



北京大学  
PEKING UNIVERSITY

# 计算概论B

## 上机课 (2)

2021-2022年秋

# 01

## 第一章 编写程序注意事项

# 在openjudge上练习时



样例输入

\*

```
pythonProject - main.py
1 # This is a sample Python script.
2
3 # Press ⌘R to execute it or replace it with your code.
4 # Press Double ⌘ to search everywhere for classes, files, tool windows, actions, and settings.
5
6
7 def print_hi(name):
8     # Use a breakpoint in the code line below to debug your script.
9     print(f'Hi, {name}') # Press ⌘F8 to toggle the breakpoint.
10
11
12 # Press the green button in the gutter to run the script.
13 if __name__ == '__main__':
14     print_hi('PyCharm')
15
16 # See PyCharm help at https://www.jetbrains.com/help/pycharm/
17
```

样例输出

```

  *
 ***
*****
 ***
  *
```

为检测算法的正确性，同一道题目的测试样例有很多组，需要根据不同情况输出对应的正确答案，检查代码时可以手动构建测试点观察程序输出，修改代码。

# 程序报错时——阅读报错信息



```
main.py x
1 print(sqrt(2))
```

在main程序第一行

```
Traceback (most recent call last):
  File "/Users/shimengzhi/PycharmProjects/pythonProject/main.py", line 1, in <module>
    print(sqrt(2))
NameError: name 'sqrt' is not defined
```

错误类型为没有  
找到sqrt的定义

```
main.py x
1 import math
2 print(math.sqrt(2))
```

- 可以上网查询报错原因

- 库函数——通过math引入sqrt
- 变量——检查变量定义

# 02

## 第二章 数据类型

# 处理数据时需要了解数据类型——常用的数据类型

通过type()函数查看变量的数据类型：

**type(object)**

Return the object's type (The argument of 'object' can be any object.)

```
In [1]: type(100)
```

```
Out[1]: int
```

```
In [2]: type(4.67)
```

```
Out[2]: float
```

- 布尔型 bool
  - 包括True, False

- 整数 int
  - $\mathbb{Z}$ : -1, 0, 1, 2, 3...
- 浮点数 float
  - 数学计算中常用的数据类型，保留一定位数的小数，得到有限精度的结果
- 字符串 str
  - 类似文本，如果文本内容需要进行数学计算，则需要数据类型转换
  - 例如：x = int('30')

# 类型不匹配 & 强制类型转换

- input()的返回类型为字符串类型，如果需要进行数学计算，在读入数据以后要类型转换。

```
main.py ×  
1 rate = input()  
2 amount = 1000 * (1+rate/100)  
3 print(amount)
```



```
main.py ×  
1 rate = input()  
2 rate = float(rate)  
3 amount = 1000 * (1+rate/100)  
4 print(amount)
```

Traceback (most recent call last):

File `"/Users/shimengzhi/PycharmProjects/pythonProject/main.py"`, line 2, in `<module>`  
    `amount = 1000 * (1+rate/100)`

`TypeError: unsupported operand type(s) for /: 'str' and 'int'`

# 输入数据的形式

- 通过终端输入(openjudge读取数据方式):
  - `a, b, c = input().split()`
  - 注意`input()`中不需要带提示语句, 否则会输出多余信息, 干扰正确输出格式
- 通过文件输入(日常处理大量数据, 提供两种参考方式):

```
with open('test.txt', 'r') as f1:  
    list1 = f1.readline()
```

```
print(list1)
```

通过open打开文本文件, 使用  
`read()`, `readline()`, `readlines()`方法

```
import pandas as pd  
import numpy as np
```

使用pandas库读取excel数据

```
path = r'./input.xlsx' # File Path  
data = pd.read_excel(path, engine='openpyxl') # DataFrame  
data = np.array(data) # ndarray
```



# 03

## 第三章 字符串

# print()的输出格式


- print('表达式', ...), 输出一个字符串, 末尾加一个换行符
- print(1,2,3) => '1 2 3' # print()多个参数会以空格进行连接
- 直接输出 & 格式化输出方法

## 1.输出字符串、数字、变量

```
print("run"+"test")
print(100)
L = [1, 2, 'a']
print(L)    # list
D = {'a':1, 'b':2}
print(D)    # dict
```

## 2.格式化输出

```
output = "str:%s length:%d" % ('run', len('run'))
print(output)
```



占位符指明了输出数据的类型, %s为字符串类型, %d为整数类型, %f为浮点数类型  
在字符串表达式后使用%操作符将数据或变量格式化为字符串写入output中

- 由此方法可以将变量从整数、浮点数等类型转为字符串类型

# 转义字符

- '\n', '\t', '\\' 等

```
print('Hello\nWorld\n')  
print('Hello\\World')
```



Hello

World

Hello\World

- 使用raw string

```
print(r'Hello\World\n')
```



Hello\World\n

# 04

## 第四章 程序语言

# 赋值操作

temperature

float

98.6

变量 = 表达式

例: **temperature = 98.6**

一次完整的赋值操作，只写出表达式并不能给变量赋值

```
result = (x + y > x**2)
```

示例：根据右侧表达式类型结果，变量result为布尔类型，取值为True或者False

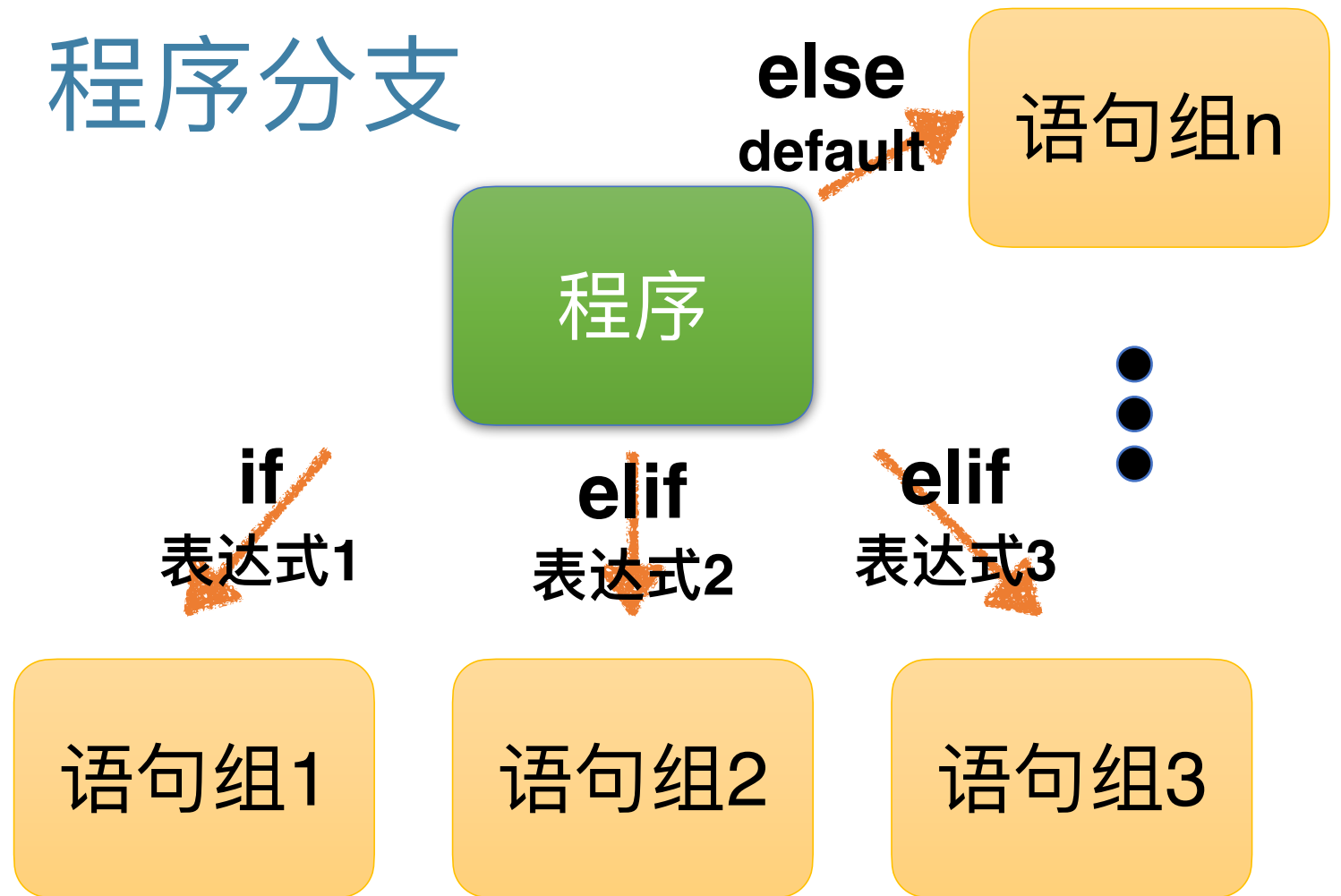
# 程序分支

**if** 表达式：  
语句组

**elif** 表达式：  
语句组

... ..

**else:**  
语句组



# for循环

**for 变量 in 可迭代对象 :**  
语句组

# 循环头部  
# 循环体

```
list1 = ['physics', 'chemistry', 1997, 2000]
for item in list1:
    print(item)
```

```
for idx in range(4):
    print(idx, list1[idx])
```

idx下标从0开始遍历为  
0,1,2,3

# for循环

- 给出n，并给出n个数据应该如何做读入处理？
- 例如：

5

1 2 3 4 5



```
print(input().split())  
1 2 3 4 5  
['1', '2', '3', '4', '5']
```

使用

```
n = int(input())  
input_list = input().split(sep=' ')  
print('input_list for test: ', input_list)  
for i in input_list:  
    data = int(i)  
    print(data*2)
```



数据处理部分



可迭代对象list



# while循环

**while** 表达式:  
语句组

# 循环头部  
# 循环体

```
n = 6
prod = 1
i = 2
while i <= n:
    prod = prod * i
    i = i + 1
print(prod)
```

```
data = {'Alice', 'Bob', 'Carol'}
while True:
    name = input('请输入您的用户名: ')
    if name in data:
        break
    else:
        print('您输入的用户名不存在, 请重新输入')
        continue
```

# 函数

**def 函数名 (参数列表):**  
**语句组**

# 函数头部  
# 函数体

```
def nums(x):  
    a = x + 1  
    b = x + 2  
    c = x + 3  
    return a, b, c
```

参数

0,1,多个返回值

函数返回值存储到变量里


```
print(nums(0))  
# get result  
A, B, C = nums(0)  
print(A, B, C)
```

(1, 2, 3)  
1 2 3

# 程序入口


## 1. 作为脚本顺序执行

```
def nums(x):  
    a = x + 1  
    b = x + 2  
    c = x + 3  
    return a, b, c
```



```
print(nums(0))  
# get result  
A, B, C = nums(0)  
print(A, B, C)
```

## 2. 作为脚本可以正常执行；当该代码被其他python脚本import时，不会执行if以下的语句



```
def nums(x):  
    a = x + 1  
    b = x + 2  
    c = x + 3  
    return a, b, c  
  
if __name__ == '__main__':  
    print(nums(0))  
    # get result  
    A, B, C = nums(0)  
    print(A, B, C)
```



北京大學  
PEKING UNIVERSITY

谢谢大家！