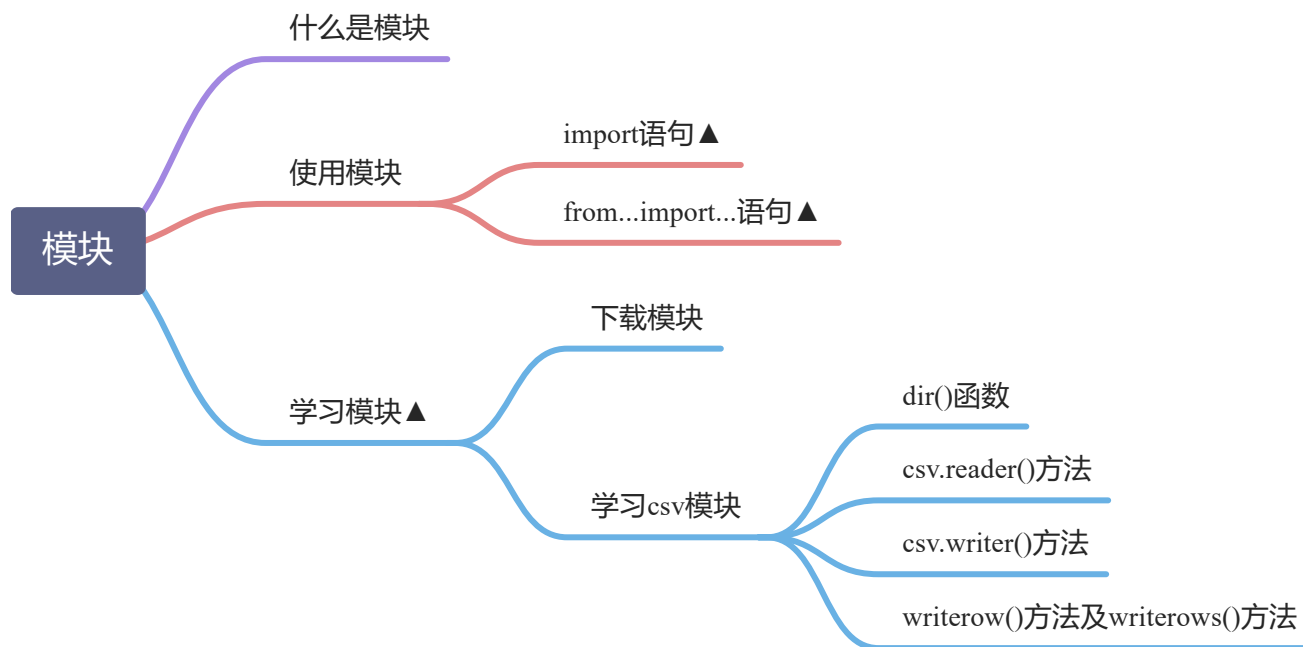


第16课 模块的定义与使用

一、课程结构导图



注：▲为重点知识点。

二、知识点讲解

2.1 什么是模块

概念：模块是一个包含代码的文件，其中定义了一组Python函数或其他对象，并且模块的名称来自文件名。¹

用法：新建一个.py后缀的python代码文件，向新建的py文件写入一组代码后保存并运行，即可在其它的py文件里调用它。我们不仅可以自己创建模块来使用，也可以直接使用Python自带的模块（如time模块）或下载Python第三方模块（如mysql模块）进行使用。

示例:

```
1 # 新建一个名为my_module的模块
2 def printf():
3     print('这是一个新的模块!')
4
5 class Circle:
6     pi = 3.14159
7
8     @classmethod
9     def area(cls,r):    # 计算圆的面积
10         print('圆的面积为: ',cls.pi*r*r)
11
12     @classmethod
13     def girth(cls,r):  # 计算圆的周长
14         print('圆的周长为: ',2*cls.pi*r)
```

注意: 我们创建Python文件时要遵循文件名不与Python自带模块或者下载的第三方模块重名的规则。

2.2 使用模块

2.2.1 import语句

用法: 使用import语句导入一个模块，以调用其中已经封装好的变量，函数以及类。同时我们可以使用import...as...的方式导入模块并给模块一个别名。

示例:

```
1 # 本文件为main.py
2 import my_module    # 导入我们自己写的my_module模块
3 import my_module as ms # 给my_module模块一个别名ms
4
5 my_module.printf() # 打印'这是一个新的模块!'
6 my_module.Circle.girth(1) # 打印'圆的周长为: 6.28318'
7
8 ms.printf() # 使用模块的别名之后调用
```

```
9 ms.Circle.girth(1)
```

代码解析：第4行代码调用了my_module模块中的printf()函数，第6行调用了my_module模块里面Circle类的类方法girth()，第8、9行代码是使用了别名最后的调用。

注意：我们执行模块调用的Python文件要和我们自己写的Python模块文件处在同一个文件夹下（比如我们建立一个文件夹my_code，我们需要把main.py文件和my_module.py文件都放进my_code这个文件夹里。）

2.2.2 from...import...语句

用法1：from 模块名 import 函数名

示例1：

```
1 # 本文件为main.py
2 from my_module import printf    # 导入my_module模块的printf()函数
3
4 printf()    # 执行printf()函数
```

用法2：from 模块名 import 类名

示例2：

```
1 # 本文件为main.py
2 from my_module import Circle    # 导入my_module模块中的Circle类
3
4 Circle.area(2) # 打印半径为2的圆的面积
5 Circle.girth(2) # 打印半径为2的圆的周长
```

注意：同import语句相比，from...import...语句可以直接导入模块中需要的部分来使用，不用再写模块名、类名或是模块名.函数()这种形式。

示例3:

```
1 from my_module import printf,Circle # 混合上面两种用法同时导入函数与类
2
3 printf()
4 Circle.girth(1) # 打印半径为1的圆的周长
```

2.3 学习模块

2.3.1 下载模块

用法: windows用户在终端 (cmd) 输入命令pip install xxx (xxx为模块名)来安装第三方模块, mac os用户在终端 (terminal) 输入命令pip3 install xxx来安装模块。因为绝大多数模块的源头在国外, 所以下载速度可能很慢, 这时候我们可以使用国内的镜像源来加速安装。我们用清华源来做示例, 需要在终端输入的命令为:

```
1 pip install -i https://pypi.tuna.tsinghua.edu.cn/simple xxx
```

2.3.2 学习csv模块

2.3.2.1 dir()函数

用法: 使用dir()函数来查看模块内有哪些变量、函数、类和类方法。

示例:

```
1 import requests # 需要先使用pip install requests命令先安装requests库
2
3 print(dir(requests))
```

2.3.2.2 csv.reader()方法

用法：csv模块的reader()方法接收open函数返回的文件对象作为参数，返回一个可以读取csv文件的对象，我们可以迭代这个对象来打印csv文件的每一行信息。

示例：

我们建立一个名为test.csv的文件，里面的内容如下图：

	A	B	C	D	E
1	1	2	3	4	5
2	6	7	8	9	10
3	11	12	13	14	15
4	16	17	18	19	20
5	21	22	23	24	25
6	26	27	28	29	30
7	31	32	33	34	35
8	36	37	38	39	40

```
1 import csv
2
3 with open("test.csv",newline = '') as f:
4     reader = csv.reader(f)
5     #使用csv的reader()方法，创建一个reader对象
6     for row in reader:
7         #遍历reader对象的每一行
8         print(row)
9
10 print("读取完毕！")
```

2.3.2.3 csv.writer()方法

用法：csv模块的writer()方法同样接收open()函数返回的文件对象作为参数，不过这里我们需要写入内容，所以要求open()函数里的读写模式参数是'a'或者是'w'。writer()方法返回一个可进行写入操作的对象。

示例：

```
1 import csv
2 with open('test.csv','a', newline='',encoding='utf-8') as f:
3     writer = csv.writer(f)
4     print(writer)
5     # 打印<_csv.writer object at 0x7f53ce7ae090>
```

代码解析：第5行打印结果的'0x7f53ce7ae090'代表这个对象在内存中的地址，打印的地址可能每次都会不同。

2.3.2.4 writerow()方法及writerows()方法

用法：csv.writer()方法返回一个可对csv文件进行写入操作的对象，此对象有两个方法writerow()和writerows()，writerow()方法接收一个列表作为参数，将列表写进csv文件的一行里。writerows()方法接收一个嵌套的列表作为参数，并换行写入列表中的每个元素。

示例：

```
1 import csv
2 with open('test.csv','a', newline='',encoding='utf-8') as f:
3     writer = csv.writer(f)
4     print(writer)
5     writer.writerow([41,42,43,44,45])    # 将41,42,43,44,45写进一行里
6     writer.writerows([[46,47,48,49,50],[51,52,53,54,55]])
7     # 将46,47,48,49,50写进一行，51,52,53,54,55写进下一行里
```

三、巩固练习

- 1.（单项选择）如果我们要打印math模块中的自然常数e，下面哪个操作是正确的（）？

A、

```
1 import math
2 print(e)
```

B、

```
1 from math import e
2 print(math.e)
```

C、

```
1 import math
2 print(math.e)
```

D、

```
1 print(e)
```

2. (判断题) 下面的代码是否可以将1, 2, 3, 4, 5这五个数字写进csv文件的一行呢 () ?

```
1 import csv
2 with open('test.csv','a', newline='',encoding='utf-8') as f:
3     writer = csv.writer(f)
4     print(writer)
5     writer.writerow(1,2,3,4,5)
```

参考文献:

Naomi Ceder 著, 戴旭译, 《Python快速入门》(The Quick Python Book), 第120页。
Swaroop C.H 著, 漠伦译, 《简明python编程》(A Byte of Python), 第72页。

