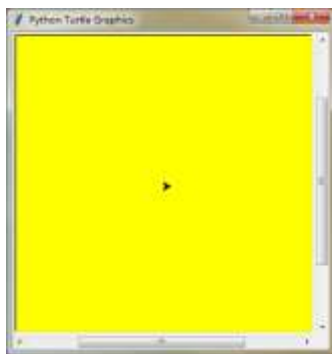
含义是用 range 构造一个数字范围，从 10 开始到 0 结束，每次加上-1，也就是减去 1。也就是 [10, 9, 8, 7, 6, 5, 4, 3, 2, 1]

i 从前到后每次取 1 个数字，然后用于循环中的语句计算（带有缩进的部分语句）。

作业讲解 2——绘制彩色图案（实验教材 P18(2)）

需要掌握的知识点

- Random 库中的随机数函数——choice、randint
- Setup() 和 screensize(): 注意看下图，整个窗口是由 setup 确定的大小，包括各种边框，标题栏等。Screensize() 设置的是画布大小，就是中间黄色的部分，screensize 可以在设置画布大小的时候同时设置背景色，如教材所述。画布大于窗口，则在窗口添加滚动条；画布不能小于窗口，如画布小于窗口，就用窗口内实际区域表示画布。



二 海龟 IPO 结构（I 指输入数据，P 指处理数据，O 指输出数据）

我们学的大多数程序，一般是需要在程序运行时输入一些数据，然后由程序对这写数据进行处理，然后再输出程序运行结果。这就是 IPO 结构。上两节课我们没有对运行中的程序输入数据，通过本例来体会一下。

请大家键入以下程序（老师很无耻的用了截屏, 没有放代码，因为输入的过程也是熟悉的过程）。红色注释不用输入。

```
turtle库的ipo.py - E:\老数据\2020python\第3周\turtle库的ipo.py (3.6.8)
File Edit Format Run Options Window Help
from turtle import *
setup(800,600)
bgcolor("lightgreen")
hideturtle()
for i in range(1000000): #尚未讲解while, 就用for吧
    n=numinput("输入","1 圆\n2 正方形\n3 正5边形\n0 结束")
    if n==1:
        clear()
        home()
        circle(100)
    elif n==2:
        clear()
        home()
        circle(100, steps=4)
    elif n==3:
        clear()
        home()
        circle(100, steps=5)
    elif n==0:
        bye()
        break
    else:
        write("没有这个选项", font=(32))
```

大家先观察运行结果，然后老师讲解。

turtle.numinput——海龟中的数字输入函数，可以为运行中的海龟提供数字。

turtle.textinput——海龟中的文字输入函数，可以为运行中的海龟提供文字。

turtle.bye()——结束海龟的运行

三 程序设计的 IPO 结构

本节实验做一个求解一元二次方程根的程序，先不做教材上的个税计算器。

1 新建一个程序文件，输入以下程序，注意红色代码为注释，可以不用输入。3 单引号中的绿色内容也是注释，不用输入。

```
一元二次方程.py - C:/Users/Administrator/AppData/Local/Programs/Python/Python36-32/
File Edit Format Run Options Window Help
a=eval(input('输入a: ')) # 输入二次项系数
b=eval(input('输入b: ')) # 输入一次项系数
c=eval(input('输入c: ')) # 输入常数项
'''以上为程序的输入部分
作为 a*x*x + b*x + c=0 的系数部分
'''
d= b**2 -4*a*c #d为△
if d>=0:
    x1=(-b+d**0.5)/(2*a)
    x2=(-b-d**0.5)/(2*a)
    print(x1,x2)
else:
    print("输入系数无解")

##以上程序为一个标准的Ipo结构程序
##什么是ipo呢，就是输入数据-处理-输出数据
##设置程序中的某行内容为注释，可以使用快捷键Alt+3
```

2 解释：

input(“提示语”)，可以在 idle 中显示提示语，等待用户输入，然后将用户输入作为字符串赋值给前面变量。eval 的作用是将字符串转为数字。

当 $d \geq 0$ 为 True 时，执行 if 和 else 之间的三条语句，求出方程的两个解；当 $d < 0$ 计算出 False 时，执行 else 后的语句，提示用户方程无解。

这个程序是典型的 IPO 结构程序，输入方程系数，进行求解，输出结果。

其实这个程序是主教材例题 2.5 (p26)。

四：仿照上例编写代码练习

三角形知道三边长度求面积，会用到“海伦公式”。海伦公式又称希伦公式，传说是古代的叙拉古国王希伦二世发现的公式，可以通过测量三角形的三条边长来求取三角形面积。但根据 Morris Kline 在 1908 年出版的著作考证，这条公式其实是阿基米德所发现，以托希伦二世的名发表。

假设有一个三角形，边长分别为 a、b、c，三角形的面积 S 可由以下公式求得：其中 P 为 $(a+b+c)/2$ 。

$$S = \sqrt{P(P-A)(P-B)(P-C)}$$

操作：新建一个 python 程序，要求输入三角形的三条边长，当输入的 3 个边长构成三角形时，计算该三角形的面积，当输入的 3 个边长不构成三角形，请给出提示。

下面是代码的中国解释，请将其翻译回 python 代码：

- 1 输入三角形边长，赋值给 a （可以用 `a=eval(input("输入第一边长:"))`）
- 2 输入三角形边长，赋值给 b
- 3 输入三角形边长，赋值给 c
- 4 计算 p 值为 $(a+b+c)/2$
- 5 用 if 判断输入的 a,b,c 是不是符合三角形构成 ($a+b>c$ and $b+c>a$ and $a+c>b$)
- 6 如果构成三角形，计算三角形面积 s， s=上面公式 （注意：开平方，可以用类似 `5**0.5`，表示 5 的平方根）。
- 7 否则(不构成三角形),用 print 输出一句话， ” 输入的 3 个数字不构成三角形”

五:简单作曲

理论课：

详见讲义

阅读文献：

练习与思考：

1 依据程序的 IPO 结构，使用输入输出函数和表达式，完成“古尺计算器”程序设计。古代武侠小说里，形容人物的勇猛时，经常这样的描述：此人身高八尺，威风凛凛。“八尺”到底有多高？ 请同学们设计完成一个程序，用来实现古尺的转换计算。



教学反思：

第 4 课 (4 学时)

教材章节:

第 3 章 Python 数据类型和表达式

3.1 基本数据类型

3.2 常量与变量

3.3 运算符与表达式

3.4 常用内置函数

教学目的和要求:

了解 Python 的基本数据类型

掌握变量的概念和赋值方法

熟悉 Python 的运算符和表达式

掌握常用内置函数的使用方法

教学重点和难点

变量的赋值方法

常用函数的使用

教学方法与手段

使用超星学习通线上直播授课, 理论与实践相结合, qq 群辅助教学。

教学内容和教学过程

实践课:

一 实验五 表达式与内置函数

1 在 idle 交互方式下, 输入命令, 查看命令结果后写在书上 (实训教材 P27 P28)

2 输入三角形三边求面积(实训教材 P29)。

三角形知道三边长度求面积, 会用到“海伦公式”。海伦公式又称希伦公式, 传说是古代的叙拉古国王希伦二世发现的公式, 可以通过测量三角形的三条边长来求取三角形面积。但根据 Morris Kline 在 1908 年出版的著作考证, 这条公式其实是阿基米德所发现, 以托希伦二世的名发表。

假设有一个三角形, 边长分别为 a、b、c, 三角形的面积 S 可由以下公式求得: 其中 P 为 $(a+b+c)/2$ 。

$$S = \sqrt{P(P-A)(P-B)(P-C)}$$

操作: 新建一个 python 程序, 要求输入三角形的三条边长, 当输入的 3 个边长构成三角形时, 计算该三角形的面积, 当输入的 3 个边长不构成三角形, 请给出提示。

下面是代码的中国解释, 请将其翻译回 python 代码:

1 输入三角形边长, 赋值给 a (可以用 `a=eval(input("输入第一边长:"))`)

2 输入三角形边长, 赋值给 b

3 输入三角形边长, 赋值给 c

4 计算 p 值为 $(a+b+c)/2$

5 用 if 判断输入的 a, b, c 是不是符合三角形构成 ($a+b>c$ and $b+c>a$ and $a+c>b$)

6 如果构成三角形, 计算三角形面积 s, `s=上面公式` (注意: 开平方, 可以用类似 `5**0.5`, 表示 5 的平方根)。

7 否则(不构成三角形),用 print 输出一句话, ” 输入的 3 个数字不构成三角形”
新建一个程序,输入代码如下,注意红色注释文字可不输入,绿色字符串可简写:

```
File Edit Format Run Options Window Help
#三角形知道三边求面积:
a=eval(input("输入第1条边长度(输入后回车确定): "))
b=eval(input("输入第2条边长度(输入后回车确定): "))
c=eval(input("输入第3条边长度(输入后回车确定): "))
p=(a+b+c)/2 #p为周长的一半

if (a+b)>c and (b+c)>a and (c+a)>b:
    s=(p*(p-a)*(p-b)*(p-c))**0.5
    print("三角形面积为%f"%(s))
    #print("三角形面积为: ",s)
else:
    print("您输入的数据不构成三角形,无法求面积")
```

3 实验, 进制转换器 (实训教材 P30)

实验教材上, e5.2 为将十进制转为其它进制 (2、8、16), e5.3 为将其他进制转化为十进制。下面老师将两个实验综合, 编写以下代码, 大家新建程序, 复制下面代码, 实验运行成功后, 将教材上空格部分填写好。

#进制转化

```
ask=eval(input("将十进制转为其它进制输入 1 回车\n将其它进制转为十进制输入 2 回车"))
```

```
if ask==1:
```

```
    #下面代码就是书上 e5.2
```

```
    x=eval(input("输入一个十进制数字"))
```

```
    x2=bin(x)
```

```
    x8=oct(x)
```

```
    x16=hex(x)
```

```
    print("二进制为%s\n八进制为%s\n十六进制为%s"%(x2,x8,x16))
```

```
elif ask==2:
```

```
    #下面代码就是书上 e5.3
```

```
    base=eval(input("输入转换数的进制 2 或 8 或 16, 回车确定"))
```

```
    x=input("输入一个整数")
```

```
    x10=int(x,base)
```

```
    print("%s 进制的%s 对应的十进制数字为: %d"%(base,x,x10))
```

```
else:
```

```
    print("输入不对, 请重新运行程序, 输入 1 或者 2")
```

二 实验六 常用标准函数库

类似于 turtle 的使用, Python 的部分功能, 需要使用 Import 导入后才可以使用。

1 在 idle 交互方式下, 输入命令, 查看命令结果后写在书上 (实训教材 P32)

2 home 出发的五彩星星 P33 页

实验做完后，填写书上 3 个空

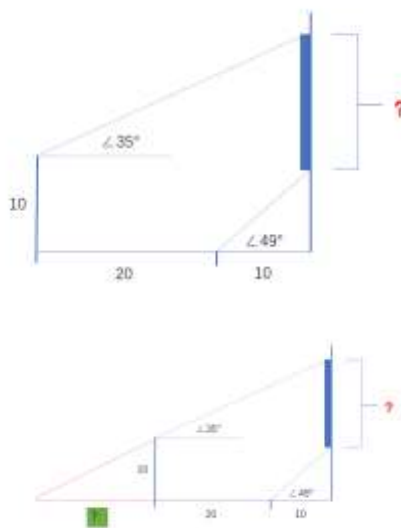
3 随机万花筒程序 (P33) 和随机数求和程序，都是随机库的使用，作为课下学习。

4 time 库的内容等理论课老师讲完后布置作业填写。

5 实训教材 P36, (3) 数学函数 math

1 填好 P36 教材的填空

2 编写程序 P37 平面几何题。



理论课：
详见讲义

思考与练习：

实训教材中有关的填空练习题，在 python 环境中测试，将结果写入
课后复习常用的函数

阅读文献：

《Python 语言程序设计》 第三章

教学反思：

第 5 课（4 学时）

教材章节：

第 3 章 Python 数据类型和表达式

3.5 常用标准库函数

第 4 章 Python 控制语句

4.1 分支结构

教学目的和要求：

掌握随机函数的使用方法

掌握分支结构的基本语句

教学重点和难点

结构化程序设计方法的基本思想

逻辑功能设计和数据流关系

分支结构（选择结构）

教学方法与手段

使用超星学习通线上直播授课，理论与实践相结合，qq 群辅助教学。

教学内容和教学过程

实践课：

因受大多数同学核酸检测影响，本节内容改为课下作业内容，在完成作业的过程中，有问题可以随时 qq 询问，请大家尽量完整描述问题，我看到了会回复。

本节为分支结构的实验，实验教材 7 的 P40 “水费计算器”实验的扩展。根据给定数据区间，利用分支结构进行按类别计算。

作业内容：

0 简单阅读“**阅读材料**”

1 新建程序，将下面**实验一**代码复制到新建程序中，保存运行。

2 按照**实验二**图片，修改实验一代码，保存运行。

3 在实验二的基础上，完成**作业 1**

4 巩固练习，完成**作业 2**

阅读材料：

以下均以家庭人数 4 人以下（含）计算，超过 4 人另有计算方法。

【沈阳居民水费标准】

现行收费标准，于 2016 年 1 月 1 日起实施。

[阶梯水价]

第一阶梯：年度用水量 0~192（含），到户价格 3.30 元/立方米

第二阶梯：年度用水量 192~240（含），到户价格 4.475 元/立方米

第三阶梯：年度用水量 240 以上，到户价格 8.00 元/立方米

【沈阳居民电费标准】

现行收费标准，于 2017 年 8 月 1 日起实施。

[阶梯电价]

第一阶梯：年度用电量 0~2640（含），电价 0.50 元/度

第二阶梯：年度用电量 2640~3720（含），电价 0.55 元/度

第三阶梯：年度用电量 3720 以上，电价 0.80 元/度

【沈阳居民天然气收费标准】

现行收费标准，于 2016 年 1 月 1 日起实施。

[阶梯气价]

第一阶梯：年度用气量 0~280（含），气价 3.16 元/立方米

第二阶梯：年度用气量 280~360（含），气价 3.63 元/立方米

第三阶梯：年度用气量 360 以上，气价 4.74 元/立方米

实验一 实训教材 P40 水费计算器——修改一下教材的内容，根据上面实际数据计算。

要求：输入某家庭年度用水量，计算该年度水费。

知识点：1 input 函数 2 eval 函数 3 分支结构

1 input 函数：

作用：可以在程序运行时暂时中断程序，出现提示信息，等待用户输入数据，用户输入后回车，程序继续。

但要注意的是，输入的数据是字符串。

例子： `x=input("请输入一个数字")`

程序运行后，执行到该行代码，则 idle 中出现“请输入一个数字”，比如输入 10 车，则变量 x 的值为字符“10”。

2 eval 函数

作用：可以将参数转为对应的数据格式，相当于去掉参数的定界符（单引号或双引号）。

例子： `x=eval(input("请输入一个数字"))`

可以将 input 输入的数字去掉双引号，转化为可以进行数学运算的数字。

3 本例可以选择多分支结构编写

操作：请启动 idle，使用“file-new file”菜单命令建立程序，将以下代码复制到新程序中运行通过。

```
#-----
print("-"*30)
x=eval(input("请输入您家的年度用水量: "))

if 0<=x<=2640:
    水费=x*0.5
elif 2640<x<=3720:
    水费=2640*0.5+(x-2640)*0.55 #理解否?
elif x>3720:
    水费=2640*0.5+(3720-2640)*0.55+(x-3720)*0.8 #理解否?

print("本年度您家的用水量为%.0f 立方米，产生水费%.1f 元，赶紧交钱！" % (x,水费))
#-----
```

实验二 根据上例，加上输入的用水量是第几档，在结果中反馈给用户。请修改上面程序，运行通过（注意添加的地方哦）。

```
print("-"*30)
x=eval(input("请输入您家的年度用水量: "))
if 0<=x<=2640:
    水费=x*0.5
    基准="第一档"
elif 2640<x<=3720:
    水费=2640*0.5+(x-2640)*0.55
    基准="第二档"
elif x>3720:
    水费=2640*0.5+(3720-2640)*0.55+(x-3720)*0.8
    基准="第三档"
print("本年度您家的用水量为%.0f立方米，属于%s计费，水费%.1f元，赶紧交钱！" % (x,基准,水费))
```

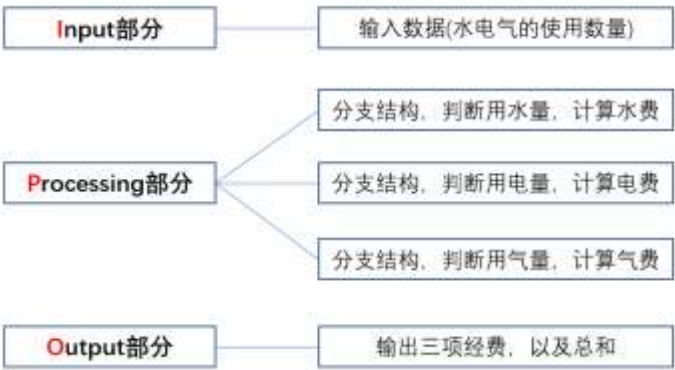
作业 1 请输入您家一年的用水量、用电量、用气量。算算一年开销。

提示 1：可以使用以下方式输入 3 种资源价格。

```
x,y,z=eval(input("请输入您家的年度用水量，用电量，用气量（以英文逗号,分割）："))
```

提示 2：善于使用复制粘贴创建程序结构，并修改对应的部分。

提示 3：结构为 IPO 结构（输入数据、处理数据、输出结果）：



作业 2 bim 指数计算器

BMI 指数（即身体质量指数，简称体质指数又称体重，英文为 Body Mass Index，简称 BMI），是用体重公斤数除以身高米数平方得出的数字，是目前国际上常用的衡量人体胖瘦程度以及是否健康的一个标准。主要用于统计用途，当我们需要比较及分析一个人的体重对于不同高度的人所带来的健康影响时，BMI 值是一个中立而可靠的指标。其计算方法为：

体质指数（BMI）=体重（公斤）÷身高**2（米）

例如：一个人体重 70KG，身高 1.75 米，则 BMI 为

$70\text{kg} \div (1.75 \times 1.75) = 22.86$

根据世界卫生组织定下的标准，亚洲人的 BMI（体重指标 BodyMassIndex)若高于 22.9 便属于过重。亚洲人和欧美人属于不同人种，WHO 的标准不是非常适合中国人的情况，为此制定了中国参考标准：

	WHO标准 [1]	亚洲标准	中国标准 [2]	相关疾病发病危险性
偏瘦	<18.5			低（但其它疾病危险性增加）
正常	18.5~24.9	18.5~22.9	18.5~23.9	平均水平
超重	≥25	≥23	≥24	
肥胖	25.0~29.9	23~24.9	24~27.9	增加
肥胖	30.0~34.9	25~29.9	≥28	中度增加
重度肥胖	35.0~39.9	≥30	——	严重增加
极重度肥胖	≥40.0			非常严重增加

最理想的体重指数是22。

根据上表中的**中国标准**，制作一个 BMI 计算器，当输入某人的身高和体重，计算出相应的 BMI 值，并指出该人是偏瘦还是肥胖等情况。

新建程序，程序简要结构如下

- 1 通过 Input 函数输入某人的身高（单位米）和体重（单位公斤）；
- 2 计算 bmi=身高/体重**2；
- 3 针对 bmi 值进行判断，按照下表给出结论，文字可复制；

Bmi 值	输出的结论
bmi < 18.5	"您的身材偏瘦！ 请注意加强营养"
bmi >= 18.5 And bmi < 24	"恭喜您，您的身材正常！ "
bmi >= 24 And bmi < 28	"您偏胖，请多注意锻炼！ "
bmi > 28 And bmi < 34	"您已经进入肥胖行列，注意您的健康!! "

Bmi>=34	"你完蛋了,比老师都胖! "
---------	----------------

4 显示结论。

理论课:

详见讲义

阅读文献:

《Python 语言程序设计》 第 4 章 P67—P74

练习与思考:

教学反思:

第6课（4学时）

教材章节：

第4章 Python 控制语句

4.3 循环结构

教学目的和要求：

- 1、了解循环结构及其操作
- 2、掌握循环结构基本语句

教学重点和难点

- 1、for 语句循环
- 2、while 语句循环

教学方法与手段

使用超星学习通线上直播授课，理论与实践相结合，qq 群辅助教学。

教学内容和教学过程

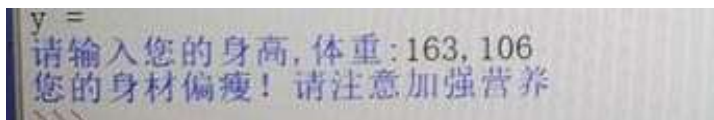
实践课：

No05 循环结构

实验目的：掌握 for 循环结构和 while 循环结构的特点

作业情况：

1 部分同学没有注意到 bmi 计算器的要求：通过 **Input** 函数输入某人的身高（单位米）和体重（单位公斤）；导致计算结果判断有误。



2 部分同学在水电气费用计算的时候，复制粘贴练习代码后未加修改，结果一般不会错，但是建议大家**程序结构修改一下更好**。

```
x,y,z=eval(input("请输入您家的年度用电量,用水量,用气量: "))
```

```
if 0<=x<=2640:
```

```
    电费=x*0.5
```

```
    基准="第一档"
```

```
elif 2640<x<=3720:
```

```
    电费=2640*0.5+(x-2640)*0.55
```

```
    基准="第二档"
```

```
elif x>3720:
```

```
    电费=2640*0.5+(3720-2640)*0.55+(x-3720)*0.8
```

```
    基准="第三档"
```

```
print("本年度您家的用电量为%.0f 度，属于%s 计费，电费%.1f 元，赶紧交钱！" % (x,基准,电费))
```

```
if 0<=y<=192:
```

```
    水费=y*3.30
```

```
    基准="第一档"
```

```
elif 192<y<=240:
```

```
    水费=192*3.30+(y-192)*4.475
```

```
    基准="第二档"
```

```
elif y>240:
```

```
    水费=192*3.30+(240-192)*4.475+(y-240)*8.00
```

```

基准="第三档"
print("本年度您家的用水量为%.0f 立方米，属于%s 计费，水费%.1f 元，赶紧交钱！" % (y,基准,水费))
if 0<=z<=280:
    气费=z*3.16
    基准="第一档"
elif 280<z<=360:
    气费=280*3.16+(z-280)*3.63
    基准="第二档"
elif z>360:
    气费=280*3.16+(360-280)*3.63+(z-360)*4.74
    基准="第三档"
print("本年度您家的用气量为%.0f 立方米，属于%s 计费，气费%.1f 元，赶紧交钱！" % (z,基准,气费))

```

一 for 循环练习

书接上节，我们了解到，在 python 中，一般来说 for 是用来进行已知次数的循环，而 while 用来进行未知次数的循环（并不是绝对的）。上周理论课程讲解了实验教材 P42 （1）和(2) 两种方法求得 100 内所有奇数和，所有偶数和。其中(2)是比较通用的编程方式，即用 for 循环构造所求问题的数字范围，然后利用分支结构选取所需的数字，这种方式更加便于理解。

<pre> sum1,sum2=0,0 for i in range(1,101,2): sum1=sum1+i for i in range(2,101,2): sum2+=i print("奇数和为%d" % (sum1)) print("偶数和为%d" % (sum2)) </pre>	<pre> sum1,sum2=0,0 for i in range(1,101): if i %2==1: sum1=sum1+i else: sum2+=i print("奇数和为%d" % (sum1)) print("偶数和为%d" % (sum2)) </pre>
------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------

练习 1

建立 1 个程序文件，做完(1) 用 Alt+3 注释一下，再做(2)即可。

- 1 求 425~1380 之间所有的偶数和
- 2 求 822 到 2000 之间所有能被 13 整除、并且不能被 5 整除的数字的和（不等号在教材 P51 表格 3.2 中）

练习 2

老师讲解 P43 （3）绘制 9 行 “@” 符号组成的菱形



1 这个菱形，实际上是由上下两个三角形组成的，上三角 5 行@，下三角 4 行@。

2 每行由不同数量的空格和@符号组成。如果红线代表 Idle 的边缘，第五行正好顶头输出，那么能计算出来其它每行前面的空格数么？

3 先来看上面的 5 行正三角形，请填写下表

行号 i	空格数	@ 符号数	空格用含 i 表达式	@符号用含 i 表达式
1	(4)	(1)	(5-i)	(2*i-1)
2	(3)	(3)		
3	(2)	(5)		
4	(1)	(7)		
5	0	9		

做出正三角形程序

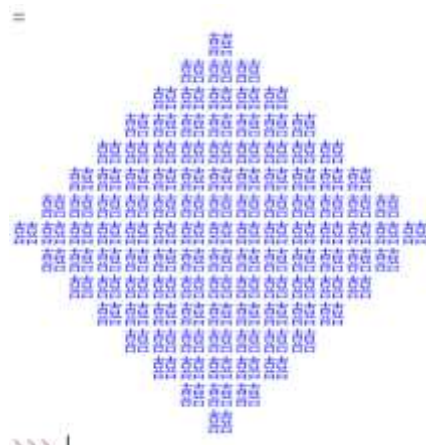
4 再来看下面的 4 行倒三角形，请填写下表



行号 j	空格数	@ 符号数	空格用含 j 表达式	@符号用含 j 表达式
1	1	7	(j)	()
2	(2)	(5)		
3	(3)	(3)		
4	(4)	(1)		

自行做出倒三角形程序。

5 请画出一个由“囍”字组成的 15 行菱形，以上程序该如何修改呢？☺



练习 3

老师讲解 实训教材 P43 (2) 素数 (质数) 算法

素数，又叫质数，一般定义是：只能被 1 和本身整除的数字。

素数算法转 PPT 查看。

理论课
详见讲义

阅读文献：

《Python 语言程序设计》第 4 章 P74

思考与练习：

1 编写程序：已知 2018 年末中国人口为 142706 万，人口增长为 0.381%；印度人口为 135405 万，人口增长为 1.11%，如果按此数据进行推演，印度将在哪一年人口超过中国。程序文件保存为实验 8.3.py。

2 教材第 3 章练习 3.16-3.26

教学反思：

第 7 课 (4 学时)

教材章节:

第 4 章 Python 控制语句

4.3.3 循环的嵌套

4.4 break 语句和 continue 语句

教学目的和要求:

掌握循环在实例中的应用

掌握 break 和 continue 关键字在循环中的使用

教学重点和难点

循环语句在实际中应用

Break 关键字

Continue 关键字

教学方法与手段

使用超星学习通线上直播授课,理论与实践相结合,qq 群辅助教学。

教学内容和教学过程

实践课:

NO07 多重循环和循环关键字

一 关于水仙花数字的另一个算法。

算法 1: 循环数位上的数字,组合成三位数,进行判断。

```
for i in range(1, 10):
    for j in range(0, 10):
        for k in range(0, 10):
            num=i*100+j*10+k
            if num==i**3+j**3+k**3:
                print(num)
```

以上代码是循环某个三位数的百、十、个位数字,然后将循环的变量分别乘以 100, 10, 1, 组成一个三位数 Num, 再进行是不是水仙花数的判断。

算法 2: 循环数字,拆分为百、十位、个位数,进行判断

```
for i in range(100, 1000):
    a=i//100      #a 是 i 百位上的数,为什么?
    b=i//10%10    #b 是 i 十位上的数
    c=i%10        #c 是 i 个位上的数
    if i==a**3+b**3+c**3:
        print(i)
else:
    print("运行完毕")
```

二 循环中的 break 和 continue

1 break: 在一个循环中,可以使用 break 关键字来中断循环的执行。

例 1:

```
for i in range(1,11):  
    print(i)  
    if i==6:  
        break
```

例 2:

如果将 print(i) 放在分支下面呢?

```
for i in range(1,11):  
    if i==6:  
        break  
    print(i)
```

例 3 修改求某范围内素数的算法, 和老师的计算机比比运行速度。

```
import time as tm  
T1=tm.time()  
for n in range(1000000, 1000050):  
    flag=True  
    for i in range(2, n):  
        if n %i==0:  
            flag=False  
    if flag ==True:  
        print("%d 是素数" % (n))  
T2=tm.time()  
print("程序运行时间%. 3f 秒"%(T2-T1))
```

- 1) 将以上程序复制到一个新程序窗口中, 查看程序运行时间。
- 2) 将循环检验的终值 n 改为 $n//2$, 再查看程序运行时间。
- 3) 在循环检验中如果判断出来整除, 在 `flag=False` 后, 加一行 `break`, 再查看程序运行时间。

2 continue: 作用类似 break。但是它只结束当前的 1 次循环, 而不终止循环。

例 1:

```
for i in range(1,11):  
    if i==6:  
        continue  
    print(i)
```

例 2:

```
for i in range(1,11):  
    if i%2 == 0:  
        break  
    print(i)
```

```
for i in range(1,11):  
    if i%2 == 0:  
        continue  
    print(i)
```

以上两段代码显示的数字是什么? 为什么?

三 转演示文稿，第五章 组合数据结构

理论课：
详见讲义

阅读文献：

《Python 语言程序设计》 第 4 章 P107

思考与练习：

以其它形式输出九九乘法表

教学反思：

第 8 课（4 学时）

教材章节：

阶段复习与总结

教学目的和要求：

总结前几周内容，使学生跟上教学进度

教学重点和难点

教学方法与手段

使用超星学习通线上直播授课，理论与实践相结合，qq 群辅助教学。

教学内容和教学过程

实践课：

阅读材料

很久很久很久以前，那时网络还没有普及，人们为了在不同计算机中传输数据，设计了一种存储设备，它的对角线长 5 英寸，称为 **5 寸软盘**，如下面左图，老师在 1996 年上大学的时候用的这种，容量是 1.2 兆，一首 mp3 都存不了，听我的老师说，再以前还有一种 8 英寸软盘，我没用过。后来科技发展了一点点，可以把磁介质的密度变大，又因为 5 英寸软盘是软壳的，容易折断从而损失数据，下一代的软盘就采用了硬壳，对角线是 3.5 英寸，容量是 1.44M，老师大学毕业时，或者说同学们干刚刚出生的时候，基本都是用的这种，下面右图。



当时这两种磁盘价格便宜，基本上成了台式计算机的标配，3.5 寸软盘也在一些笔记本电脑上使用。因为太常用，所以就将 A 盘，B 盘默认分配给了这两种磁盘，所以**硬盘盘符就从 C 开始分配了**。延续到现在，大家看到的计算机即使没有这两种磁盘的驱动器（早就淘汰了），硬盘也是从 C 开始的。



可以读取光盘的设备，称为光盘驱动器，简称光驱

可以读取5英寸软盘的设备，称为5英寸软驱，一般指A盘

可以读取3.5英寸软盘的设备，称为3.5英寸软驱，一般指B盘

这图是老师网上搜的，当年没有透光侧板，没有炫光风扇，也没有CPU水冷设备。这是复古定制主机，定制价格2007美元起。

一块硬盘可以从逻辑上分配多个盘符。比如 C 盘，D 盘，E 盘等等，是用户自己随意分配的（在控制面板的磁盘管理里就可以分配）。C 盘是第一个硬盘盘符，所以操作系统和应用程序一般都会安装在 C 盘，**用户文件最好从 D 盘起保存**，因为一旦系统崩溃，重新安装系统是需要清除 C 盘内容的。

按照一种非常古老的操作系统 Dos 的操作方式（就像大家安装 pillow 用的 cmd 命令窗口），盘符用字母加冒号：组成，文件夹用\+文件夹名。所以：

C:\windows\ 代表了 c 盘下的 windows 文件夹
D:\aaa\bbb\ 代表了 d 盘下的 aaa 文件夹下的 bbb 文件夹
D:\aaa\bbb\abc.jpg 代表了 d 盘下的 aaa 文件夹下的 bbb 文件夹中的 abc.jpg 文件

在 python 中，字符串有个特殊的表达形式叫**转义字符**，是使用\这个符号，加上一些字母等，构成一定特殊的表达形式。比如\n代表着换行，\t代表着 tab 符号。大家可以在 python 的 idle 试试以下命令，看看结果。

```
print("沈阳\n师范\n大学")
```

```
print("沈阳\t师范\t大学")
```

第一个命令的结果是沈阳师范大学分三行显示，第二个结果是沈阳和师范和大学间，都加了 tab 制表位。

那么如果在文件夹描述的时候，总有这种特殊的表达，难免会出现错误。比如

t=im.open("c:\aaa\abc.jpg")就是错误的表达，因为\a当做转义字符了。所以在描述文件路径的时候，我们一般采用\\，因为\\在 python 字符串中的转义就是\。或者可以在字符串前加 r，表示该字符串中所有转义无效。比如下面两个都是对的

```
t=im.open("c:\\aaa\\abc.jpg")
```

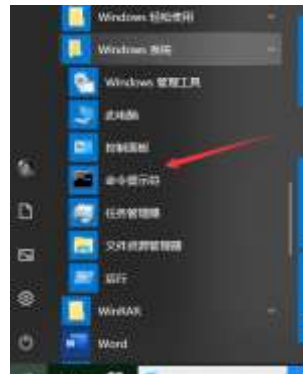
```
t=im.open(r"c:\aaa\abc.jpg")
```

常用转义字符	描述
\\	反斜杠符号
\'	单引号
\"	双引号
\a	响铃
\b	退格(Backspace)
\n	换行
\v	纵向制表符
\t	横向制表符
...	...

以上

5.1 安装 pillow

1) 按 win+r 键，进入 cmd 模式，也可以按照以下位置找命令提示符图标进入。



按 win+r 打开“运行”，输入 cmd

直接在程序组找到“命令提示符”也行

2) 执行命令 `pip install pillow` 如果 pip 命令的版本太古老，要先执行

`python -m pip install --upgrade pip` 进行升级 pip 之后才能再执行 `pip install Pillow` 安装 pillow 库。



5.2 利用 pillow 打开和保存图像文件

1 打开图像 `Image.open(“文件所在位置”)`

例如有一个图像文件在 d 盘下的 aaa 文件夹下的 bbb 文件夹内，可以使用命令“`t=Image.open("d:\\aaa\\bbb\\abc.jpg")`” 使用 t=是为了以后可以使用变量 t 来代替你打开的图像，当然也可以使用 x,y,z 等其它名字，命令执行后，t 就是一个图形对象了。

2 打开图像后，可以直接使用 t 来获得该图像的一些基础信息。

例如：

```
>>> t=Image.open("d:\\aaa\\bbb\\abc.jpg")
>>> t
<PIL.JpegImagePlugin.JpegImageFile image mode=RGB size=250x250 at 0x2BFE128>
>>>
```

信息表示了图片模式(mode=RGB)、图形大小(size=250*250)、内存位置等。

2 显示图像 `show()`

当加载了图像后，可以使用 `t.show()` 来调用当前计算机中的图形查阅软件来查看图像。每台计算机默认的图像打开的软件工具不同，所以在不同的计算机上显示图像所用的工具也不同。

3 保存图片 `save(文件名)`

保存图像命令，不但可以将打开的图像对象另外存储一份，而且可以改变原有图像文件的文件格式，比如打开的是 bmp 图像，在保存时，可以保存为 jpg、gif、psd 等格式。例如：

```
t.save("f:\\photo\\b.bmp")
t.save("f:\\photo\\b.gif")
```

可以分别将图片对象 t 保存为 bmp gif 格式，只要在保存的时候，文件的扩展名写为相应的图片格式的扩展名，就可以。

注意：磁盘和文件夹名称必须要正确。比如上面例子中，要保存在 f 盘的 photo 文件夹，该文件夹必须已经存在。

4 新建图片 new()

参数为 new(模式, size, 颜色)

例如: s=Image.new("RGB", (500,500), "red")

意思是建立一个 rgb 模式 (一般都是这个) 的大小为 500*500 像素的红色图形文件。

如果使用 s.show(), 你将看到一个 500*500 像素的大红方块。如果省略最后的颜色, 默认建立个黑色的图像。模式可以选择以下内容, 通常选用 rgb 模式。

1	1位像素, 黑和白, 存成8位的像素
L	8位像素, 黑白
P	8位像素, 使用调色板映射到任何其他模式
RGB	3×8位像素, 真彩
RGBA	4×8位像素, 真彩+透明通道
CMYK	4×8位像素, 颜色隔离
YCbCr	3×8位像素, 彩色视频格式
I	32位整型像素
F	32位浮点型像素

用程序方式, 一次建立多个颜色文件。

新建程序文件, 输入以下代码 (为了让同学们注意细节, 这几行暂不能复制粘贴):

```
from PIL import Image as im
color=["white","black","yellow","blue","green","pink"]
for i in color:
    t=im.new("RGB",(500,500),i)
    t.save("d:\\%s.jpg"%(i))
```

那么，都可以建立什么颜色的大方块呢！？表格中为常见颜色的英文单词

aliceblue	darkslategray	lightsalmon	palevioletred
antiquewhite	darkturquoise	lightseagreen	papayawhip
aqua	darkviolet	lightskyblue	peachpuff
aquamarine	deeppink	lightslategray	peru
azure	deepskyblue	lightsteelblue	pink
beige	dimgray	lightyellow	plum
bisque	dodgerblue	lime	powderblue
black	firebrick	limegreen	purple
blanchedalmond	floralwhite	linen	red
blue	forestgreen	magenta	rosybrown
blueviolet	fuchsia	maroon	royalblue
brown	gainsboro	mediumaquamarine	saddlebrown
burlywood	ghostwhite	mediumblue	salmon
cadetblue	gold	mediumorchid	sandybrown
chartreuse	goldenrod	mediumpurple	seagreen
chocolate	gray	mediumseagreen	seashell
coral	green	mediumslateblue	sienna
cornflowerblue	greenyellow	mediumspringgreen	silver
cornsilk	honeydew	mediumturquoise	skyblue
crimson	hotpink	mediumvioletred	slateblue
cyan	indianred	midnightblue	slategray
darkblue	indigo	mintcream	snow
darkcyan	ivory	mistyrose	springgreen
darkgoldenrod	khaki	moccasin	steelblue
darkgray	lavender	navajowhite	tan
darkgreen	lavenderblush	navy	teal
darkkhaki	lawngreen	oldlace	thistle
darkmagenta	lemonchiffon	olive	Tomato
darkolivegreen	lightblue	olivedrab	turquoise
darkorange	lightcoral	orange	violet
darkorchid	lightcyan	orangered	wheat
darkred	lightgoldenrodyellow	orchid	white
darksalmon	lightgreen	palegoldenrod	whitesmoke
darkseagreen	lightgray	palegreen	yellow
darkslateblue	lightpink	paleturquoise	yellowgreen

给大家一个以上颜色代码的列表，然后为每种颜色建立一个文件，大小为 1920*1080 的，当然其它尺寸也可以。先在 d 盘建立一个文件夹，aaaaa。

```
['aliceblue','antiquewhite','aqua','aquamarine','azure','beige','bisque','black','bl
anchedalmond',\
'blue','blueviolet','brown','burlywood','cadetblue','chartreuse','chocolate','coral'
```

```
,\
'cornflowerblue','cornsilk','crimson','cyan','darkblue','darkcyan','darkgoldenrod','
darkgray',\
'darkgreen','darkkhaki','darkmagenta','darkolivegreen','darkorange','darkorchid','da
rkred',\
'darksalmon','darkseagreen','darkslateblue','darkslategray','darkturquoise','darkvio
let',\
'deeppink','deepskyblue','dimgray','dodgerblue','firebrick','floralwhite','forestgre
en','fuchsia',\
'gainsboro','ghostwhite','gold','goldenrod','gray','green','greenyellow','honeydew',
'hotpink',\
'indianred','indigo','ivory','khaki','lavender','lavenderblush','lawngreen','lemonch
iffon',\
'lightblue','lightcoral','lightcyan','lightgoldenrodyellow','lightgreen','lightgray'
,'lightpink',\
'lightsalmon','lightseagreen','lightskyblue','lightslategray','lightsteelblue','ligh
tyellow','lime',\
'limegreen','linen','magenta','maroon','mediumaquamarine','mediumblue','mediumorchid'
,\
'mediumpurple','mediumseagreen','mediumslateblue','mediumspringgreen','mediumturquoi
se',\
'mediumvioletred','midnightblue','mintcream','mistyrose','moccasin','navajowhite','n
avy',\
'oldlace','olive','olivedrab','orange','orangered','orchid','palegoldenrod','palegre
en',\
'paleturquoise','palevioletred','papayawhip','peachpuff','peru','pink','plum','powde
rblue',\
'purple','red','rosybrown','royalblue','saddlebrown','salmon','sandybrown','seagreen'
,'seashell',\
'sienna','silver','skyblue','slateblue','slategray','snow','springgreen','steelblue'
,'tan','teal','thistle',\
'tomato','turquoise','violet','wheat','white','whitesmoke','yellow','yellowgreen']
```

1 新建程序文件，第一行可以写 `c=` 复制上面内容到=后面，例如下图：



```
c=['aliceblue','antiquewhite','aqua',
'blue','blueviolet','brown','burlywood',
'cornflowerblue','cornsilk','crimson',
'darkgreen','darkkhaki','darkmagenta',
'darksalmon','darkseagreen','darkslate
blue','deeppink','deepskyblue','dimgray','dc
gainsboro','ghostwhite','gold','goldenrod',
'indianred','indigo','ivory','khaki',
'lightblue','lightcoral','lightcyan',
'lightsalmon','lightseagreen','lightsla
teblue','limegreen','linen','magenta','maroon',
'mediumaquamarine','mediumblue','mediumorchid',
'mediumpurple','mediumseagreen','mediumslateblue',
'mediumspringgreen','mediumturquoise',
'mediumvioletred','midnightblue','mintcream',
'mistyrose','moccasin','navajowhite','navy',
'oldlace','olive','olivedrab','orange','orangered',
'orchid','palegoldenrod','palegreen',
'paleturquoise','palevioletred','papayawhip',
'peachpuff','peru','pink','plum','powderblue',
'purple','red','rosybrown','royalblue',
'saddlebrown','salmon','sandybrown','seagreen',
'seashell',
'sienna','silver','skyblue','slateblue',
'slategray','snow','springgreen','steelblue',
'tan','teal','thistle',
'tomato','turquoise','violet','wheat',
'white','whitesmoke','yellow','yellowgreen']

from PIL import Image
for i in c:
    s=Image.new("RGB", (100,100), i)
    s.save('d:\\aaaaa\\%s.jpg'%i)
```

2 执行以下代码：

```
from PIL import Image
for i in c:
    s=Image.new("RGB", (500, 500), i)
    s.save("d:\\aaaaa\\%s.jpg"%(i))
```

3 然后看看 D:\\aaaaa 文件夹：



每种都是很漂亮的颜色，可以做桌面或者插入 ppt 做背景。

5.3 旋转图片 rotate(角度) transpose(旋转或翻转)

将图片旋转某个角度。例如导入图片后，使用 `t.rotate(45)` 图片将逆时针旋转 45 度，可以使用 `t.rotate(45).show()` 来查看旋转后的图形。

也可以使用 `s=t.rotate(45)` 来将旋转后的图像赋值给另一个图形对象 `s`。再使用 `s.save(各参数)`，就可以另存一张旋转后的图片了。

`rotate(旋转角度, expand=0|1, fillcolor=颜色)`

`expand=0` 保持图像与原图像大小不变，`expand=1` 扩大图像以容纳旋转后的图形。

`fillcolor`，图像旋转后，空白部分填充的颜色。

实验要求：将狗子图片旋转为几个角度，保存，例如下图。然后上传到你的手机，中间放你的图像，发个朋友圈…截图。



```
jd=[45,90,135,180,225,270,315]
```

```
for i in jd:
```

```
    s=t.rotate(i)
```

```
    s.save("d://doge%s.png"%(i))
```

也可以使用以下命令来翻转图片。

```
s=t.transpose(Image.ROTATE_90)      逆时针旋转 90 度，还可以 180 和 270
s=t.transpose(Image.FLIP_LEFT_RIGHT) 水平翻转
s=t.transpose(Image.FLIP_TOP_BOTTOM) 垂直翻转
```

5.4 改变图片大小

打开一个图片后，使用 `resize()` 函数可以改变图片的大小，但是一般是将改变大小后的图片复制给另一个图片对象。例如：

```
t=Image.open('~~~~~')
s=t.resize((500,500))      将 t 改为长宽各 500 像素后赋值给 s, 注意 (500,500) 元组外还有括号
s.show()
s.save('~~~~~')
t= t.resize((500,500))    t 自身改变大小。
```

以上代码可以把图片改为固定的长宽，有时候需要按照图片原有比例缩放图片，可以使用类似以下语句

```
t=t.resize((t.width//5,t.height//5))  将长宽变为原来尺寸的 1/5，因为长宽为整数，所以用整除符号//，不能用除号/。
```

实验要求：你们班级上交了很多的一寸照片，要导入网络平台中，同学们交上来的文件有很多种尺寸（图片在 `bbbbbb` 文件夹中，以下为缩略图，实际大小不一致），但是网络平台要求图片必须是 400 宽 600 高的。请将 `bbbbbb` 文件夹内的 15 张图片都改为符合规定的图片。



```
from PIL import Image as im
import os
path="D:\\bbbbbb\\"
for i in os.listdir(path):      #os.listdir 可以找到 path 路径下的所有文件名
    t=im.open(path+i)
    y=t.resize((400,600))
    y.save(path+"三年九班"+i)
```

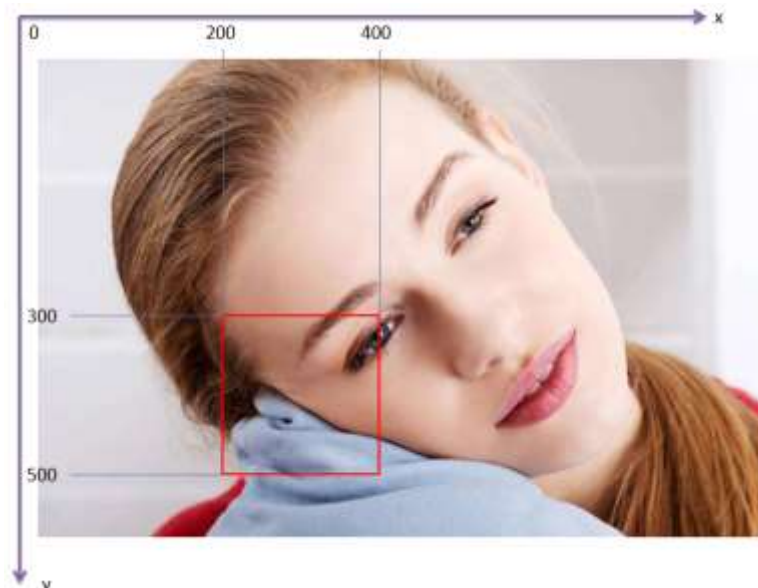


5.5 复制图像的一个区域 crop(区域描述元组)

crop() 方法可以将一个图像的某个区域复制下来。参数是一个 4 个数值的元组。

例如：

B=(200,300,400,500) 首先定义一个元组，各数字含义是：从水平方向的 200 像素，垂直方向的 300 像素起，截止到水平方向的 400 像素，垂直方向的 500 像素。复制的区域是一个大小 200*200 的区域。即宽度 400-200，高度 500-300。



B=(200,300,400,500)

r=t.crop(B) 复制 t 图像中 B 指定的区域，赋值给图像对象 r
r.show()



r.save(.....) 将截取出来的部分保存为一张图片。

实验操作：

请打开“葛优和姜文.jpg” 分别把葛优和姜文的头像截取出来，保存成 2 个 jpg 图片 名称分别为 葛优.jpg 姜文.jpg。(图片大小自定义，大约就好，不用精确)。



已知该图片宽 644，高 364

代码填空(建立一个新的程序文件，命名并保存,将以下代码复制到程序文件，并填写横线上的代码，运行):

```
from PIL import _____
t=Image.open("_____")
b1=(_____)    #葛优头像所在位置，大致即可，不用精确
b2=(_____)    #姜文头像所在位置
r1=t.crop(b1)
r1.show()
_____=t.crop(b2)
r2.show()
r1.save("d:\\葛优.jpg")
r2.save(_____)
```

5.6 粘贴图片 paste()

将一个图像对象粘贴在另一个图像对象上。

#假设 t 和 s 为两个图片对象，即:

t=Image.open(...) #打开一个图片，赋值给图形对象 t，假设是..



s=Image.open(...) #打开一个图片，赋值给图形对象 s，假设是..



s.paste(t) #将 t 粘贴在 s 上面，默认是从 (0,0) 开始粘贴，注意左上角是 0,0。



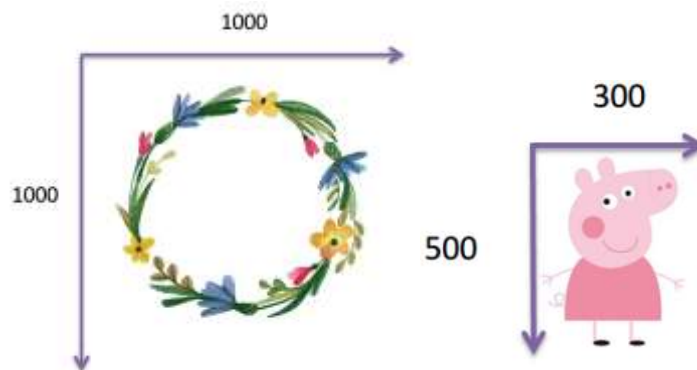
`s.paste(t, (200,200))` 将 `t` 在 `s` 的 `(200,200)` 位置开始粘贴。
例如：



假设，两个图片的大小如下，尝试将佩奇粘贴到底纹的中间。

实验操作：

建立一个新的程序，并将以下中文需求翻译为代码：



- 1 导入 pillow 库中的 Image 模块
- 2 打开图片“底纹.jpg”，赋值给图像对象 dw
- 3 打开“佩奇.jpg”，赋值给图像对象 pq
- 4 将 dw 对象更改大小为 (1000,1000)，赋值给 dw1
- 5 将 pq 对象更改大小为 (300,500)，赋值给 pq1
- 6 将 pq1 粘贴在 dw1 的中间位置上（此处需要小学水平的数学计算）。
- 7 显示 dw1
- 8 将 dw1 保存在 d 盘，起名为组合.jpg

作业 现在有一张沈阳师范大学校徽的图片“校徽.jpg”（如下），导入后（设图形对象名为 `t`），通过 `t.size` 命令知道该图片的大小为 394*394 像素。



现在需要进行如下操作：

1 建立一个 450*450 的红色图片（当然，也可以是别的颜色），然后将校徽图片粘贴到红色图片中间，效果如下图所示。



2 将其拆分为 9 张图片，分别保存图片文件，名字随意取。如下图所示：



3 将其导入手机中。

4 发个朋友圈，写一些祝福的话，获得 1 个以上赞，截图交作业。

理论课：

详见讲义

阅读文献：

《Python 语言程序设计》 第 1-3 章

思考与练习：

复习前 3 章的内容

教学反思：

第 9 课（4 学时）

教材章节：

5.1 组合数据类型

5.2 列表

5.3 元组

教学目的和要求：

了解数据类型

掌握列表的创建、访问和常见更新操作

掌握元组的创建和访问

教学重点和难点

列表的创建和访问

元组的创建和访问

教学方法与手段

使用超星学习通线上直播授课，理论与实践相结合，qq 群辅助教学。

教学内容和教学过程

实践课：‘

实验 1 利用循环将一个图像拆分成若干张。

上次的作业，是将一个（450*450）的图像拆分为了 9 张，每张为（150,150）大小，同学们大多用 2 种方法。

第一种，顺序结构，设置裁剪的 B 值，然后 crop，然后保存，类似下面这样。

```
#-----  
  
.....  
B=(0,0,150,150)  
s=t.crop(B)  
s.save("某某路径文件名")  
  
#-----
```

然后修改 B 值，修改保存的文件名，就可以生成 9 张图片。

```
File Edit Format Run Options
from PIL import Image as im
t=im.open("d:\\1.jpg")
B1=(0,0,150,150)
B2=(150,0,300,150)
B3=(300,0,450,150)
B4=(150,0,150,300)
B5=(150,150,300,300)
B6=(300,150,450,300)
B7=(0,300,150,450)
B8=(150,300,300,450)
B9=(300,300,450,450)
r=t.crop(B2)
r.save("d:\\11.jpg")
```

第二种，循环结构，首先将需要裁剪的 B 值，放在一个列表里，然后用循环来一次性生成 9 张图片。类似下面这样。

```
#-----
.....
B=[(0,0,150,150),(150,0,300,150).....等 9 个 4 坐标数组元组]
n=0
for i in B:
    s=t.crop(B)
    n=n+1
    s.save("d:\\文件名%s.jpg" %(n))
#-----
```

这样做可以，同学们也都运行出来了，但是如果一旦让你将图像拆成 4*4, 5*5, 或者更多的小图，则很难手动定义列表了。比如我有一张 1920*1080 的图片，需要拆分为 10*10 张，应该怎么办呢？如下图所示



这就要求你对循环结构非常熟悉。大家观察以下图片，你能写出每个区域的 4 元组

值么。



下面，我们写出所有 9 个 4 元组，来找找规律。

用循环来表示这个数字序列的**前两个数字**该如何做呢？

实验 2 将任意图片拆分成你希望的张数。

1 在 d 盘建立文件夹 ddddd，复制老师的图片 luck.jpg 到该文件夹下。那么获取该文件的路径就是“d:\\dddddd\\luck.jpg”。

2 启动 idle，使用“file_new file”命令，建立一个程序文件，将以下代码键入程序文件，运行。

```
color.py - C:\python3.6\color.py (3.6.4)
File Edit Format Run Options Window Help
from PIL import Image as im
t=im.open("d:\\dddddd\\luck.jpg")
W=t.width
H=t.height
x=eval(input("您想将图片横向拆成几份？输入数字 (1~10) "))
y=eval(input("您想将图片纵向拆成几份？输入数字 (1~10) "))
w=W//x
h=H//y
B=[(j*w, i*h, (j+1)*w, (i+1)*h) for i in range(y) for j in range(x)]
n=0
for i in B:
    s=t.crop(i)
    n=n+1
    s.save("d:\\dddddd\\a%s.jpg" % (n))
else:
    print("图片拆分完毕，请到d:\\dddddd查看！")
```

代码解释：

1 t.height、 t.width 可以获取图片对象的高度和宽度。本图高度 1080，宽度 1920，即 t.height=1080,t.width=1920

2 h,w 为小图的高度和宽度。为了保持为整数，所以用了整除符号//，比如输入的 x 和 y 都是 10。则小图的尺寸为 192*108。（如果用/，则 w 是 192.0，h 是 108.0，就不能往下进行了，因为图片的像素没有小数）

主题讨论 1 仿照上面操作，将 parrot.jpg 分为 5 行 6 列的小图形，将文件夹截图放于主题讨论。

实验 3 照片墙

如果将若干张图片粘贴为 1 张照片，形成照片墙。也可以利用循环来处理。比如有 15 张大小为 (400,600) 的图片，把他们粘贴为一个 3 行 5 列的大图片。思路是新建一个(2000,1800)的图片，什么颜色无所谓，会被小图片盖住。然后利用循环设定每一张的粘贴点，把小图片粘贴上去。

例如：已知 d:\\ecccc\\下有 15 张图片，大小不一，现需要将他们统一调整为 400*600 大小。然后按照 3 行 5 列的方式，粘贴到一个大图上，如下图所示。



思路：

1 打开 ecccc 下所有图片，修改为(400,600)，这点和上方实验 1 一样，只不过打开

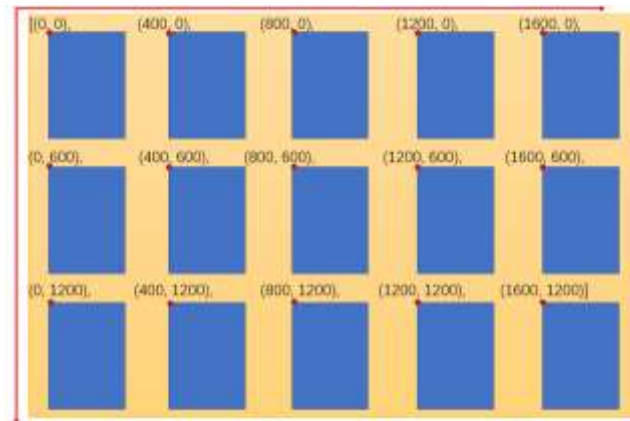
后需要进行往大图上 **paste**，而不是保存。

2 新建一个大图，大小为 (2000,1800)，即宽度是 5 个 400，高度是 3 个 600。

2 形成一组坐标点，见下面示意图（实际上没有缝隙，为了方面查看，我特意留出缝隙了）。

这样，我们需要一个坐标点列表，里面的的是需要粘贴的图像的坐标点。如：

[(0, 0), (400, 0), (800, 0), (1200, 0), (1600, 0),
(0, 600), (400, 600), (800, 600), (1200, 600), (1600, 600),
(0, 1200), (400, 1200), (800, 1200), (1200, 1200), (1600, 1200)]



重点：这样的坐标组，可以通过双循环，或者列表推导式产生

`a=[(j,i) for i in range(0,1800,600) for j in range(0,2000,400)]`

代码的意思是，a 是一个列表，列表中由坐标对 (j,i) 元组组成，j,i 的数值，是双循环的变量组合：

`for i in range(0,1800,600)`

`for j in range(0,2000,400)`

例如 i=0 时，j 可以循环到 0,400,800,1200,1600。组成的坐标(j,i)为

(0,0) (400,0) (800,0) (1200,0) (1600,0)。正好是第一行的粘贴点。

操作：

1 在 d 盘建立文件夹 eeeee，将老师提供的 15 张图片放进去。确保文件夹内没有其它多余内容。

2 启动 idle，使用“file_new file”命令，建立一个程序文件，将以下代码键入程序文件，运行。

```
from PIL import Image as im
import os
back=im.new("RGB",(2400,1800),"white")
a=[(j,i) for i in range(0,1800,600) for j in range(0,2000,400)]
path="d:\\eeeeee\\"
n=0
for i in os.listdir(path):
    t=im.open(path+i).resize((400,600))
    back.paste(t,a[n])
    n=n+1
back.show()
# 最后，将 back 对象保存到 d 盘，形成 jpg 图像，起个名，怎么做？
```

实验 4：以下 1、2 任选 1 个。

1 疫情期间，同学们在校（家）学习很辛苦，请拍摄（或从你的相册选择）若干张图片组合成一个大图（不可以是 3 行 5 列的，因为和以上代码几乎重合），保存为 1 张图片交作业。大家可以自由发挥，比如沈师景色，我的寝室小伙伴，抗疫主题，同学们的学习状态，班级同学大头照，我的一个喜欢物品等等。resize 为相同尺寸，再用这些小图组成 1 个大图片。

2 如果实在没有照片，可以将第一课生成的颜色块选择 35 张，改为小一些，比如（160,90），然后粘贴为 5 行 7 列。



理论课：

详见讲义

阅读文献：

《Python 语言程序设计》第 5 章 P86

思考与练习：

教材第五章后习题：P105。

教学反思：

第 10 课（4 学时）

教材章节：

第 5 章 Python 数据结构

5.3 字典

5.4 集合

教学目的和要求：

1、掌握创建集合的基本方法

2、掌握访问集合的常用方法

教学重点和难点

修改列表元素值

教学方法与手段

使用超星学习通线上直播授课，理论与实践相结合，qq 群辅助教学。

教学内容和教学过程

2022 年 3 月 14 日，GIF 动图的发明者 Steve Wilhite 因新冠并发症去世，享年 74 岁。GIF 是一种广泛应用于互联网的图像，可以是静态的，也可以是动态的。今天的第一个实验，我们做一个 gif 来纪念这个创始人吧。



实验一：利用 pillow 库生成 gif 动图。

原理：将准备好的若干张图片依次打开，并存入 1 个列表，然后利用 Pillow 库的 save 方法将列表保存为间隔若干毫秒的动态图像。

第 1 步，导入库，设定图片所在文件夹

```
from PIL import Image
```

```
path="d:\\aaa\\" #假设你有几张图片放在了 d:\\aaa\\下
```

第 2 步，建立一个空列表

```
piclist=[]
```

第 3 步，依次打开图片，将图片加入列表中

```
im=Image.open(path + 文件名 1)
```

```
piclist.append(im)
```

```
im=Image.open(path + 文件名 2)
```

```
piclist.append(im)
```

```
im=Image.open(path + 文件名 3)
```

```
piclist.append(im)
```

```
.....
```

#有几个图片需要加入 gif 图像组，就执行几次上面命令，发现了么，**利用循环更方便处理**，代码附后。

第 4 步，用列表第一张图像的 save 方法，将列表其它图像加入，生成 gif。

```
piclist[0].save("d:\\文件名.gif",save_all=True,append_images=piclist[1:],duration=300,loop=3)
```

参数解释：

save_all=True ： 如果加入图片列表的是动态图像，那么是只将动态图像的第一帧加入 gif，还是将动态图像的所有帧加入 gif，设置为 True 就是所有帧，默认选项是 True。

append_images=piclist[1:] ： 列表的切片，从 1 号索引到最后，即将 piclist 中所有图片加入 gif。

Duration=300: 两张图片之间的时间间隔，单位毫秒，数字越小切换的越快

Loop :是否循环， 没有 loop 参数，只循环 1 次；加上 loop=0，则一直循环，直

到关闭图像；loop=n，循环 n 次。

第 5 步 将加入到图像列表的图片改为统一大小。

只需要将 im=Image.open(path + 文件名 1)这样的语句改为：

```
im=Image.open(path + 文件名 1).resize((宽度值,高度值))
```

附循环部分代码，要求，在 Path 下(d:\\aaa\\)只有图片文件，没有其他文件或者子文件夹等，就可以用下面循环将文件夹下的所有图片加入 gif 了。

```
import os
```

```
for i in os.listdir(path):
```

```
    im=Image.open(path + i) .resize((宽度值,高度值))
```

```
    piclist.append(im)
```

实验二：签到打卡机系列

这个系列的实验，主要是要掌握 time 库的三个函数 localtime(), strftime(), asctime()的基础上，结合分支、循环、列表等结构进行扩充的实验。

1 简化版，输入 1 个学号，显示该学号什么时间打卡的

新建一个 py 文件，将以上代码复制进去，运行。输入一个学号（随便输入即可），可以看到输出的信息。

```
import time as t
```

```
xh=input("请输入学号：")
```

```
print (xh,"号",t.asctime(),"打卡")      #标准字符格式显示时间元组
```

2 P35 签到打卡机 1.0 版——比简化版多了显示日期的功能，以下代码比教材的多加了 2 条线。

注释上面代码，将下面代码复制到程序中运行

```
import time as t
```

```
xh=input("请输入学号：")
```

```
时间格式="%Y 年%m 月%d 日"    #这里跟教材有点区别，你也可以自己改造下
```

```
t1=t.localtime()
```

```
print("+"*40)
```

```
print("当前日期",t.strftime(时间格式,t1))
```

```
print (xh,"号",t.asctime(),"打卡")
```

```
print("+"*40)
```

运行结果大致如下图

```
请输入学号: 332223
++++++++++++++++++++++++++++++++++++
当前日期 2021年04月25日
332223 号 Sun Apr 25 20:05:27 2021 打卡
++++++++++++++++++++++++++++++++++++
```

可以看到，在显示打卡时间的时候，用的 `asctime` 有点复杂，实际上只显示时分秒就可以了，下面对这个进行改造。

3 签到打卡机 1.0 改版。P35 下方。

```
import time as t
```

```
xh=input("请输入学号: ")
```

```
时间格式="%Y 年%m 月%d 日" #定义打卡日期格式
```

```
时间格式 1="%H: %M: %S" #定义打卡时间格式
```

```
t1=t.localtime() #t1 为当前系统时间
```

```
print("+"*40)
```

```
print("当前日期",t.strftime(时间格式,t1))
```

```
print (xh,"号",t.strftime(时间格式 1,t1),"签到打卡")
```

```
print("+"*40)
```

运行结果大致如下图

```
请输入学号: 123456
++++++++++++++++++++++++++++++++++++
当前日期 2021年04月25日
123456 号 20: 16: 25 签到打卡
++++++++++++++++++++++++++++++++++++
```

在打卡时间显示上，精简了很多

4 签到打卡机 2.0 P49 前面的打卡机运行后只能 1 个人签到，2.0 版变为了可以连续签到打卡的版本。

```
import time as t
```

```
时间格式="%Y 年%m 月%d 日"
```

```
时间格式 1="%H: %M: %S"
```

```
while _____?_____ #怎么填写能让循环一直进行下去来着
```

```
    xh=input("请输入学号: 输入 0 退出程序")
```

```
    t1=t.localtime()
```

```
    if xh=="0":____? _____ #如果输入 0，就退出循环。什么关键字一运行就结束循环?
```

```
    print("+"*40)
```

```
    print("当前日期",t.strftime(时间格式,t1))
```

```
    print (xh,"号",t.____?_____(时间格式 1,t1),"签到打卡") #白给的空
```

```
    print("+"*40)
```

运行结果如下

```
-----
请输入学号：输入0退出程序123456
+++++
当前日期 2021年04月25日
123456 号 20: 28: 22 签到打卡
+++++
请输入学号：输入0退出程序234567
+++++
当前日期 2021年04月25日
234567 号 20: 28: 25 签到打卡
+++++
请输入学号：输入0退出程序0
```

4 签到打卡机 P52，这个你自己写啦

实验三 P54（6）身份证验证真伪程序(见数组 ppt)

课堂小测验（15 分钟）

理论课：
详见讲义

阅读文献：

《Python 语言程序设计》第 5 章 P95

思考与练习：

教学反思：

第 11 课（4 学时）

教材章节：

第 6 章字符串与正则表达式

6.1 字符串的格式化

6.2 字符串的基本操作

6.3 字符串函数与方法

教学目的和要求：

掌握字符串的索引与分片

教学重点和难点

字符串的格式化

字符串的索引

教学方法与手段

使用超星学习通线上直播授课，理论与实践相结合，qq 群辅助教学。

教学内容和教学过程

一 索引和切片练习

Python 中，具有顺序性质的数据类型有字符串，列表，元组等。这些对象都可以使用中括号 []，在括号内加入数字表达式，来截取其中的 1 个或者一段元素，称作索引或者切片。

今天的练习，以字符串为例，大家要熟练掌握索引和切片的使用。

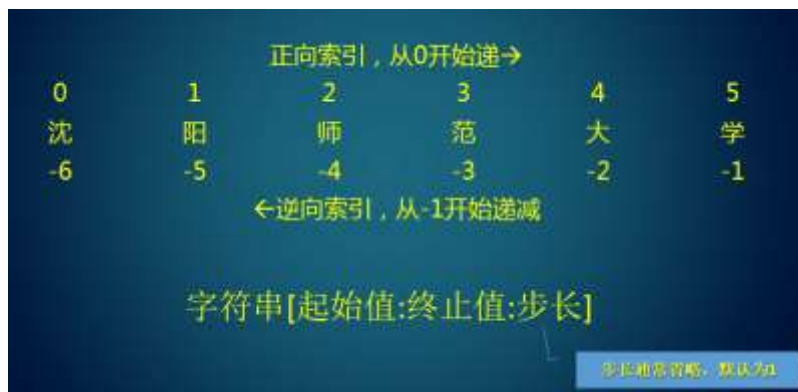
知识点：

1 一个字符串从**前**往**后**的索引号是从 0 开始的；

2 一个字符串从**后**往**前**的索引号是从 1 开始的；

3 切片表达为 字符串[起始值:终止值:步长]，起始值包括，终止值不包括。

步长通常会省略，默认为 1。步长为正，截取的切片是正向的；步长为负，截取的切片是逆向的。



例如：

```
x="沈阳师范大学"
```

```
print(x[0])           #结果为 沈
```

```
print(x[-2])          #结果为 大
```

```
print(x[1:4])          #结果为 阳师范
```

```
print(x[-5:-2:1])      #结果为 阳师范
```

```
print(x[-2:-5:-1])     #结果为 大范师
```

知识点：

4 [:] 省略冒号前面数字，表示从头开始，包含第 0 位；省略冒号后面数字，表示到最后，包含最后 1 位。

```
x="沈阳师范大学"
```

```
print(x[:])            #结果为 沈阳师范大学
```

```
print(x[:4])           #结果为 沈阳师范
```

```
print(x[4:])           #结果为 大学
```

```
print(x[-4:])          #结果为 师范大学
```

```
print(x[:-4])          #结果为 沈阳
```

知识点：

5 当字符串切片起始值、终止值、步长有矛盾时，切片为空。

字符串[小:大:正数] 正序字符串

字符串[小:大:负数] 空串

字符串[大:小:正数] 空串

字符串[大:小:负数] 逆序字符串

```
string="沈阳师范大学"
```

```
print(string[1:4:2])      #阳范
```

```
print(string[1:5:-1])    #空
```

```
print(string[5:1:1])    #空
```

```
print(string[4:1:-1])    #大范师
```

二 索引和切片练习

请在教材或练习册上认真整理以下命令和运行结果。(也许、大概、可能会提问)

```
string="abcdefghij"
print(string[:5:1])
print(string[:5:])
print(string[5:1])
print(string[5:])
print(string[:1])
print(string[::])
print(string[:])
print(string[-8:-4:+1])
print(string[-4:-8:-1])
print(string[: -4: +1])
print(string[-8: :+1])
print(string[: -8: -1])
print(string[-4: : -1])
print(string[: : -1])
```

三 字符串的运算

1 字符串可以进行加法和乘法运算。加法表示连接，乘法表示重复。

```
>>> a="沈阳"
```

```
>>> b="师范"
```

```
>>> c="大学"
```

```
>>> print(a+b+c)
```

#结果为 _____

```
>>>print("海草"*4+"随风飘摇"*2)
```

#结果为 _____

2 字符串可以使用 in 进行包含判断,看前面字符串是否完整在后面字符串出现,结果为 True 或 False。

```
>>>print("大学" in "沈阳师范大学")
```

#结果为 _____

3 字符串可以比较大小,按编码比较,简单说就是 0—9, A—Z , a—z 依次增大。

```
print("abc"<"aba")
```

#结果为 _____

```
print("abcde"<"abcdef")
```

#结果为 _____

四 字符串的函数和方法

1 len(s): 可以求字符串一共多少个字符,也可以求列表,元组,集合等的元素个数。

```
print(len("沈阳师范大学"))
```

#结果为 _____

2 ord(s): 求一个字符的编码,ord(“A”)结果是 65, ord(“a”)结果是 97, 其它字符依次加 1。注意,汉字和各种符号的编码也可以用 ord 求得。尝试以下命令

的结果，最好用 `print`(下面表达式)：

```
ord("π")
```

```
ord("我")
```

```
ord("※")
```

每一个字符都有对应的数字编码。

3 `chr(n)`：求某个编码对应的字符是什么字符，是 `ord` 的反函数。

试试以下命令的结果。

```
chr(100)
```

```
chr(25105)
```

```
chr(29233)
```

```
chr(65509)
```

4 `str(n)` 将数字转化为字符串。在 Python 中，数字和字符是不能相加的，比如 `"a"+5` 就是一个错误的表达，但是有些情况下还需要将字符和数字连在一起，这时可以使用 `str` 函数将数字转化为字符后，进行操作，比如 `"长江"+str(7)+"号"` 结果为“长江 7 号”。以下代码的运行结果为：

```
for i in range(1,10):
```

```
    print("小朋友" + str(i) + "号")
```

五 字典练习（作业，不留学习通，课下完成即可）

1 实训教材 P56 字典部分填空题

2 实训教材 P57 字典程序——建立用户名：密码字典。

新建程序，运行通过。

理论课：

详见讲义

阅读文献：

《Python 语言程序设计》第 6 章 P107

思考与练习：

教学反思：

第 12 课（4 学时）

教材章节：

第 7 章 自定义函数与模块

7.1 函数的定义

7.2 函数的调用

7.3 函数的参数和返回值

教学目的和要求：

- 1、掌握调用函数的三种方式。
- 2、了解参数传递的方式。

教学重点和难点

- 1、系统函数的使用方法。
- 2、自定义函数的使用方法。

教学方法与手段

使用超星学习通线上直播授课，理论与实践相结合，qq 群辅助教学。

教学内容和教学过程

实践课：

一 字符串格式化练习 实训教材 P61

```
>>> s='Python'
>>> "{:20}".format(s)           #显示结果为_____
>>> "{:>20}".format(s)          #显示结果为_____
>>> "{:^20}".format(s)          #显示结果为_____
>>> "{0:*^10}I love {0:*<10}".format(s) #显示结果为_____
>>> "{0:\xa9^10}I love {0:\xa9<10}".format(s) #显示结果为_____
>>> "{0:-^20}".format(1234567890) #显示结果为_____
>>> "{0:-^20,}".format(1234567890) #显示结果为_____
>>> "{0:e},{0:E},{0:f},{0:%}".format(3.1415926) #显示结果为_____
>>>
```

进入教室早的同学自己先做，老师会讲解。其他字符串部分的练习，在理论课讲解。

二 自定义函数和模块

先不用看书，概念太多，会懵。

代码的重复使用，在编程中非常重要，当设计出了一项功能时，该功能可不可以方便地使用在其它地方，能不能在另一个文件中也使用，是个很重要的问题。

二 自定义函数

1 建立一个输出星星的函数

请大家建立一个程序，名字起的简单点，比如 abcde 啥的，不要更改保存文件夹，

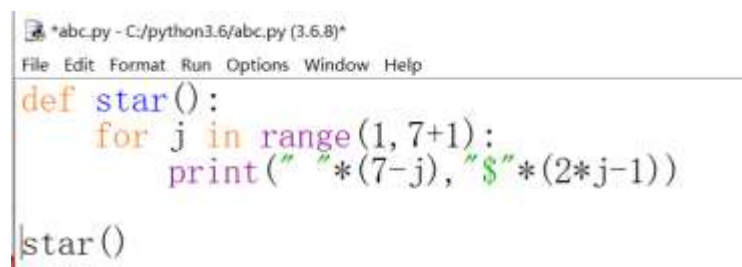
就保存在 python 的安装路径下。写两行代码：

```
for j in range(1,7+1):  
    print(" "*(7-j),"$"*(2*j-1))
```

运行后，你会看到一个 7 行的由*组成的等腰三角型。如果将程序中的数字 7 改为其它数字，就会出现不同行数的图形；将\$改为其它符号，三角形就由你指定的符号构成。

如果在一个程序中，需要多次使用这样的等腰三角形，可以将这两行代码使用 def 语句起一个名字，定义为一个自定义函数。

例如：



```
*abc.py - C:/python3.6/abc.py (3.6.8)*  
File Edit Format Run Options Window Help  
def star():  
    for j in range(1,7+1):  
        print(" "*(7-j),"$"*(2*j-1))  
star()
```

然后，在程序中就可以使用函数的名字调用它。

当这样的文件保存在 python 的安装路径下时，我们可以使用 `import <文件名>` 来导入该文件中使用 def 定义的函数，就可以在 idle 等位置使用了。

2 建立一个切分图片的函数

将以下代码复制到一个 python 文件中，运行前需要先在 d 盘准备一个叫 luck.jpg 的图片，或者将下面 `t=im.open("d:\\luck.jpg")` 代码中的文件名改为你自己的图片文件名。然后在 d 盘下需要有 ddddd 文件夹，或者在下面 `s.save("d:\\dddddd\\a%s.jpg"%(n))` 和 `print("图片拆分完毕，请到 d:\\dddddd 查看！")` 代码中的 `d:\\dddddd` 改为你自己的文件夹。

```

from PIL import Image as im
t=im.open("d:\\luck.jpg")
W=t.width
H=t.height
x=eval(input("您想将图片横向拆成几份？输入数字（1~10）"))
y=eval(input("您想将图片纵向拆成几份？输入数字（1~10）"))
w=W//x
h=H//y
B=[(j*w,i*h,(j+1)*w,(i+1)*h) for i in range(y) for j in range(x)]
n=0
for i in B:
    s=t.crop(i)
    n=n+1
    s.save("d:\\dddddd\\a%s.jpg" % (n))
else:
    print("图片拆分完毕，请到 d:\\dddddd 查看！")

```

然后将这段程序改为一个自定义函数的形式

3 函数的返回值

以上调用函数的方法，都是以一种单行命令的形式调用的。例如：

```

star()

star1(6)

```

但是我们使用函数，更常见的是放在一个表达式中。如

```

x=round(105.76,1)+ord("a")-len("abcd")

y=eval(input("输入数字"))

print(chr(100))

```

如果自定义的函数，也要使用在表达式内，必须使用返回值语句 **return**。

比如我们定义一个函数 `add`，接受两个参数 `x` 和 `y`，函数的结果是这两个参数的和。

```

def add(x,y):

    return x+y

```

注意这个 return ，表示函数 add 的结果，定义好之后，就可以使用类似下面语句来调用这个函数。

```
x=add(3,9)

print(x)

print(add(4,5))
```

4 建立一个输出斐波那契数列第 n 项值的函数。

```
def fib(n):
    flist=[1,1]
    for i in range(2,n):
        flist.append(flist[i-1]+flist[i-2])
    return flist[n-1]
```

该函数 return 后的值，称为函数的返回值，即函数的运行结果，这样函数就可以应用到表达式计算中了。

理论课：
详见讲义

阅读文献：

《Python 语言程序设计》第 5 章 P123

思考与练习：

教材第六章后习题：6.18、程序练习题 4.4-4.5
自定义阶乘函数，求组合数。

$$C_m^n = \frac{m!}{n!(m-n)!} = C_m^{m-n}$$

自定义绘制五角星函数，绘制不同参数的五角星。（自行设计参数）
使用随机数函数和自定义函数，完成可以随机位置绘制五角星的程序。

教学反思：

第 13 课（4 学时）

教材章节：

- 第 8 章 文件与异常处理
- 8.1 文件的概念和基本操作
- 8.2 文本文件的操作

教学目的和要求：

- 掌握函数作用范围
- 掌握函数的嵌套定义
- 理解递归的核心思想

教学重点和难点

- lambda 函数的定义。
- 函数的调用过程。
- 自定义函数的应用实例。

教学方法与手段

使用超星学习通线上直播授课，理论与实践相结合，qq 群辅助教学。

教学内容和教学过程

二 文件操作

文件是数据的集合，存储在存储器中。按照访问形式不同，可以分为文本文件和二进制文件，图形图像就属于二进制文件。这里只讨论文本文件的读写操作。文件的操作过程为：

首先，打开文件

然后，操作文件（读取/写入）

最后，关闭文件

1 写入一个文本文件内容

在讲字符的时候大家做过一个实验，就是向文本文件中写入文本信息。

下面来写入一个文本文件，建立一个程序，复制以下代码，然后运行，在 d 盘找到文件 1.txt，打开看看。

```
file=open("d:\\文件 1.txt","w",encoding="utf-8")
file.write("你想写点什么就写点什么吧")
file.write("每个 write 都会连续写入文件，不会换行的")
file.write("想要换行，要使用转义字符啥来着")
file.write("这是最后一段")
```

```
file.close()
```

说明，`open()` 函数用来打开一个文件，如果文件不存在，会新建一个。 参数

“w” 代表打开文件的目的是写入信息，`encoding` 函数说明文件编码格式，该参数可以省略。`Write` 在写入文件内容时，不会自动换行，如果需要换行，可以在字符串中使用转义符 `\n`。

“w” 方式打开文件后，如果原有的文件里有内容，w 方式会删除源文件中的所有信息，再写入新的内容，每次都是程序最新的运行结果。

2 往文件中追加数据

```
file=open("d:\\文件 1.txt","a",encoding="utf-8")
file.write("添加的内容")
file.close()
```

说明：参数 “a” 代表打开文件的目的是写入信息，并且是不改变原有文件内容，往文件的最后追加内容，a 即 `append` 的意思。

3 读取文本文件内容

如果有一个文本文件，我们需要读取其中的内容到 python 中，也可以使用 `open` 语句。先在 d 盘建立一个文本文件，假设名字叫 “字符.txt”，输入任意字符内容，比如下面这几段：

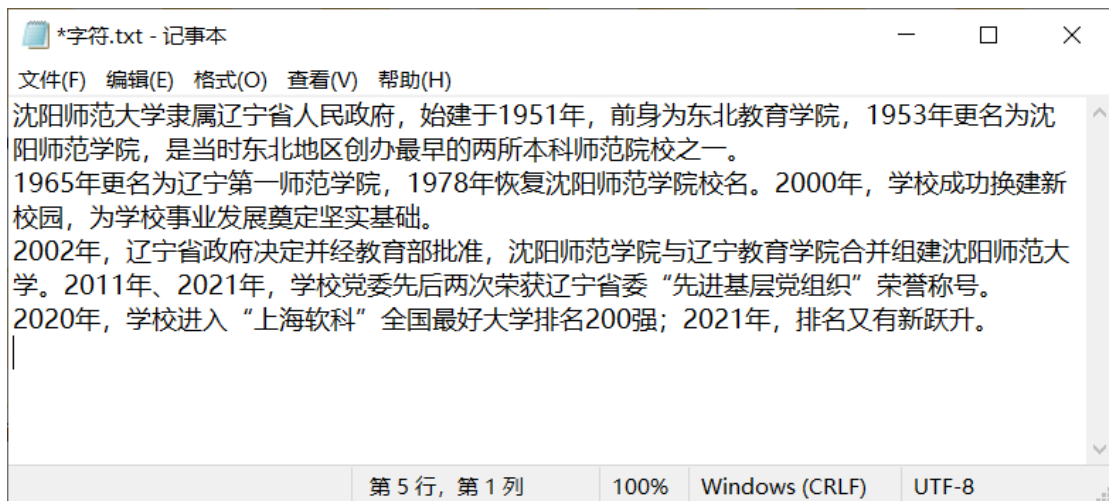
沈阳师范大学隶属辽宁省人民政府，始建于 1951 年，前身为东北教育学院，1953 年更名为沈阳师范学院，是当时东北地区创办最早的两所本科师范院校之一。

1965 年更名为辽宁第一师范学院，1978 年恢复沈阳师范学院校名。2000 年，学校成功换建新校园，为学校事业发展奠定坚实基础。

2002 年，辽宁省政府决定并经教育部批准，沈阳师范学院与辽宁教育学院合并组建沈阳师范大学。2011 年、2021 年，学校党委先后两次荣获辽宁省委“先进基层

党组织”荣誉称号。

2020 年，学校进入“上海软科”全国最好大学排名 200 强；2021 年，排名又有新跃升。



建立程序：

```
file=open("d:\\字符.txt","r",encoding="utf-8")
str=file.read()
print(str)
file.close()
```

说明：open 中使用关键字”r”，指打开文件的目的是读取其中的内容，这时文件必须存在，否则出错。文件对象的 read() 方法会读取文件中所有内容。

修改程序为：

```
file=open("d:\\字符.txt","r",encoding="utf-8")
for i in range(5):
    str=file.read(10)
    print(str)
file.close()
```

则程序每次读取文件中的 10 个字符。

修改程序为：

```
file=open("d:\\字符.txt","r",encoding="utf-8")
str=file.readline()
print(str)
file.close()
```

readline 每次读取文件中的 1 段内容。

理论课：

详见

阅读文献：

《Python 语言程序设计》 第 5 章 P143

思考与练习：

教材第 6 章后习题

教学反思：

。

第 14 课（4 学时）

教材章节：

期末总复习

教学目的和要求：

复习学过的所有知识点

教学重点和难点

函数

数据结构

教学方法与手段

使用超星学习通线上直播授课，理论与实践相结合，qq 群辅助教学。

教学内容和教学过程

复习讲过的内容

知识点总结

测试题

阅读文献：

《Python 语言程序设计》 第 1-7 章

思考与练习：

将学习通上的实验做一遍

教学反思：

。