

Python 第2天

作业评讲

1. 三角形判断

- 分析：先判断是否能组成三角形，再判断具体是什么三角形

2. 素数判断

- 素数（质数）：如果某一个正数只能被1和其自身整除，则该数字是素数。规定：1不是素数
- 分析：用一个数从2开始，不断的去尝试除number，如果都除不尽（最多尝试到number-1），则number是素数，如果有任意一个其间数能把number除尽，则number不是素数（也就终止尝试了）

一、循环结构（while、for）

1. 循环主体语句中的控制关键字（一定要在循环的主体中用）

- `break`：立即打断整个循环结构，跳到循环之外，后面的语句。
- `continue`：立即结束本轮循环，继续下一轮循环的条件判断。

2. while循环

3. for遍历（循环）

- 本质：依次从容器（一组数据）中，访问其中的每一个元素。
- 语法

```
for 变量名 in 容器:
    循环遍历的主体语句，可以使用变量名
else:
    当容器中没有元素可取了，遍历结束时，执行此语句（可省略的）
    一般，当因break退出循环遍历时，就不执行次语句
```

- 特点：

- 可以使用for和range函数搭配
- ```
for 变量名 in range(...):
```

 循环遍历的主体语句，可以使用变量名
- 数据的生成靠range函数，遍历数据靠for

- 示例：

```
for b in ("西游记", "红楼梦", "水浒传", "三国演义"):
 print("《" + b + "》", end=" ")

for b in [89, 93, 78, 96, 52, 91]:
 print(str(b) + "分", end=" ")
```

```

for stu in {"s1013": "张无忌", "s1019": "赵敏", "s2001": "周芷若"}:
 print(stu, end=" ")

 # 传入2个参数时候：前闭后开
for n in range(5, 21):
 print(n, end=" ")
print()

传入1个参数的时候：产生0到这个参数-1的一组数字
for n in range(10):
 print(n, end=" ")

传入3个参数的时候：第3个参数是“步长（step）”
for n in range(10, 40, 1):
 print(n, end=" ")

for n in range(10, 40, 3):
 print(n, end=" ")

第3个参数为负数，第1个参数必须大于第2个参数：从大到小了
for n in range(40, 10, -1):
 print(n, end=" ")

```

## 二、python中的常用的函数

### 1. 内置函数（不需要导入）

- print、input
- int、str、float、bool、type
- round
- len、sorted
- abs、max
- 示例

```

内置函数直接使用，无需导入
len：计算容器或字符串的长度
print(len([87, 98, 43, 78, 56, 73, 91]))
print(len("我太爱你们了，你们微信的讨论，我看了有时候想哭。"))

abs：计算数字的绝对值
print(abs(13), abs(-13), abs(7.53), abs(-7.53))

max：从多个参数的数字中，返回最大值
print("最高分数为：", max(98, 73, 96, 59, 100, 67, 92))

```

### 2. 非内置函数，需要导入

- 导入语法（推荐导入代码写在一个文件的顶部）
  - 从模块中导入函数：`from 模块名 import 函数名`
  - 从模块中导入多个函数：`from 模块名 import 函数名1, 函数名2`

- 从模块中导入函数并给函数起别名（不推荐）：`from 模块名 import 函数名 as 别名`
- 导入模块：`import 模块名`
  - 导入后，使用 `模块名.函数名(...)` 来调用函数
- 导入模块并给模块起别名：`import 模块名 as 别名`
  - 导入后，使用 `别名.函数名(...)` 来调用函数
- 可以从模块中导入各种元素：函数、变量、类

## • 常用的函数

### ◦ 随机数模块 ( random ) 中的常用函数

- `random()` —— 生成一个0到1之间的随机浮点数（不能达到1）
- `randint(50, 70)` —— 生成一个范围之间的随机整数（两边都包含）
- `choice(容器)` —— 返回容器中的一个随机的元素

### ◦ 时间模块 ( time ) 中的常用函数

- `sleep(3.5)` —— 让程序休眠3.5秒
- `time()` —— 返回系统当前的时间戳（一个以秒为单位的浮点数）
  - 是1970年1月1日0点0分0秒开始计算，到当前经过了多少秒
- `strftime('%Y-%m-%d')` —— 返回格式化后的系统当前时间
  - 日期时间的占位符：
    - %Y (年)、%m (月)、%d (日)
    - %H (时)、%M (分)、%S (秒)

## 三、模块(module)和包 ( package )

- 一个python文件，就是一个模块，文件名称（除去.py）被称为模块名
- 当我们的某个模块（文件）中的代码，使用了另一个模块（文件）中的元素（函数、变量、类）时，需要导入
- 包
  - 分门别类的管理python文件
  - 包就是文件夹
  - 包中，建议永远都有一个 `__init__.py` 文件，它会在别的模块导入这个包的时候，被自动执行。
  - 如果要调用的函数、类、变量所在的模块，在一个包中，要连包名一起出现在导入的语句中
    - 语法示例：

```
from 包名.模块名 import 函数, 变量, 类
from 包名 import 模块名

from 包名1.包名2.模块名 import 函数, 变量, 类
from 包名1.包名2 import 模块名
```

## 四、自定义函数

### 1. 定义函数的基本语法

- 语法

```
def 函数名(可选的参数):
 函数的主体语句1
 函数的主体语句2
 函数的主体语句3
 可能有返回值语句
```

- 函数命名规则：
  - 只能出现字母、数字、下划线，数字不打头
  - 不能使用系统关键
  - 建议全小写
  - 建议命名有含义

## 2. 定义函数的注意点

- 定义函数，并不是执行函数主体的语句，只是加载。
- 执行到定义函数的语句时，只是做计划，并没有实施。
- 函数的主体语句不能为空，如果没有什么可写的，至少要有pass占位符。
- 分角色：
  - 定义者
  - 调用者

## 3. 函数的返回值

- 函数主体可以没有return语句，那么这个函数返回值为 None
- 有return语句，如果return后不跟任何值，也返回 None
- return 返回一个值，并结束这个函数的调用
- return 可以跟任意的表达式，返回的是最终的计算结果
- 函数有没有返回值跟函数是否要打印输出，没有任何关系
- 更多的函数定义的时候，并不输出任何东西，是以返回值的方式，将数据返回给调用者

## 4. 函数的参数

- 定义函数的时候，可以为函数定义0到任意多个形参
- 形参在定义的时候只占位置，代表未来调用这个函数的人，会传进来的数据。
- 形参可以有默认值，代表如果未来调用的时候不传入时，就是用这个默认值
  - 无默认值的参数必须定义在有默认值的参数之前。
- 参数定义时可以使用【\*形参名】来定义可变参数，收集调用时传入的任意多个实际参数。
  - 收集起来，构成一个元组
  - 如果一个参数都不传，那么就是一个空的元组
- 调用函数时，传入的实参情况要“对应”定义的形参情况
- 调用函数时，可以使用【\*容器数据】，将容器中的元素拆包，一个一个的传入函数

# 五、模块中的入口点

- 在一个模块中的 if \_\_name\_\_ == '\_\_main\_\_': 语句块里的代码，仅当直接启动这个模块时，才执行。

## 六、初步理解“面向对象”程序设计的思想

- OOP ( Object Oriented Programming ) —— 面向对象编程
- 为了解决复杂的软件程序设计过程中，千头万绪的复杂关系，而出现的架构代码的一种思路。
- 将软件程序设计成多个“对象”之间的相互通信。
- 短期目标：
  - **掌握面向对象中的语法**，而不是用这个思想去设计程序。
  - 让你定义一个怎么样的类，能正确定义出来，就很好了。
- 理解的关键词
  - 万物皆对象
  - 先设计（定义）类(class)，再根据类来创建对象(object)
  - 三大特征
    - 封装：将相关的数据和行为封装到一个名字下面，这个名字就是类
    - 继承
    - 多态
- **定义类**
  - 语法

```
class 类名:

 # 构造方法的名字固定为__init__
 # 不是让对象来调用的，是当我们用类名来创建对象时，系统自动调用的
 def __init__(self, 可选的参数):
 构造方法的主体，经常将参数来给属性赋值
 self.属性名1 = 参数1 # 定义类的属性
 self.属性名2 = 参数2 # 定义类的属性
 self.属性名3 = 参数3 # 定义类的属性

 def 方法名1(self, 可选的参数):
 self是未来调用这个方法的时候的对象自身
 方法的主体

 def 方法名2(self, 可选的参数):
 self是未来调用这个方法的时候的对象自身
 方法的主体
```

- 类名的命名规则：
  - 不能出现系统关键字。
  - 只能使用字母、数字、下划线，数字不打头。
  - 自定义的类名每个单词首字母大写
  - 见名知义（名词）
- 理解：
  - 定义类的过程，只是制作模板、蓝图的过程，只是一个计划，并不会真正执行类中的函数
  - 定义类以后，是为了未来使用者来调用（根据类来创建对象，然后使用对象）
  - 你的身份是“上帝”
- 定义“方法”（method）——本质就是类中定义函数

- 类中的方法，可以调用这个类中的属性：`self.属性名`
- 类中的方法，还可以调用这个类中的其他方法：`self.方法名(可选的参数)`
- 定义“属性”（attribute）
  - 可以在类的方法（什么方法都可以）中，使用 `self.属性名 = 值` 来定义属性。但是，一般我们在构造方法 `__init__` 中定义属性，能保证属性及时的创建并赋值。

- **创建类的对象，操作对象**

- 创建对象的语法：

```
变量名 = 类名(可选的参数) # 创建了一个类的对象
```

- 调用对象的方法

```
变量名.方法名(可选的参数)
```

- 访问对象的属性

```
变量名.属性名
```