高级语言程序设计
Python
授课教案

课程名称：高级语言程序设计 Python

课程性质：通识必修课

授课时间：2020-2021 学年第二学期

授课对象：2020 级非计算机专业本科生

授课教师：

第 0 周（2 学时）

教材章节：

第 1 章 Python 语言概述

1.1 从计算机到编程　1.2 产生与特性　　1.3 安装与运行

教学目的和要求：

1、 了解计算机语言的演变、Python 语言特性

2、 掌握 Python 环境的下载和安装、熟悉 IDLE 环境使用

教学重点

1、 Python 环境的下载和安装

教学难点

1、 Python 安装中常见问题

2、 IDLE 环境下两种程序执行方式

教学方法与手段

课前发布任务，学生进行线上学习，线上完成任务

线上教学过程设计

第 1 章 Python 语言概述

1.1 从计算机到编程

一、程序的演变

编程其实就是把人类的需求用计算机语言来表达，是一场人与计算机的对话。

计算机语言经历了从机器语言、汇编语言，再到高级语言的演变过程。

二、高级语言的运行机制

编译程序对源程序进行解释的方法相当于日常生活中的“整文翻译”。

解释程序对源程序进行翻译的方法相当于日常生活中的“同声传译”。

线上资源：课堂实录 1.6.1

1.2 Python 产生与特性

Python 语言的诞生，Guido van Rossum，Python 语言创立者，2002 年 Python 2.x，2008 年 Python 3.x。Python 语言的优势：1. 语法简单 2. 可移植性 3. 粘性扩展 4. 开源理念 5. 面向对象。例：.Hello World，体会 Python 语言的简洁

```
>>> print("Hello World")
Hello World
>>> print("世界，你好")
世界，你好
>>>
```

```
#include <stdio.h>
int main(void)
{
printf("Hello World\n");
return 0;
}
```

一般来说，同样功能的程序，Python 语言实现的代码行数仅相当于 C 语言的 1/5 至 1/10，简洁程度取决于程序的复杂度和规模。

线上资源：课堂实录 1.6.1

1.3.Python 环境安装与运行

一、下载与安装

网址：www.python.org/downloads/，根据操作系统不同选择不同版本，下载相应的 Python 3.0 系列版本程序。

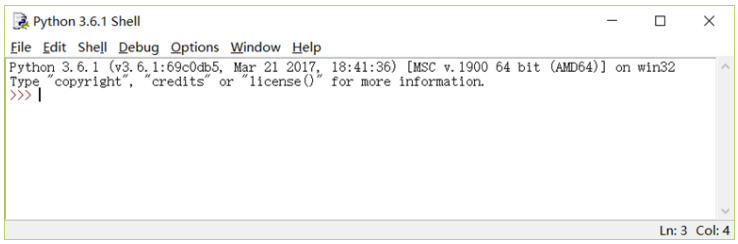
线上视频资源： 1.2.1 下载与安装

线上帖子：安装常见问题“Python 安装遇到的坑汇总，看这一篇就够了！”

二、Python 的运行

Windows 的“开始”-“程序”-“Python 3.5”-“IDLE (Python 3.5 64bit)”

可以启动内置的解释器（IDLE 集成开发环境）



线上资源：课堂实录 1.6.1

线上学习需完成的任务

1. 观看发布的视频：学习指导和课程要求
2. 完成问卷：课程要求及考试形式的问卷
3. 线上任务 1：相关试题
4. 主题讨论：挑战任务 1，自行完成安装并发帖

教学后记

1. 疫情原因开学晚一周，因此，提前布置了线上学习任务，学生任务完成情况达到 90%以上，同学们能够积极反馈，效果不错。

2. 提前发布了“学习指导和课程要求”的视频，在开始前让学生了解课程的考核方式、线上活动形式、线上成绩权重、课程的学习技巧，同时提出具体的要求，提醒和约束同学们认真对待课程学习，尽快适应。

3. 布置线上了问卷，以此形式检查学生对课程要求的了解情况，借此让学生熟悉并使用学习通，学会利用线上课程资源，熟悉授课形式，逐渐形成学习习惯，能够适应线上线下的学习形式。

4. 通过观看线上发布的实验 1 操作视频，环境的下载和安装比较顺利。在线上发布了“主题讨论”同学们可以把安装遇到的问题发一来，一起讨论。

第一周（4 学时）

教材章节：

第 1 章 Python 语言概述

1.4 基本语法　　1.5 turtle 库绘图

教学目的和要求：

1. 掌握基本语法规则
2. 命令行、文件执行方式
3. 标准库的导入
4. Turtle 库绘制基本图形

教学重点

1. 基本语法、程序的执行方式
2. 标准库的导入
3. Turtle 库绘制基本图形

教学难点

1. IDLE 环境下两种程序执行方式
2. 标识符命名规则、缩进、注释
3. Turtle 库常用函数使用

教学方法与手段

理论课利用学习通开展线上活动与学生的互动

实验课利用超星线上资源，开展线上自主学习，教师进行具体的辅助指导

理论课教学过程设计

第 1 章 Python 语言概述

Python 的运行方式：P19

1．命令行方式

命令行方式是一种交互式的命令解释方式

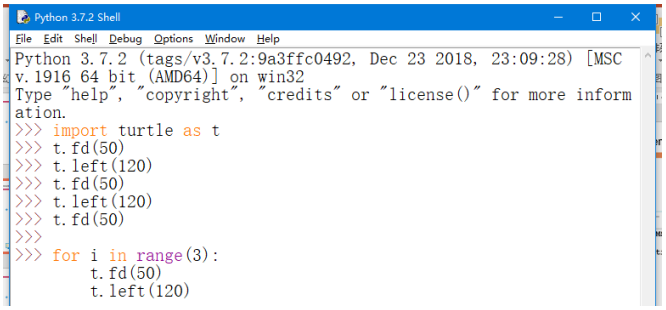
输入命令，解释器（Shell）即负责解释并执行命令

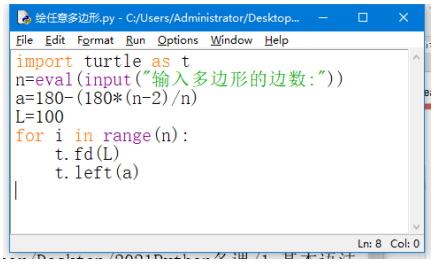
2.文件执行方式

建立程序文件，然后调用并执行这个文件

以.py 为扩展名

【例】绘制直线、三角形、正方形（命令行=>循环=>文件方式）





☺线上抢答：

程序文件扩展名？

Python 官网？　　http://www.python.org/downloads

1.4 Python 基础语法 P27

1.注释：#，’’’

2.关键字：保留字是 Python 系统内部定义和使用的特定标识符。Python 3.5.X 中共有 33 个关键字。

```
>>> import keyword
>>> print(keyword.kwlist)
 ['False', 'None', 'True', 'and', 'as', 'assert', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']
>>>
```

3. 标识符：标识符用来表示常量、变量、函数、对象等程序要素的名字。必须符合命名规则：

（1）首字符必须是字母、汉字或下划线。

（2）中间可以是字母、汉字、下划线或数字，但不能有空格。

（3）字母区分大小写（大写 S 和小写 s 代表了不同的两个名称）。

（4）不能使用 Python 的关键字。

☺线上选人：下面正确的标识符：　age_18 ,_year_2021, class, 211school

4.强制缩进：

使用缩进来表示代码块，不需要使用大括号“{}”。

缩进的空格数是可变的，但是同一个代码块的语句必须包含相同的缩进空格数。

5.一句多行：\

6. 多句一行：　**;**

☺线上抢答：

1.程序错在哪里？怎么修改才正确？

2.缩进的空格数必须为 4 个，是否正确？

1.5 turtle 库绘图

一、标准库的导入 P5

函数库又被称为模块，它是一个包含所有定义函数和变量的文件，其扩展名是.py。

函数库中的标准库和第三方库都需要先导入再调用。

导入模块的语句是 import，它有下面的三种形式：

（1）导入一个或多个模块的全部函数，格式为：

import　<模块名 1> [,<模块名 2>[,...<模块名 N>]　[as <别名>]

（2）导入某个模块的指定函数，格式为：

from　<模块名>　import <函数名 1> [,<函数名 2>[,...<函数名 N>]

（3）导入某个模块的全部函数，格式为：

from　<模块名>　import　*

说明演示 P7

i.在使用第 1 种形式导入模块后，在调用函数名前需要加上模块名做为前缀。

>>> import turtle

>>> turtle.forward(15)

ii.使用第 2 种方式和第 3 种方式导入模块后，函数名的前缀则可省略。

>>> from turtle import *

>>> forward(15)

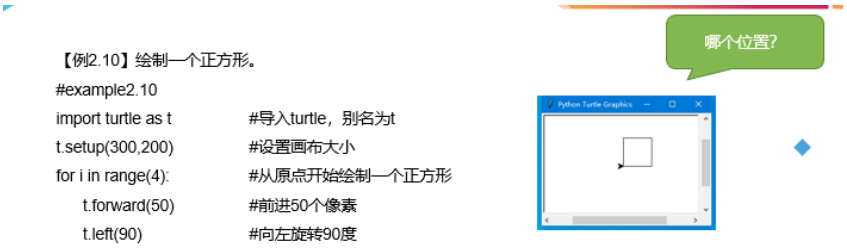
iii.为了增加程序的可读性，可以使用模块别名的方式来简化函数名的前缀。

>>> import turtle as t

>>> t.forward(15)

思考：P8

上例中绘制了一个三角形，此例绘制的是一下正方形，那么它的位置是什么？画布中心是？ 如何具体描述位置？



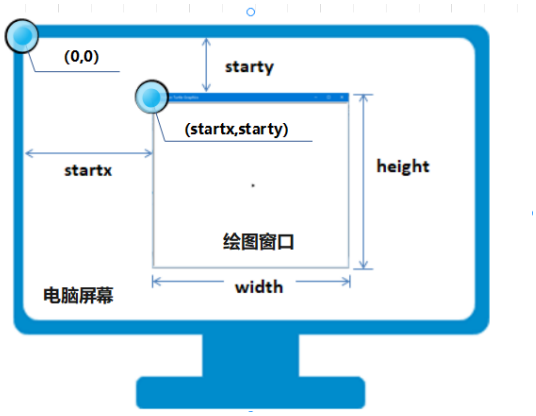
二、窗口与画布

1. 绘图窗口 P9

（1）设置窗口 P9

```
turtle.setup(width,height,startx,starty)
```

（2）位置参数 P10



说明演示：P11

```
>>> import turtle
>>> turtle.setup (200, 200, 0, 0) # 设置窗口大小为 200x200 像素，初始位置在屏幕的最左上角
>>> turtle.setup(0.75, 0.5, None, None) #屏幕宽度和高度 75%和 50%，位置居中
>>> turtle.setup()                    # 当参数都省略时，表示设置窗口为默认的初始状态
```

2.设置画布 P12

画布就是 turtle 的绘图区域。默认情况下，画布的大小为 400*300，位于窗口中心。可以使用 screensize 函数设置它的大小和背景颜色。

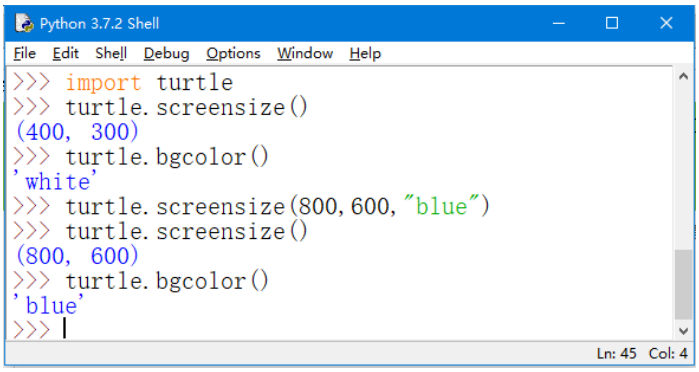
turtle.screensize(canvwidth=None, canvheight=None, bg=None)

说明：

canvwidth：正整数，表示画布的像素宽度。

canvheight：正整数，表示画布的像素高度。

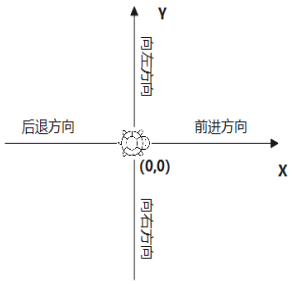
bg：颜色字符串或颜色元组，表示画布的背景颜色。



抢答：绘图窗口的初始大小是固定的吗？默认窗口的大小是多少？默认画布的大小？

默认情况下，窗口宽度为当前屏幕宽度的 50%，高度为当前屏幕高度的 75%，位置在屏幕中心。默认的绘图窗口的大小会根据当前使用的电脑屏幕分辨率而各有不同。

3. 坐标系统 P14




```
>>> import turtle as t
>>> t.pos()                    #海龟的初始位置在坐标原点(0,0)
(0.00,0.00)
>>> t.forward(150)             #沿初始的前进方向移动150像素
>>> t.pos()
(150.00,0.00)                  #海龟的当前位置坐标为（150，0）
>>>t.left(90)                  #控制海龟向左转动90度
>>> t.forward(150)             #沿当前的前进方向移动150像素
>>> t.pos()
(150.00,150.00)                #海龟的初始位置坐标为（150，150）
>>> t.home()                   #海龟回到初始位置（0，0）和方向（X轴正方向）
>>> t.pos()
(0.00,0.00)
>>> t.goto(200,200)            #海龟移动到坐标点（200，200）
>>> t.pos()
(200.00,200.00)
>>>
```

三、绘图动作与状态 P15-17

1.绘图状态与控制

turtle.pendown() | turtle.pd() | turtle.down()

turtle.penup() | turtle.pu() | turtle.up()

turtle.pensize(width=None) | turtle.width(width=None)

2.绘图动作与方向

（1）相对移动

turtle.forward(distance) | turtle.fd(distance)

turtle.back(distance) | turtle.bk(distance) | turtle.backward(distance)

（2）绝对移动

turtle.goto(x, y=None)

turtle.setpos(x, y=None) | turtle.setposition(x, y=None)

（3）相对方向

turtle.right(angle) | turtle.rt(angle)

turtle.left(angle) | turtle.lt(angle)

（4）绝对方向

turtle.setheading(to_angle) | turtle.seth(to_angle)

设置海龟方向为一个绝对角度（相对 x 轴正方向的角度），整数或浮点数。

（5）初始化海龟

turtle.home()

初始化海龟的位置和方向，海龟回到位置（0，0），方向指向 x 轴正方向。

实验课教学过程设计：

1、观看视频，课程章节 1.3.1，并完成下面的实验：（作业 1）

实验 2 基本语法：完成实验内容，并将生成的程序文件"e2.1.py"和"e2.2.py"，提交上来。

2、绘制基本图形：绘制正多边形（作业 2）

观看视频 1.4.1: 学习使用 Turtle 库绘制正方形、六边形

参照实验中代码，自由绘制一个或一组多边形（要求与实验中不一样的哦！）。

将代码和图片粘贴到，答案处，提交。

3、绘制基本图形：五角星（作业 2）

观看视频 1.4.1

参照实验中代码，绘制五角星。

将代码和图片粘贴到，答案处，提交。

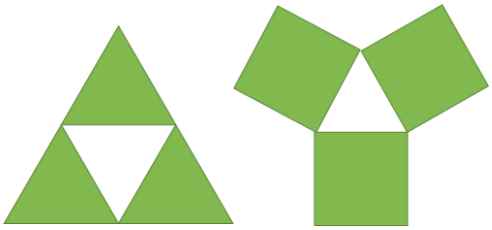
4、绘制不连续图形：（作业 2）

观看章节 1.4.2 视频教程。

绘制数字“2021”。

挑战任务：选做加分

线上讨论区回贴：挑战 2，多个图形的任意叠加效果，要求用至少三个相同图形进行叠加，效果自行设计。截图及代码回帖。



教学后记

0、课前学习通发布通知：第 1 周任务清单。让学生了解本周学习任务要求。

1、第一次课不单纯讲语法，直接引入绘制正方形程序。以兴趣引导为主“先做起来，再学起来”。

2、通过小程序让学生体会，Python 的简洁和高效，语法规则、常见错误。由于这个程序功能简单，学生理解无障碍，语法及语句格式潜移默化地被植入。

3、继续分析 turtle 库的使用，循环绘制正方形程序。图形动态的绘制过程，方便学生理解程序的执行方式、缩进、注释。

4、函数库的导入使用，相关的绘图函数，快速讲解基本的函数语句，坐标系统等。课上学生调试程序乐在其中，完成的作品五花八门。学习效果非常好。

第二周（4 学时）

教材章节：

第 1 章 Python 语言概述

1.5 Turtle 绘图

1.6 程序设计基础

教学目的和要求：

1. 掌握 turtle 库的基本绘图语句（函数）
2. 理解程序设计的基本结构 IPO
3. 掌握基本输入输出语句

教学重点

1. turtle 库的基本绘图语句
2. 问题求解的 IPO 结构
3. 基本输入输出

教学难点

1. 基本输入输出函数 Input()、print()、eval()
2. 使用 IP
3. O 结构进行问题的抽象和求解

教学方法与手段

理论课利用学习通开展线上活动与学生的互动

实验课利用超星线上资源，开展线上自主学习，教师进行具体的辅助指导

理论课教学过程设计

前课小练：

线上投票：2.X　3.X 兼容吗？

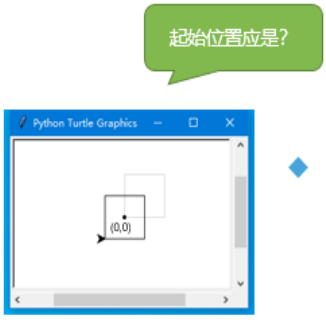
线上选人：下面语句的运行结果？　　关键字的命名规则，区别大小写

```
>>>x=100
>>>y=200
>>>z=X+y
```

1.5 turtle 绘图

【例2.11】绘制一个以原点为中心，边长为50的正方形。

```
#example2.11
import turtle as t          #导入函数库，并设置别名为t
t.setup(300,200)            #设置窗口大小为300*200
t.pensize(2)                #设置画笔线条宽度为2像素
.....
```



思考：P18，起始位置应该在哪里？补全代码？

```
#example2.11
import turtle as t    #导入函数库，并设置别名为 t
t.setup(300,200)      #设置窗口大小为 300*200
t.pensize(2)          #设置画笔线条宽度为 2 像素
t.penup()             #设置画笔抬起，非绘图状态
t.goto(-25,-25)       #移动海龟至绝对位置（-25，-25）
t.pendown()           #设置画笔放下，进入绘图状态
for i in range(4):    #绘制正方形
        t.forward(50)
        t.left(90)
=>>>>>>>>>>>>>>>>扩展程序
绘制填充效果的：（color,begin_fill()）
import turtle as t
t.setup(300,200)
t.pensize(2)
t.color("blue","red")
t.penup()
t.goto(-25,-25)
t.pendown()
t.begin_fill()
for i in range(4):
        t.forward(50)
        t.left(90)
t.end_fill()
```

本课内容：

1.6 问题求解的程序结构

1.6.1 IPO 结构

高级语言的程序都是用来求解特定问题的，而问题的求解都可以归结为计算问题。利用计算机解决计算问题时，一般情况下，程序都遵循数据输入（Input）、数据处理（Process）、数据输出（Output）的基本结构。这种程序结构正是反映了实际问题的计算过程。

输入数据：输入（Input）是一个程序的开始。程序要处理的数据有多种来源，形成了多种输入方式，包括：文件输入、网络输入、控制台输入、交互界面输出、随机数据输入、内部参数输入等。

处理数据：处理（Process）是程序对输入数据进行计算产生输出结果的过程。计算问题的处理方法统称为“算法”，它是程序最重要的组成部分。可以说，算法是一个程序的灵魂。

输出数据：输出（Output）是程序展示运算成果的方式。程序的输出方式包括：控制台输出、图形输出、文件输出、网络输出、操作系统内部变量输出等。

【例2.5】求一元二次方程ax2+bx+c=0的实根。

```
# example2.5
a=int(input("请输入a：  "))   #输入a
b=int(input("请输入b：  "))   #输入b
c=int(input("请输入c：  "))   #输入c
d=b**2-4*a*c
if d>=0:                      #判断是否有实根
    x1=(-b+d**0.5)/(2*a)
    x2=(-b-d**0.5)/(2*a)
    print('方程的根：  x1=%f,x2=%f'%(x1,x2))
else:
    print('input data Error！ ')
```

数据输入（Input）

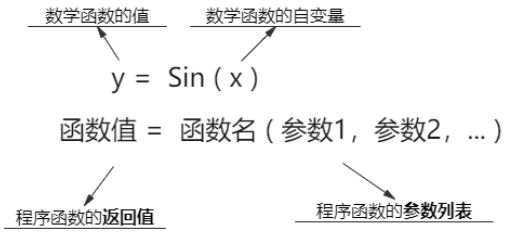
数据处理（Process）

数据输出（Output）

1.6.2.函数是什么

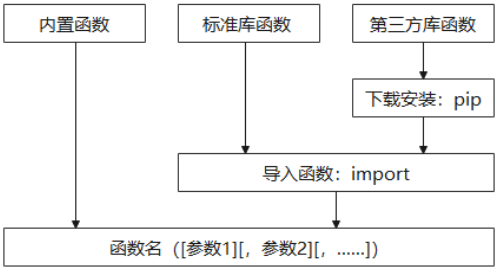
在上面的案例中用到了几个函数：input()、print()、eval()和 int()。那么，程序中的函数是什么呢？

高级语言中的函数就是一段代码的封装，用来实现某种特定的运算或功能。使用者不用关心这些运算和功能的具体实现细节，只要调用这些函数，就可以进行运算或完成想要的功能。程序中函数的调用与数学函数中方法类似，通过函数名并且给定参数的方式。函数概念的解析：



Python 的函数有三种：内置函数、标准库函数、第三方库函数。

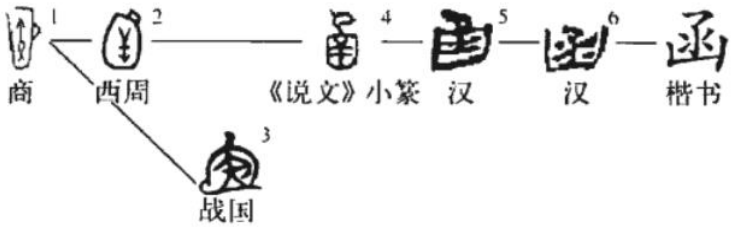
内置函数是系统自带的可以直接使用的函数，如案例中的 input、print、eval 等都是内置函数。标准库函数是系统自带的外部函数，需要先用 import 导入函数库后才能使用，如，turtle 是标准函数库。第三方库函数由相关人员或机构开发，需要先进行函数库的安装，再导入后才能使用。如，pyecharts 就是第三方函数库。函数的调用过程如图：



知识扩展：“函”字的演变

“函”字始见于商代甲骨文，它的字形很像一个袋子里装着一支箭的形状，袋子上还有一个便于手拿或挂在腰上的提手或挂钩。“函”的本义即箭袋，泛指包物的东西，又特指包信等物的封套。

函数中的“函”就是取其封装之意，程序中的函数是指一段代码的封装。封装（Encapsulation）是面向对象程序设计方法的重要原则，就是把对象的属性和操作结合为一个独立的整体，并尽可能隐藏对象的内部实现细节。



1.6.3.输入函数 input()

函数的语法格式

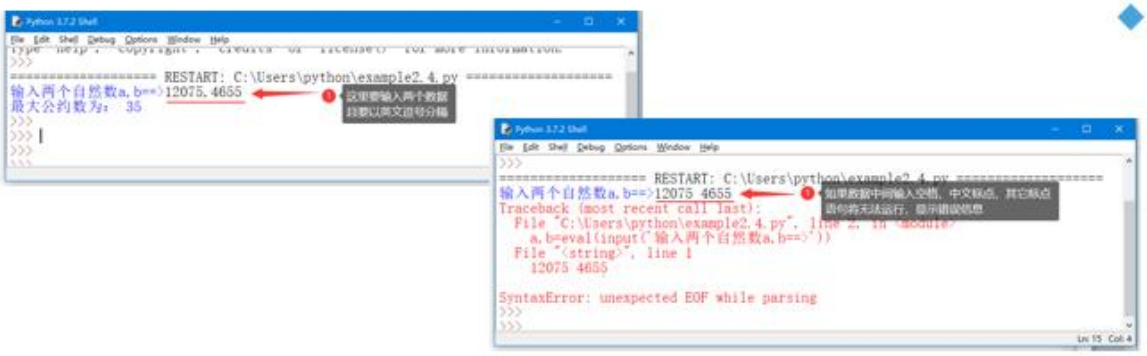
input 是输入函数，用来读取用户输入的数据，并返回一个字符串，具体格式如下：

```
<变量名>=input([“提示信息”])
```

同时为多个变量赋值时

例如：在【例 2.4】中，输入两个自然数的语句：

```
a,b=eval(input('输入两个自然数 a，b：'))
```



演示：x+y

使用函数转换数据类型

```
>>> x=input('input x=>')
input x=>5
>>> y=input('input x=>')
input x=>20
>>> print(x+y)
520
>>>
```

```
>>> x=int(input('input x=>'))
input x=>5
>>> y=val(input('input x=>'))
input x=>10
>>> x+y
15
>>>
```

1.6.4.输出函数 print()

1.函数语法格式

print 是输出函数，用来输出表达式的值，具体格式如下：

```
print(value1, value12,...,sep=' ',end='\n')
```

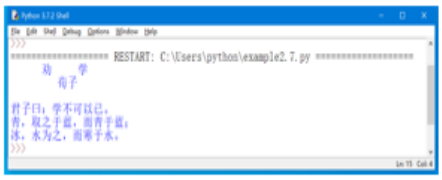
【例 2.6】print 的测试程序

2. 控制数据输出的样式

（1）分行显示字符串

【例 2.7】两种字符串分行显示的方法示例。

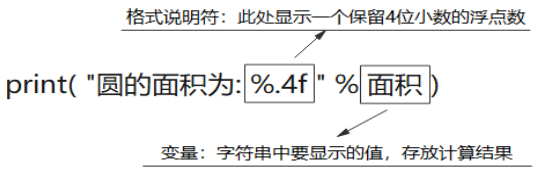
```
#example2.7 荀子之劝学
劝学="    劝    学\n      荀子"
劝学1="'
君子曰：学不可以已。
青，取之于蓝，而青于蓝；
冰，水为之，而寒于水。'"
print(劝学)
print(劝学1)
```



思考：代码与输出的对应关系？\n 的作用？

（2）设置数据的小数位数

【例 2.8】计算圆的面积并保留 4 位小数。



抢答：代码与输出的对应关系？

思考 ：程序中的 IPO？ 常量、变量，表达式？ 引入概念

1.6.5 转换函数 eval()

1.函数语法格式

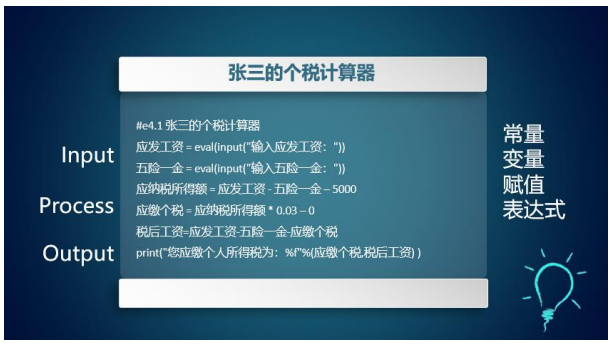
eval 函数用来执行一个字符串表达式，并返回表达式的值。具体格式如下：

```
eval(表达式[,globals[,locals]])
```

【例 2.9】简单公式计算器。

随堂练习：输入输出函数

思考：**IPO 结构实例：**个税计算器，理解常量变量，赋值与表达式。



思考：代码中常量和变量有哪些？有哪些数据类型？表达式是什么？

线上自主学习任务：

观看章节 1.5.1 中的视频，完成递进式任务：实验四：程序设计结构—IPO

作业 3：

1、 个税计算器：输入给出的代码，运行无误后保存为“实验 4.1.py”并提交。

2、 复利计算器：将代码补充完整，运行无误。代码保存为“实验 4.2.py”提交。

3、 根据图 4.7，分析程序结构，新建文件，编写代码。实现“古尺计算器”功能：程序保

存为“实验 4.3.py”并提交。

挑战任务：IPO 结构实例

线上讨论区回贴：挑战 3。输出一个整数 n，程序自动绘制一个的正 n 边形。如何实现？

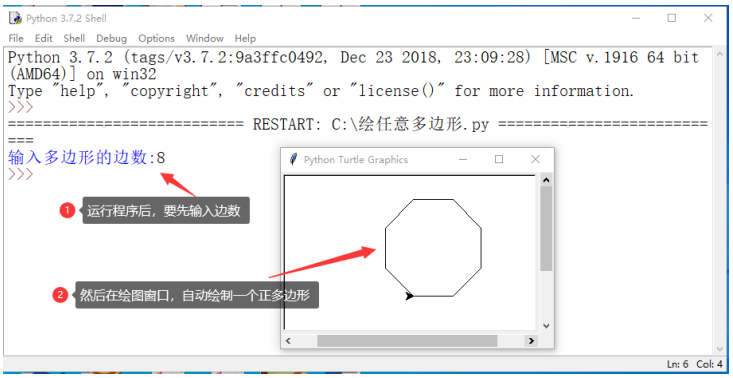
分析程序结构

I: 输入多边形的边数

P: 导入函数库，调用函数

O: 绘制任意正多边形

编写代码实现如下的功能：



教学后记

1. 课前学习通发布通知：第 2 周任务清单。让学生了解本周学习任务要求
2. 上一周的挑战任务，同学们的发帖积极，有几个同学的作品相当惊艳，有时间可以展示一下。
3. 通过上周内容的扩展，复习旧知引入新知。讲解绘图的颜色、填充，引入程序的 IPO 结构，函数的概念，常量、变量及表达式。
   绘制以原点为中心的正方形=>画布的坐标系统=>设置颜色和填充效果=>turtle 函数的参数=>参数的数据类型=>在任意坐标位置绘制正方形？=>输入与输出=>IPO 结构=>输入输出函数=>程序实例
4. 本周由于评卷，连上四节理论。同学们带电脑过来的，有布置实操的任务，大家参与度较好。
5. 博文楼 wifi 网络不理想，学习能发布线上活动和随堂练习时网络卡顿。

第三周（4 学时）

教材章节

第 2 章 数据类型和表达式
2.1 基本数据类型
2.2 常量和变量
2.3 运算符与表达式
2.4.1 常用内置函数

教学目的和要求

理解常量变量、数据类型和表达式
掌握常用内置函数的使用方法

教学重点和难点

1、 数据类型与表达式
2、 常用内置函数功能：数值运算函数、数据转换函数、类型转换函数 t

教学方法与手段

理论课利用学习通开展线上活动与学生的互动

实验课利用超星线上资源，开展线上自主学习，教师进行具体的辅助指导

理论课教学过程设计

前课小练：

☺随堂练习：输入输出函数与 turtle

☺抢答：常量，变量，表达式，IPO，函数

☺抢答：实例中如何输出浮点数保留小数的方法？

【实例 1】 张三的个税计算器

```
个税起征点 = 5000
应发工资 = eval(input("输入应发工资："))
五险一金 = eval(input("输入五险一金："))
应纳税所得额 = 应发工资 - 五险一金 - 个税起征点
if 应纳税所得额<=3000:    ##级数 1
        税率=0.03
        速算扣除数=0
elif 应纳税所得额>3000 and 应纳税所得额<=12000:    ##级数 2
        税率=0.1
        速算扣除数=210
应缴个税 = 应纳税所得额 *税率 - 速算扣除数
税后工资=应发工资-五险一金-应缴个税
print("您应缴个人所得税为：%f )
```

【实例 2】#e4.2 复利计算器

```
本金=eval(input("您的本金:"))
```

```
年利率=eval(input("年利率:"))
年期=eval(input("存期(年):"))
最终收益=本金*(1+年利率/100)**年期
print("最终收益：%f"%(最终收益) )
```

本课内容：

第 2 章 Python 数据类型和表达式

2.1 基本数据类型

1. 整型（int）：十进制，二进制整数 0B100，八进制 0O67，十六进制 0Xff
2. 浮点型（float）：十进制小数表示法 3.14、10.0，科学计数表示法 314.159=>3.14159e2
3. 字符串类型：单引号、双引号、三引号
4. 布尔类型：True，False

演示：type()，int，float，bool，str

```
>>> a=256                #int
>>> b=0b1010             #int
>>> c=355/113            # float
>>> d=True               #bool
>>> e="0.6180339887"        #str
>>> type(a)
<class 'int'>
>>> type(c)
<class 'float'>
>>> type(d)
<class 'bool'>
>>> type(e)
<class 'str'>
```

思考：为什么要有不同的数据类型？

```
>>> import sys
>>> sys.float_info.max
1.7976931348623157e+308
>>> sys.float_info.min
2.2250738585072014e-308
>>>
```

知识扩展：**有意思的浮点数。**

1.圆周率

圆周率的历史：1500多年前，南北朝时期的祖冲之计算出圆周率π的值在3.1415926和3.1415927之间，并且得出了两个用分数表示的近似值：约率为22/7，密率为355/113。

圆的周长与直径之比是一个常数，人们称之为圆周率。通常用希腊字母 π 来表示。

我国古代数学家作出了巨大的贡献，在东汉初年的数学书《周髀（毕）算经》里已经载有“周三径一”，称之为“古率”，就是说，直径是 1 的圆，它的周长是 3。《周髀算经》算经的十书之一，是中国最古老的天文学和数学著作，成书于公元前 1 世纪的西汉末或东汉初年。

西汉末年，刘歆（约分元前 50 年到公元 23 年）定圆周率为 3.1547

东汉时代，张衡（公元 78－139 年）求得约等于 5/8，3.1622

三国时，魏人刘徽（公元 263 年）用割圆术求得圆周率的前三位数字是 π≈3.14，称为徽率

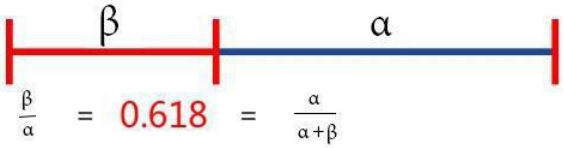
南北朝，祖冲之（公元 480 年）已推算出 3.1415926＜π＜3.1415927，他是世界上第一个将圆周率精准到 7 位小数的人。祖冲之又提出了用两个分数表示 π 的近似值，即 22/7 及 355/113，分别称为 π 的约率和密度。

在祖冲之发现密率一千多年后，欧洲的安托尼兹(1625 年)才重新发现了这个值。

2019 年 3 月 14 日，谷歌宣布圆周率现已到小数点后 31.4 万亿位

2.黄金分割

黄金分割是指将整体一分为二，较大部分与整体部分的比值等于较小部分与较大部分的比值，其比值是（√5-1）：2，近似值为 0.618，通常用希腊字母Φ表示这个值。这个比例被公认为是最能引起美感的比例，因此被称为黄金分割。



黄金分割具有严格的比例性、艺术性、和谐性，蕴藏着丰富的美学价值，这一比值能够引起人们的美感，被认为是建筑和艺术中最理想的比例。

画家们发现，按 0.618:1 来设计的比例，画出的画最优美，在达·芬奇的作品《维特鲁威人》、《蒙娜丽莎》还有《最后的晚餐》中都运用了黄金分割。而现今的女性，腰身以下的长度平均只占身高的 0.58，因此古希腊的著名雕像断臂维纳斯及太阳神阿波罗都通过故意延长双腿，使之与身高的比值为 0.618。建筑师们对数字 0.618 特别偏爱，无论是古埃及的金字塔，还是巴黎的圣母院，或者是近世纪的法国埃菲尔铁塔，希腊雅典的巴特农神庙，都有黄金分割的足迹。埃菲尔铁塔以观景台为分割点，其中下方与上方高度的比值就是黄金分割的常数。

2.2 常量与变量

常量：在程序运行过程中，其值不发生改变的数据对象称为常量。

变量：在程序运行过程中，可以随着程序的运行更改的量称之为变量。

变量的声明：Python 变量的**赋值操作即是变量的声明和定义的过程**。

变量的命名：变量命名要符合标识符的命名规则。

变量的赋值：一般形式、增量形式、链式赋值、多重赋值。P22

演示：Python 变量未经赋值就使用，解译器会提示错误。

2.3 运算符与表达式 P25

1.算术运算符：+-*/，%取模，**乘方，//整除

2.关系运算符：==，！=，>，<，>=，<=

3.赋值运算符:=,+=,-=,*=,/=,%=,**=,//=

4.逻辑运算符:and, or, not

5.表达式：运算优先级，表 2.8，P29

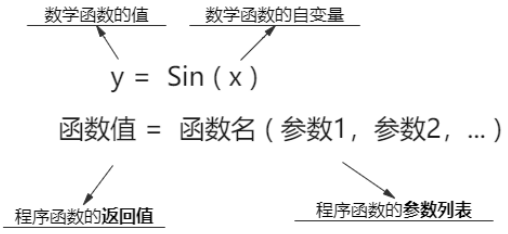
```
>>> not "Abc"=="abc" or 2+3!=5 and "23"<"3"
True
>>>
```

练习题：数据类型、运算符与表达式

2.4.1 常用内置函数 P30

1. 函数功能（表 2.9 内置函数）

   函数的使用：函数名，参数个数，参数类型



2. 函数实例

（1）数值运算函数 P32

abs(x)函数：返回一个 x 的绝对值，x 参数可以是整型或浮点型数据。

divmod(a,b)：返回由 a / b 的商和余数构成的一个元组，a、b 可以是整型或浮点型。

pow(x, y[, z])：返回 x 的 y 次幂，如果有参数 z 则返回 x 的 y 次幂与 z 的模。

pow(x, y, z)➔ pow(x, y) % z

pow(x, y)➔x**y

round(x[, n])：对 x 进行四舍五入，n 保留的小数位数，若省略则只保留整数。

```
>>> print(abs(10),abs(-10))
10 10
>>> print(divmod(10,4),divmod(10.5,2.5))
(2, 2) (4.0, 0.5)
>>> print(pow(2,3),pow(2,3,4))
8  0
>>> print(round(3.1415926,3),round(3.1415926))
3.142 3
>>>
```

【实例】构造表达式：海伦公式，实训 P29

若三角形的边长 a，b, c 分别为 3，4，5 。

p 为周长的一半，即，$p=\frac{a+b+c}{2}$。

则，三角形的面积 S，$S=\sqrt{p(p-a)(p-b)(p-c)}$。

（2）数制转换函数 P33

int([x])：将十进制数值取整（截去小数部分）。

```
>>>int(3.14)                # 3
```

```
>>>int(2e2)                    # 200
>>>int(100, 2)                 # 出错，base 被赋值后函数只接收字符串
```

int([x[, base]])：将“base”进制的合法整数字符串转换成十进制整数。

```
>>>int('23')                   # 23，base 参数省略时默认为十进制数
>>>int('23',16)                # 35，将'23'做为 16 进制数
>>>int('Pythontab',8)          # 出错，'Pythontab'不是个 8 进制数
```

字符串 0x 视作十六进制的符号，0b 视作十六进制的符号：

```
>>>int('0x10', 16)             # 16，0x 是十六进制的符号
>>>int('0b10',2)               # 2 ，0b 是二进制的符号
```

bin(x)：将十进制整数转换成二进制的数字字符串。

oct(x)：将十进制整数转换成八进制的数字字符串。

hex(x)：将十进制整数转换成十六进制的数字字符串。

```
>>> b=255
>>> print(bin(b),hex(b),oct(b))
0b11111111  0xff  0o377
>>> print(int('123', 16),int('123',8),int('123'))
291 83 123
>>>
```

【实例】进制转换器：它进制转十进制，实训 P29

（3）类型转换函数

bool([x])：返回 x 布尔值。

float([x])：返回一个浮点数。

ord()：返回单个字符的 Unicode(ASCII)码，返回值是一个整数。

chr(i)：返回整数 i 对应的字符。与 ord()函数互为反函数。

```
>>> x='A'
>>> y=97
>>> ord(x)
65
>>> chr(y)
'a'
>>> chr(ord(x))
'A'
>>>
```

思考：字符的比较？汉字的比较？ord()返回字符的 unicode(ASCII)，”0”<”a”<”A”<”安”

```
>>> ord("A")
65
>>> ord("a")
97
>>> ord("男")
30007
```

```
>>> ord("女")
22899
>>> "男">"女"
True
>>>
```

eval (str [,globals[,locals]])：将字符串 str 当成有效的表达式来求值并返回计算结果。

```
>>> x = 1
>>> eval('x+1')
2
>>> a='5,10'
>>> x,y=eval(a)
>>> print(x,y)
5 10
>>>
```

实验课教学过程设计：

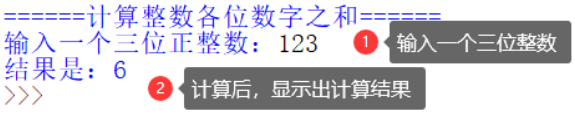
观看章节 2.4.1 中的视频“实验五：表达式与内置函数”

完成作业 4：

1. 面积计算器: 新建文件，运行无误后保存为“实验 5.1.py”并提交。
2. 进制转换器（十转它）：将代码补充完整，运行无误。代码保存为“实验 5.2.py”提交。
3. 进制转换器（它转十）：将代码补充完整，运行无误。代码保存为“实验 5.3.py”提交。

挑战任务

挑战任务 4：根据下图，分析程序结构，新建文件，编写代码。程序功能为：输入一个任意的三位正整数，计算并输出其各位数字之和。



挑战任务 5：已知，位移 S，初速度 V_0，时间 t，加速度 a，计算公式为：

$$S=V_0t+at^2/2$$

要求：程序运行后，依次输入位移、初速度和时间，程序可以计算并输出加速度。
测试数据：输入：距离 100、初速度 6、时间 10，输出：加速度 0.8。

教学后记

第四周（4 学时）

教材章节：

第 3 章 Python 控制语句
3.1 结构化程序设计
3.2 分支结构
2.4.2 常用标准函数-random

教学目的和要求：

1、了解程序结构的基本概念
2、掌握分支结构的基本语句
3、掌握随机函数的使用

教学重点和难点

1、 结构化程序设计方法的基本思想
2、 逻辑功能设计和数据流关系
3、 分支结构（选择结构）
4、random 模块的常用函数的使用

教学方法与手段

理论课利用学习通开展线上活动与学生的互动
实验课利用超星线上资源，开展线上自主学习，教师进行具体的辅助指导

理论课教学过程设计

前课小练：国王的债务。

传说国际象棋是舍罕王的宰相西萨.班.达依尔发明的。他把这个有趣的娱乐品进贡给国王。舍罕王对于这一奇妙的发明异常喜爱，决定让宰相自己要求得到什么赏赐。

西萨并没有要求任何金银财宝，他只是指着面前的棋盘奏道："陛下，就请您赏给我一些麦子吧，它们只要这样放在棋盘里就行了：第一个格里放一颗，第二个格里放两颗，第三个格里放四颗，以后每一个格里都比前一个格里的麦粒增加一倍。圣明的王啊，只要把这样摆满棋盘上全部六十四格的麦粒都赏给你的仆人，他就心满意足了"。

```
一粒小麦的重量约为(g):0.02
全世界的小麦年产量为(亿吨)：4.8
国王的债务为：9223372.03685478亿吨！！！

约为全世界192.15358万年的产量！
>>>
```

已知，假设一粒小麦的重量为 0.02g：
步骤 1：确定 IPO，
输入：全世界小麦的年产量；
处理：计算国王的债务；

输出：国王需要多少年还清债务？

步骤 2：分析问题的计算部分，构造计算表达式。

步骤 3：编写程序，调试、运行程序。

程序代码：

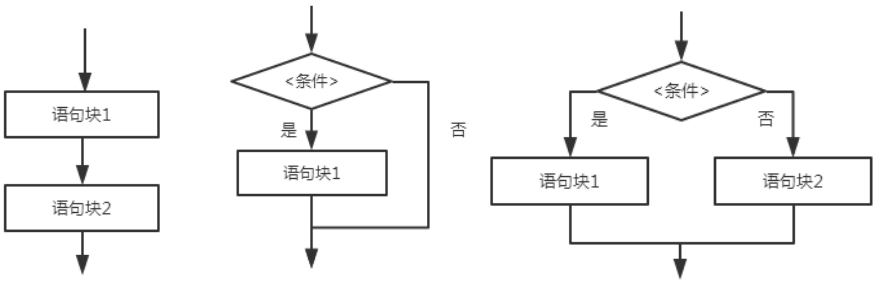
```
#国王的债务
onerice=eval(input("一粒小麦的重量约为(g):"))
production=eval(input('全世界的小麦年产量为(亿吨)：'))
rice=pow(2,64)-1
weight=rice/onerice/10**6/10**8        #亿吨
year=weight/production/10**4           #万年
print('国王的债务为：{:.8f}亿吨！！！\n'.format(weight))
print('约为全世界{:.8}万年的产量！'.format(year))
```

注意：

1. input 函数返回输入的字符串，若数学运算，则需要 eval 函数。
2. 程序中的公式，可以采用函数或运算符来设计。

本课内容：

3.1 结构化程序设计



顺序结构：程序从第一行语句开始执行，执行到最后一行语句结束，程序中的每条语句都会被执行一次。

3.2 分支结构：

也称选择结构，表示程序的处理步骤出现了分支，它需要根据某一特定的条件选择其中的一个分支执行。

1. 单分支结构

Python 中单分支 if 语句的语法格式如下：

```
if    <条件>:
      <语句块>
```

【例 1】 输入两个数字，输出其中较大数字。P46，绘制出流程图。

```
#example4.1
x=eval(input("请输入第一个数字:"))
y=eval(input("请输入第二个数字:"))
print("输入的两个数字为：",x,y)
if x>y:
    print("较大的是:",x)
```

```
if x<y:
    print("较大的是:",y)4. 双分支结构
```

2. 双分支结构

是使用比较多的一种程序结构，Python 中双分支 if 语句的语法格式如下：

```
if  <条件>:
    <语句块 1>
else:
    <语句块 2>
```

【例 2】 用双分支结构改写【上例】。P47

```
#example4.2
x=eval(input("请输入第一个数字:"))
y=eval(input("请输入第二个数字:"))
print("输入的两个数字为：",x,y)
if x>y:
    print("较大的是:",x)
else:
    print("较大的是:",y)
```

思考：例 2 与例 1 是否完全相同？绘制此例的流程图。

3. 多分支结构

多分支结构是对双分支结构的一种补充，当判断的条件有多个，结果也有多种情况的时候，可用多分支 if 语句进行判断。多分支结构语法格式如下：

```
if  <条件 1>:
    <语句块 1>
elif <条件 2>:
    <语句块 2>
elif <条件 3>:
    <语句块 3>
…
else:
    <语句块 n>
```

程序执行时，按照条件序列从上向下进行判断，当第一个条件 i 的值为 True，就执行该条件下的语句块，然后整个多分支 if 结构结束。如果没有任何条件为 True,则执行 else 下的语句块。注意：语句中的 else 是可选的。

【例】输入一个分数，判断它应该的到的学分绩点。

```python
#example4.4.1
score = eval(input("请输入分数："))
if score >= 90:
    gpa = 4
elif score >= 80:
    gpa = 3
elif score >= 70:
    gpa = 2
elif score >= 60:
    gpa = 1
else:
    gpa = 0
print("应得学分绩点为：",gpa)
```

```python
#example4.4.2
score = eval(input("请输入分数："))
if score <60:
    gpa = 0
elif score >= 60:
    gpa = 1
elif score >= 70:
    gpa = 2
elif score >= 80:
    gpa = 3
else:
    gpa = 4
print("应得学分绩点为：",gpa)
```

思考：多分支结构的流程图？

【例】分支结构嵌套 P51

3.4.1 random 模块

随机数可以用于数学、游戏以及安全等领域，还经常被嵌入到算法中，用以提高算法效率，并提高程序的安全性。Python 的常用随机数函数如表 3.11 所示：

表 3.11 random 模块常用函数功能说明

函数	描述
seed([a])	初始化随机数生成器的种子，默认为系统时间。需要在生成随机数之前调用此函数。
random()	生成一个在[0.0,1.0)之间的随机浮点数
randint(a,b)	生成一个[a,b]之间的随机整数，其中 a<=b。
randrange ([start,]stop [,step])	生成一个[start,stop)，以 step 为步长的范围内的随机整数，step 默认值为 1，等价于 choice(range(m,n,step))。
uniform(a,b)	生成一个[a,b]之间的随机浮点数，a 可以大于 b
choice(seq)	从非空序列 seq 中随机挑选一个元素
sample(seq,k)	从序列 seq 中随机获取 k 个元素，并返回一个新序列
shuffle(seq)	将序列 seq 的所有元素随机排序，并修改原序列
getrandbits(k)	生成一个 k 比特长的随机整数

函数的应用实例：

```
>>> from random import *          #导入 random 模块的所有函数
>>> random()                      #生成一个在[0.0,1.0)之间的随机浮点数
0.14836545714990013
>>> randint(1,10)                 #生成一个[1，10]之间的随机整数
5
>>> x=[1,2,3,4,5]                 #x 赋值为一个序列
>>> sample(x,2)                   #从 x 序列中随机挑选 2 个元素生成新序列
[5, 1]
>>> shuffle(x)                    #将 x 序列的元素随机排序
>>> x
```

```
[2, 5, 4, 3, 1]
>>>
```

random 是如何生成随机数的呢？Python 的随机数是用确定性的算法计算出来在[0.0,1.0)范围均匀分布的随机数序列，它并不是真正的随机被称为伪随机数。伪随机数具有类似于随机数的统计特征，如均匀性、独立性等。因此在模拟研究中可以采用伪随机数代替真正的随机数，从而提高模拟效率。

在计算伪随机数时，若使用的初值（种子）不变，那么伪随机数的序列也不变。例如，当随机种子为 2 时，生成的随机数序列如图 3.1 所示。

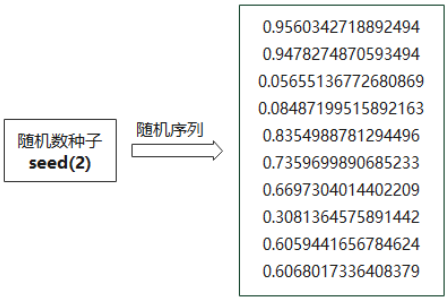


图 3.1 随机种子为 2 时的随机序列

当种子不变时，random 函数将在此序列中按顺序依次返回随机数。如执行下面语句：

```
>>> from random import *
>>> seed(2)                    #随机种子为 2
>>> random()
0.9560342718892494             #随机序列中第一个数
>>> random()
0.9478274870593494             #随机序列中第二个数
>>> random()
0.05655136772680869            #随机序列中第三个数
>>> seed(2)                    #重新生成随机序列
>>> random()
0.9560342718892494             #新生成的随机序列的第一个数
>>>
```

在默认的情况下，当未设置种子时，random 将以系统时间作为种子，这时产生的随机序列就是随时变化的了。那么，什么时候会使用 seed 函数呢？当我们的程序希望生成的随机数能够复现的时候，就需要设置 seed 函数了，此时生成的随机数就是固定的。

实验课教学过程设计：

观看章节 2.4.1 中的视频，完成实验五：表达式与内置函数

4. 面积计算器: 新建文件，运行无误后保存为“实验 5.1.py”并提交。
5. 进制转换器（十转它）：将代码补充完整，运行无误。代码保存为“实验 5.2.py”提交。
6. 进制转换器（它转十）：将代码补充完整，运行无误。代码保存为“实验 5.3.py”提交。

挑战任务

温度转换器：摄氏与华氏温度转换。

1. 通过完成问题求解的 IPO 设计。.温度转换

问题：如何利用 Python 程序进行摄氏度和华氏度之间的转换

步骤 1：分析问题的计算部分：采用公式转换方式解决计算问题

步骤 2：确定功能

输入：华氏或者摄氏温度值、温度标识

处理：温度转化算法

输出：华氏或者摄氏温度值、温度标识

F 表示华氏度，82F 表示华氏 82 度

C 表示摄氏度，28C 表示摄氏 28 度

步骤 3：设计算法，根据华氏和摄氏温度定义，转换公式如下：

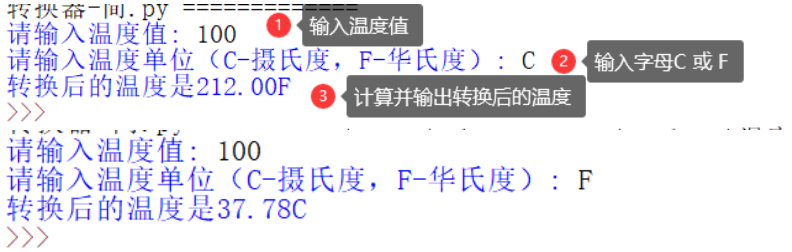
C = (F － 32) / 1.8

F = C * 1.8 + 32

其中，C 表示摄氏温度，F 表示华氏温度

步骤 4：编写程序。

步骤 5：调试、运行程序。使用 IDLE 打开上述文件，按 F5 运行（推荐），输入数值，观察输出



教学后记