第十四次课

目标

模块

包

一 模块

1.1 模块的概念

模块是 Python 程序架构的一个核心概念

每一个以扩展名 **py** 结尾的 **Python** 源代码文件都是一个 **模块**

模块名 同样也是一个 **标识符**，需要符合标识符的命名规则

在模块中定义的 **全局变量** 、**函数**、类 都是提供给外界直接使用的 工具

模块 就好比是 **工具包**，要想使用这个工具包中的工具，就需要先 **导入** 这个模块

1.2 模块的两种导入方式

1. import 导入

```
import 模块名1, 模块名2
```

也可以这样导入

```
import 模块名1
import 模块名2
```

导入之后

通过 **模块名.** 使用 模块提供的工具 —— 全局变量、函数、类

注意：

使用 **as** 指定模块的别名

如果模块的名字太长，可以使用 **as** 指定模块的名称，以方便在代码中的使用

语法格式如下：

import 模块名 1 as 模块别名

注意：模块别名 应该符合 大驼峰命名法

2. from...import 导入

如果希望 **从某一个模块** 中，导入 **部分** 工具，就可以使用 from ... import 的方式

import 模块名 是 一次性 **把模块中 所有**工具全部导入，并且通过 模块名/别名 访问

```
# 从 模块 导入 某一个工具
from 模块名1 import 工具名
```

导入之后

不需要 通过 模块名.

可以直接使用 模块提供的工具 —— 全局变量、函数、类

注意

如果 两个模块，存在 **同名的函数**，那么 **后导入模块的函数**，会 **覆盖掉先导入的函数**

开发时 import 代码应该统一写在 **代码的顶部**，更容易及时发现冲突

一旦发现冲突，可以使用 **as 关键字** 给其中一个工具起一个别名

from...import *（知道）

```
# 从 模块 导入 所有工具
from 模块名1 import *
```

这种方式不推荐使用，因为函数重名并没有任何的提示，出现问题不好排查

1.3 __name__属性

一个 独立的 **Python** 文件 就是一个 **模块**

在导入文件时，文件中 **所有没有任何缩进的代码** 都会被执行一遍！

实际开发场景

在实际开发中，每一个模块都是独立开发的，大多都有专人负责

开发人员 通常会在 **模块下方** 增加一些**测试代码**

仅在模块内使用，而被导入到其他文件中不需要执行

__name__ 属性

__name__ 属性可以做到，测试模块的代码 **只在测试情况下被运行**，而在 **被导入时不会被执行！**

__name__ 是 Python 的一个**内置属性**，记录着一个 **字符串**

如果 是**被其他文件导入**的，__name__ 就是 **模块名**

如果 是**当前执行的程序** __name__ 是 **__main__**

在很多 Python 文件中都会看到以下格式的代码：

```
# 导入模块
# 定义全局变量
# 定义类
# 定义函数
# 在代码的最下方
def main():
    # ...
    pass
# 根据 __name__ 判断是否执行下方代码
if __name__ == "__main__":
    main()
```

二 包（package）

概念

包 是一个 **包含多个模块** 的 **特殊目录**

目录下有一个 **特殊**的文件 **__init__.py**

包名的 **命名方式** 和变量名一致，小写字母 + _

好处

使用 **import** 包名 可以一次性导入 包 中 **所有的模块**

案例演练

新建一个 py_message 的 包

在目录下，新建两个文件 send_message 和 receive_message

在 send_message 文件中定义一个 send 函数

在 receive_message 文件中定义一个 receive 函数

在外部直接导入 py_message 的包

__init__.py

要在外界使用 **包** 中的模块，需要在 **__init__.py** 中指定 **对外界提供的模块列表**

```
# 从 当前目录 导入 模块列表
from . import send_message
```

```
from . import receive_message
```

三 第三方库的导入

在 pycharm 中导入第三方库步骤：

File-->project-->Python interpreter--->右边“+”号-->搜索框输入第三库名称

--->选中--->install package 即可。

四 名片管理系统第三库的使用

```
import prettytable as pt
## 按行添加数据
tb = pt.PrettyTable()
tb.field_names = ["City name", "Area", "Population", "Annual Rainfall"]
tb.add_row(["Adelaide",1295, 1158259, 600.5])
tb.add_row(["Brisbane",5905, 1857594, 1146.4])
tb.add_row(["Darwin", 112, 120900, 1714.7])
tb.add_row(["Hobart", 1357, 205556,619.5])
print(tb)
## 打印结果
```

```
+-----------+------+------------+-----------------+
| City name | Area | Population | Annual Rainfall |
+-----------+------+------------+-----------------+
|  Adelaide | 1295 |  1158259   |      600.5      |
|  Brisbane | 5905 |  1857594   |      1146.4     |
|   Darwin  | 112  |   120900   |      1714.7     |
|   Hobart  | 1357 |   205556   |      619.5      |
+-----------+------+------------+-----------------+
```