



北京大学
PEKING UNIVERSITY

信息科学技术学院

实用Python程序设计

郭 炜

微信公众号



微博: <http://weibo.com/guoweiofpku>

学会程序和算法，走遍天下都不怕!



北京大学
PEKING UNIVERSITY

信息科学技术学院

Python初探



北京大学
PEKING UNIVERSITY

信息科学技术学院

程序设计语言概述



福建省宁德市北岸公园

常见程序设计语言

➤ **Java** : 使用最广泛的语言。

主要用于互联网服务器端应用和Android手机App开发

➤ **C/C++**: 开发对速度要求较高的系统软件, 以及大型端游

➤ **C#**: 微软公司程序设计语言, 网站开发, 桌面应用开发

➤ **PHP**: 网站开发

➤ **JavaScript**: 网站前端开发。在浏览器中运行

➤ **Swift**: 苹果公司语言, 用于开发iPhone App和Mac 桌面应用程序

常见程序设计语言

Jun 2017	Jun 2016	Change	Programming Language	Ratings	Change
1	1		Java	14.493%	-6.30%
2	2		C	6.848%	-5.53%
3	3		C++	5.723%	-0.48%
4	4		Python	4.333%	+0.43%
5	5		C#	3.530%	-0.26%
6	9	▲	Visual Basic .NET	3.111%	+0.76%
7	7		JavaScript	3.025%	+0.44%
8	6	▼	PHP	2.774%	-0.45%
9	8	▼	Perl	2.309%	-0.09%
10	12	▲	Assembly language	2.252%	+0.13%
11	10	▼	Ruby	2.222%	-0.11%
12	14	▲	Swift	2.209%	+0.38%

为什么学 Python

- 简单易学
- 有众多的库支持，功能强大
- 编程效率高
- 跨平台（Windows, Linux, Mac OS...）

Python的历史

Python的作者，Guido von Rossum，荷兰人。

1982年，Guido从阿姆斯特丹大学 (University of Amsterdam) 获得了数学和计算机硕士学位。

之所以选中Python（大蟒蛇的意思）作为该编程语言的名字，是因为他是一个叫Monty Python的喜剧团体的[爱好者](#)。



Python的历史

www.python.org

Python 雏形	1991年
Python 1.0	1994年11月
Python 2.0	2000年

Python 2.7.13是Python 2的最后版本

Python 3.0 2008年

Python 3 不向后兼容Python 2

目前: Python 3.6.2





北京大学
PEKING UNIVERSITY

信息科学技术学院 郭炜

Python开发环境 的安装和使用



内蒙古阿斯哈图石林

下载Python

<https://www.python.org/downloads/>

[Mac OS X 64-bit/32-bit installer](#)

[Windows x86-64 executable installer](#)

64位Windows

[Windows x86 executable installer](#)

32位Windows

搭建Python 开发环境

- 1) 下载并安装 Python 3.x
- 2) 下载并安装 pycharm-community-4.0.4
(非必需，但强烈推荐)

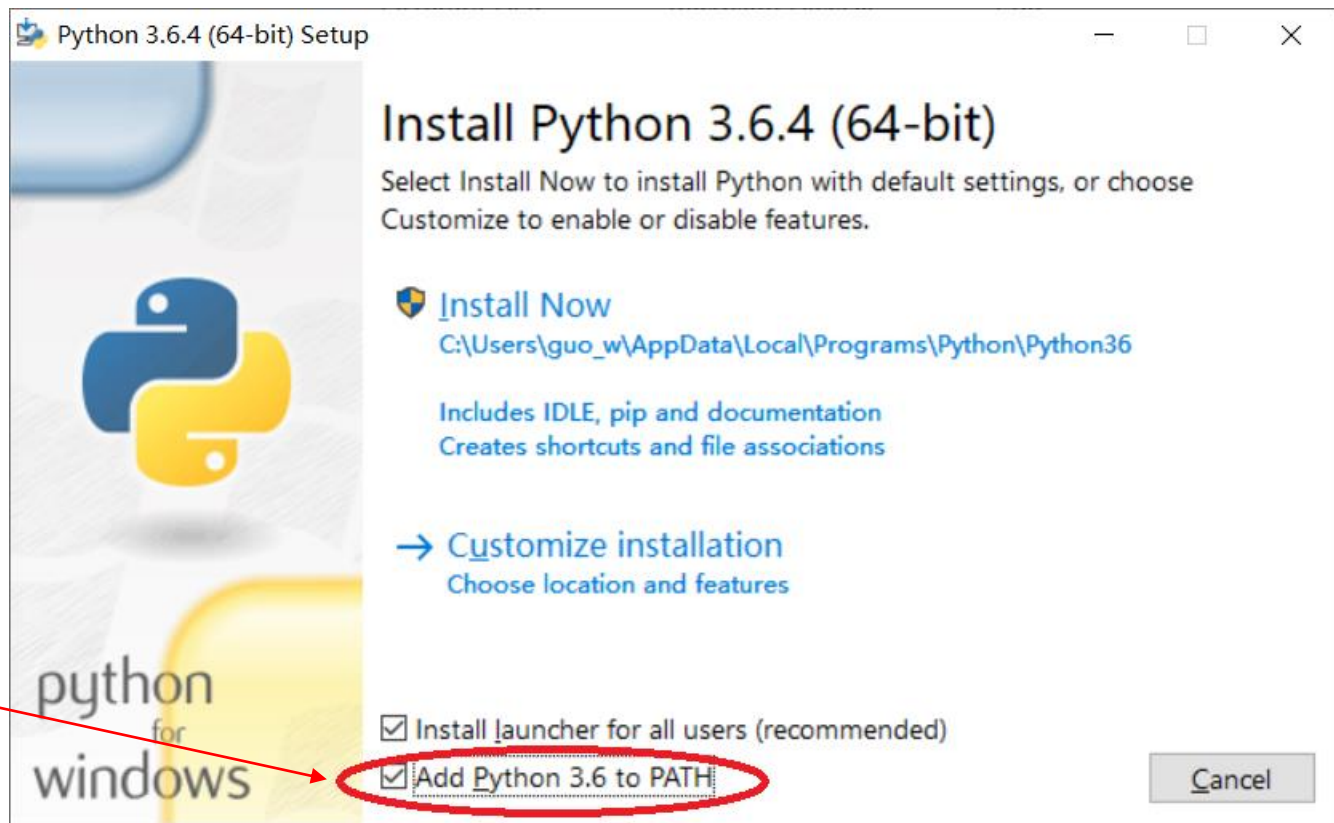
<http://www.jetbrains.com/pycharm/download/#section=windows>

- 3) 配置 pycharm
- 4) 在 pycharm建工程
- 5) 编写、运行python程序

安装Python

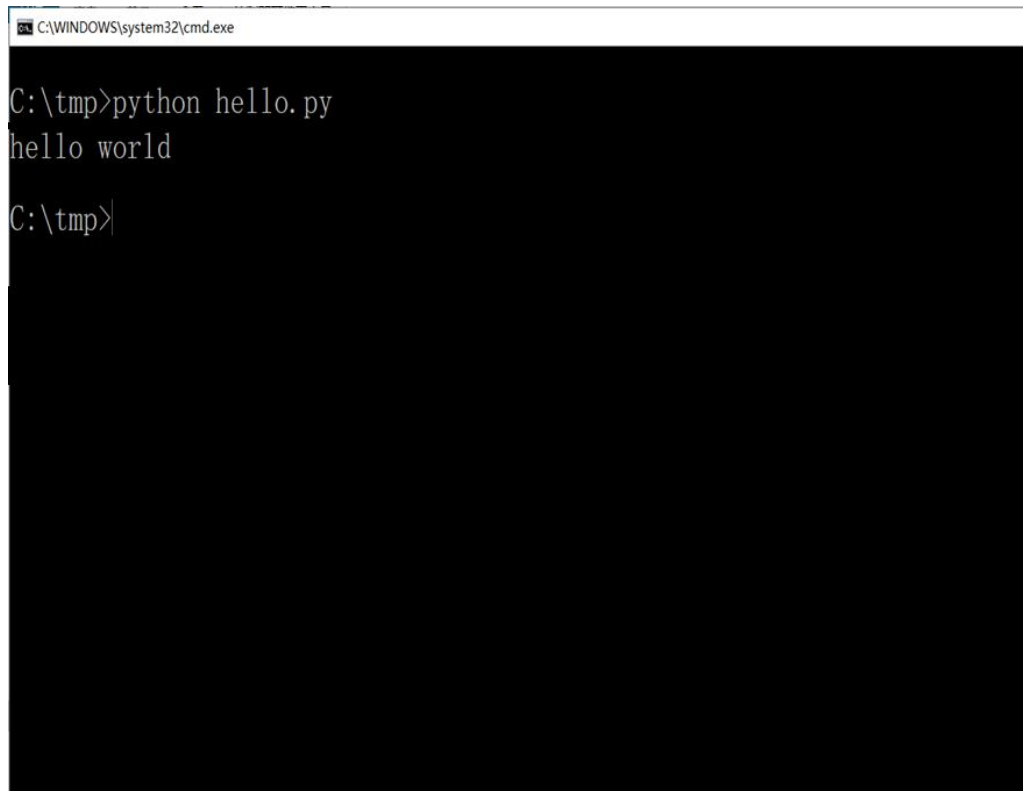
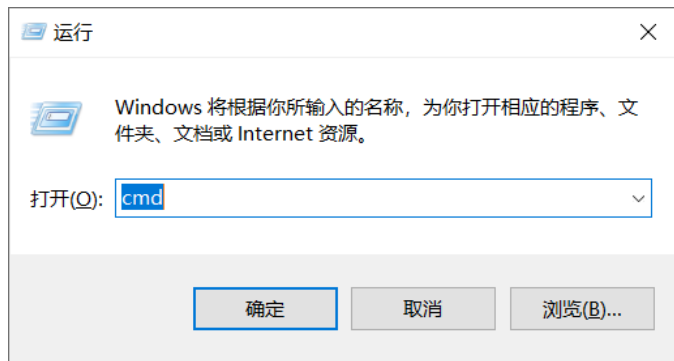
1. 不是下载下来就好了，还要运行下载下来的文件，进行安装

2. 要勾上，否则pycharm里面找不到python



以命令行方式运行python程序

Windows下，Win+R 键，可以弹出左边“运行”窗口，敲"cmd"确定，就能弹出右边cmd窗口(命令行窗口)



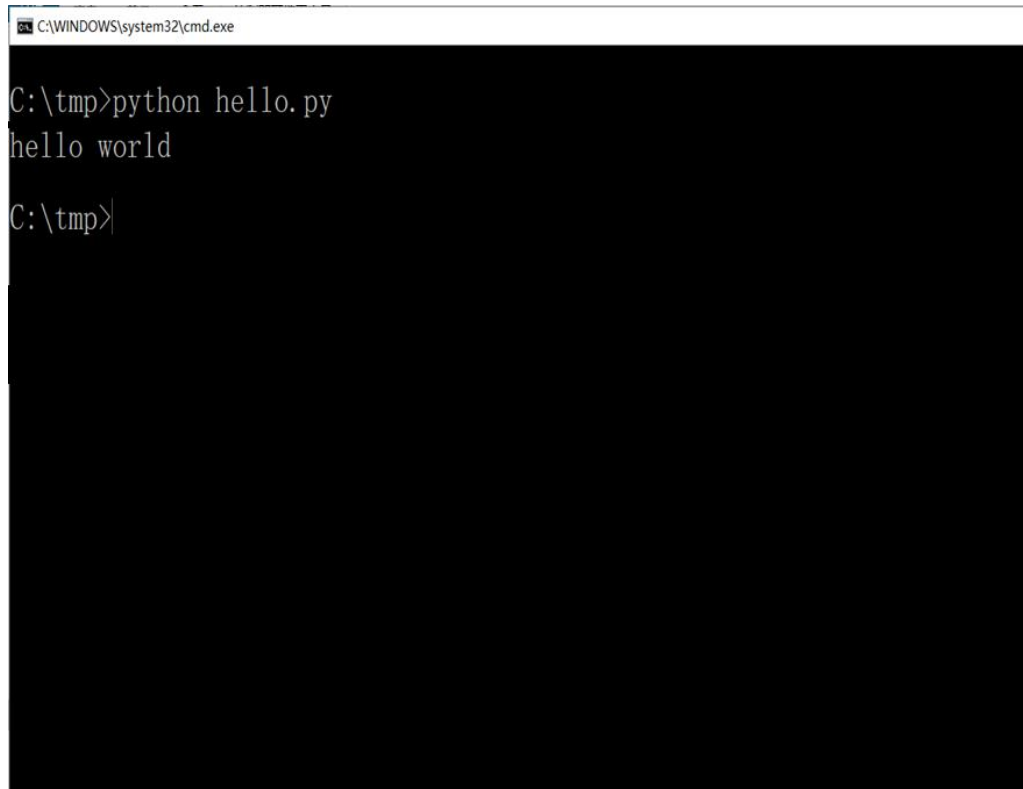
Mac上相应操作，是从LaunchPad里面启动“终端”

以命令行方式运行python程序

Windows下，Win+R 键，可以弹出左边“运行”窗口，敲"cmd"确定，就能弹出右边cmd窗口(命令行窗口)

敲 `cd \`
再敲 `cd \tmp` (或其它文件夹名)
进入存放python程序的文件夹

`python XXX.py`就可以运行XXX.py



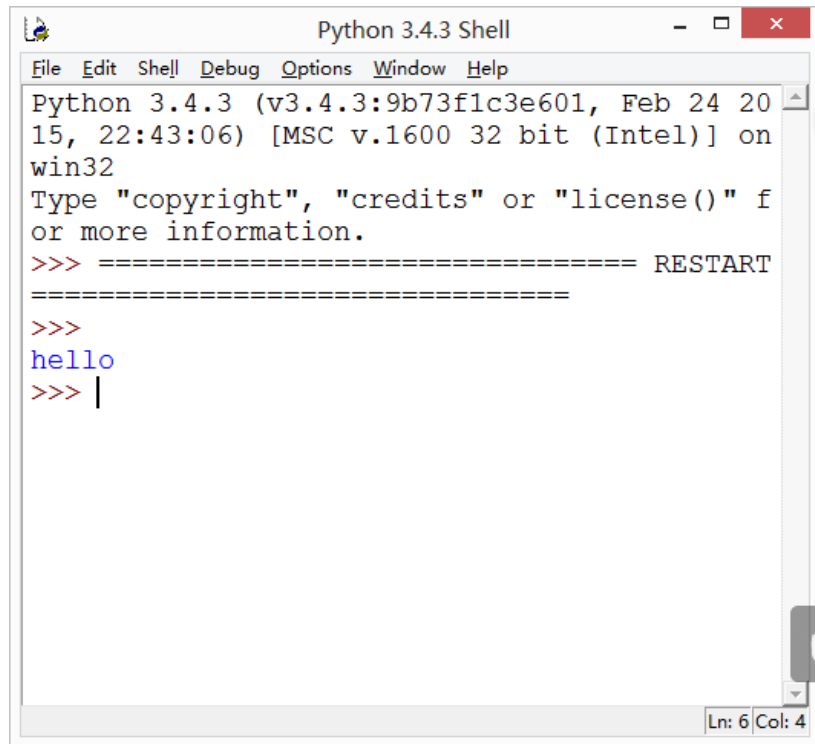
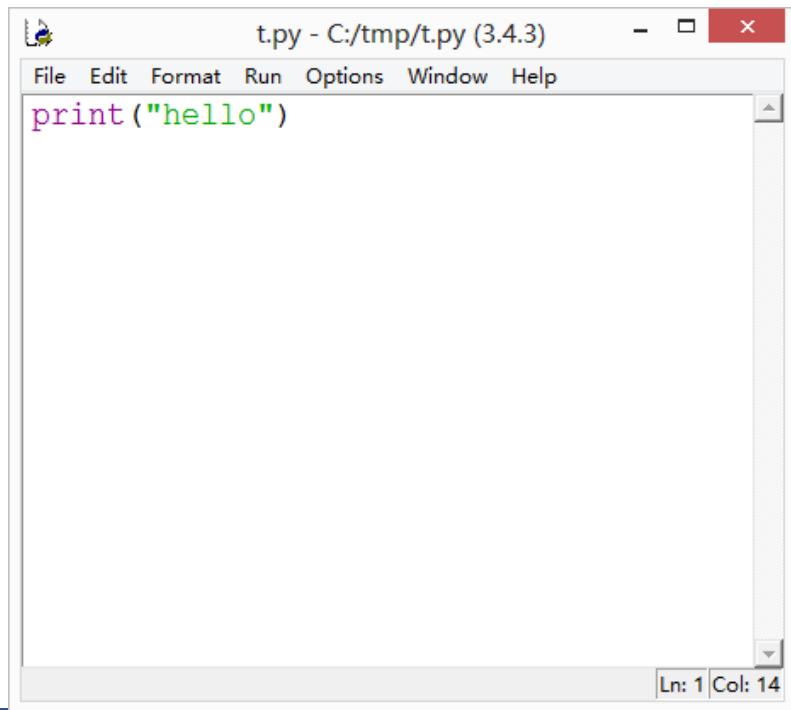
```
C:\WINDOWS\system32\cmd.exe

C:\tmp>python hello.py
hello world

C:\tmp>
```

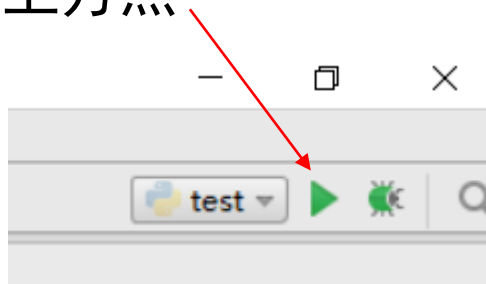
用idle 编辑程序并运行Python程序

window键，然后敲 idle，可找到idle，
然后 File | New | Run

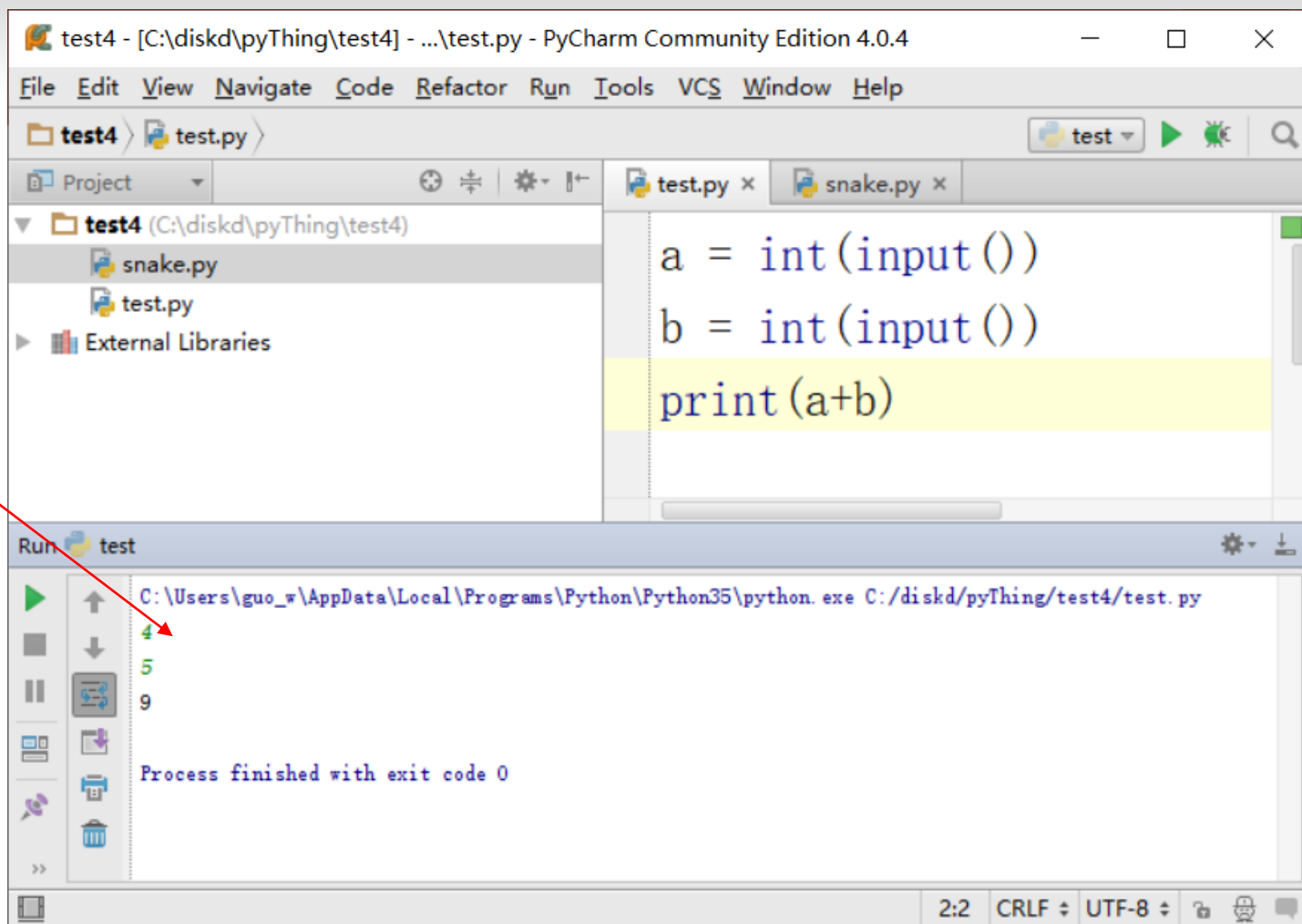


用PyCharm运行python程序

- 1) File|New Project 新建一个工程 (Mac电脑上, PyCharm的菜单在屏幕最上方)
- 2) File|New 新建一个 **Python File** 并随意命名
- 3) Run 菜单运行程序。如果同时打开多个.py程序, 则用 Run|Run 可以选一个。以后要运行同一个文件, 在右上方点

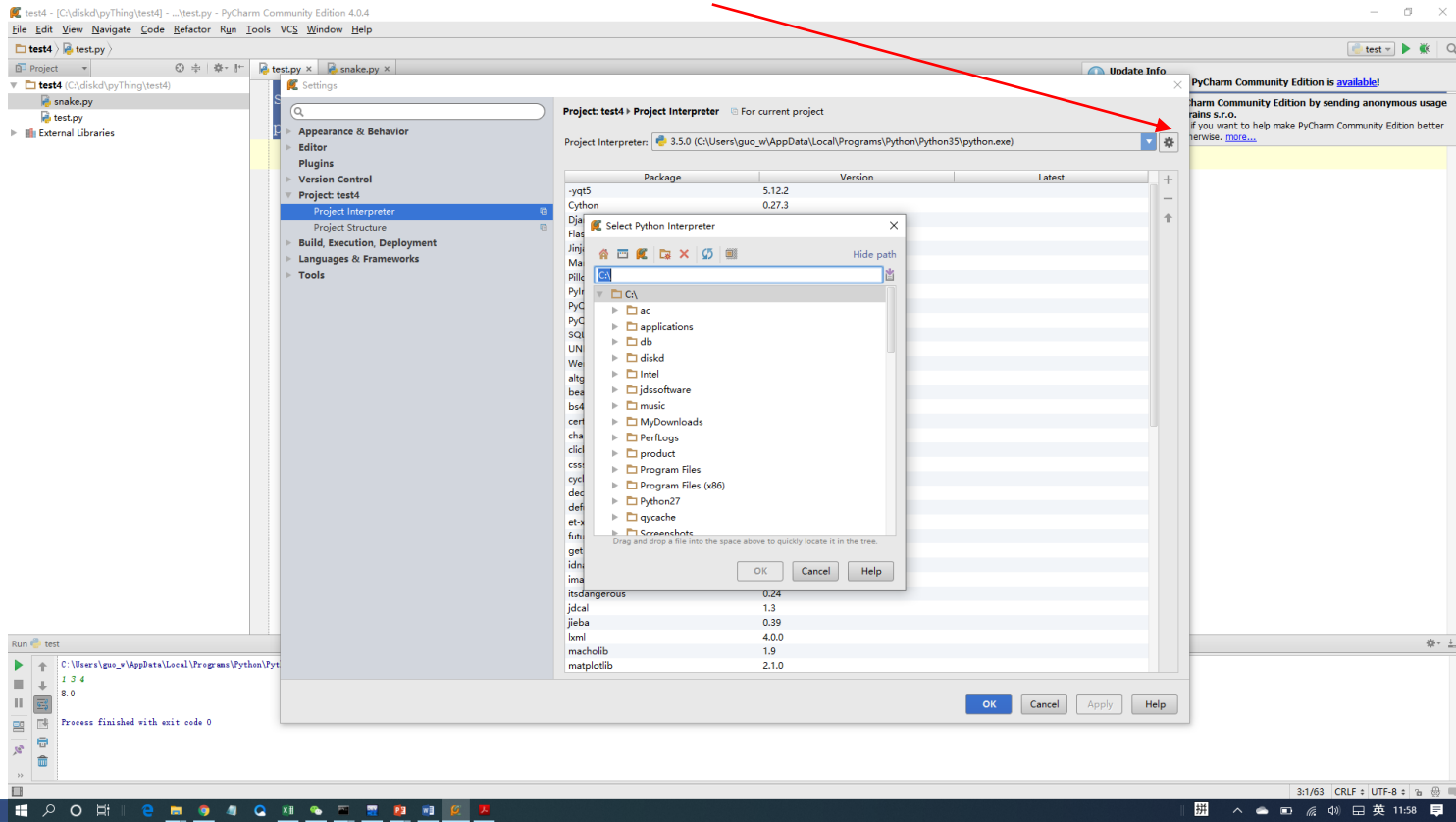


4) 如果程序需要输入，则在PyCharm下方输入，输出结果也会出现在下方



为project设置解释器

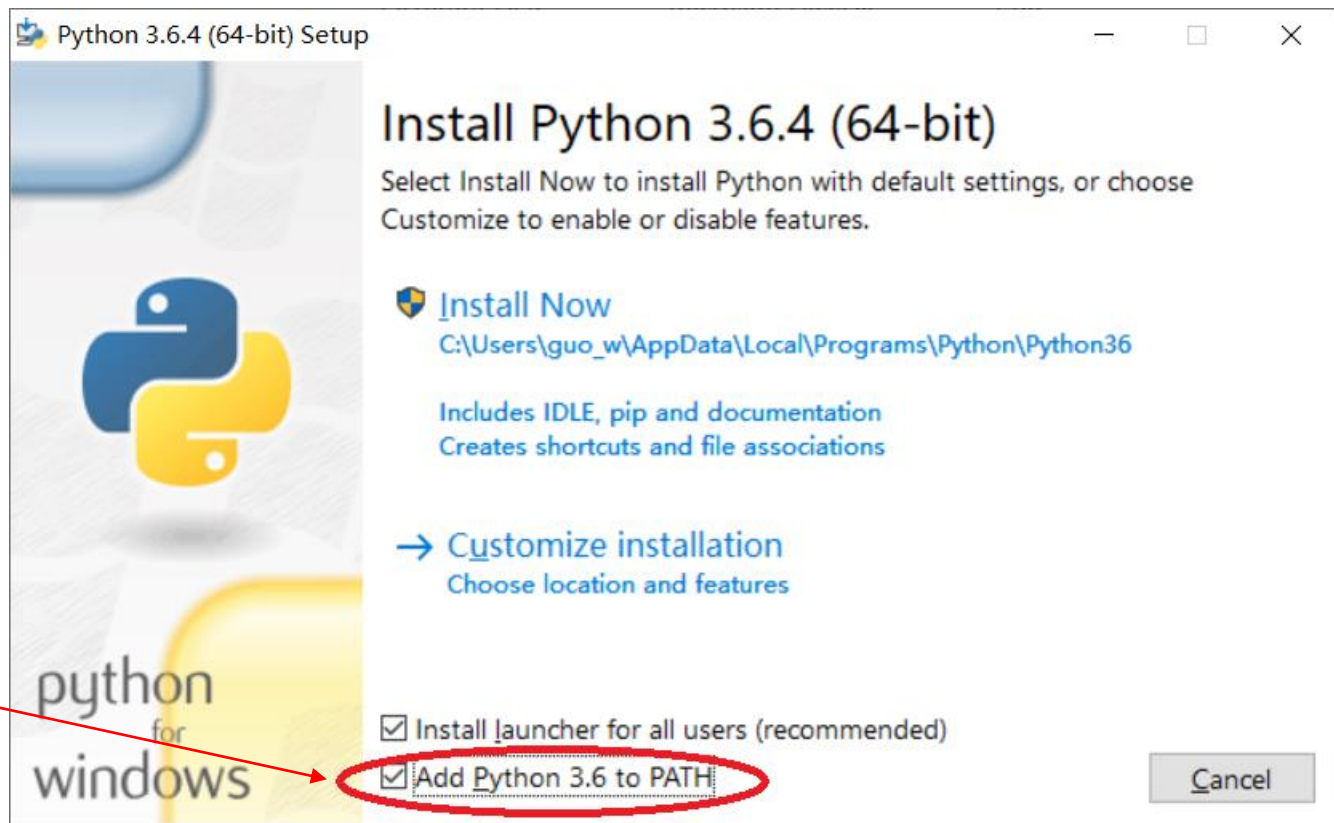
如果pycharm找不到python interpreter, 则: File|Settings|Project: XXX |Project interpreter, 然后选解释器。如果没得选, 就add local加一个, 加的办法是选python安装的文件夹



安装Python

要在命令行方式启动python，就要在安装的时候勾上

不行就卸载重装

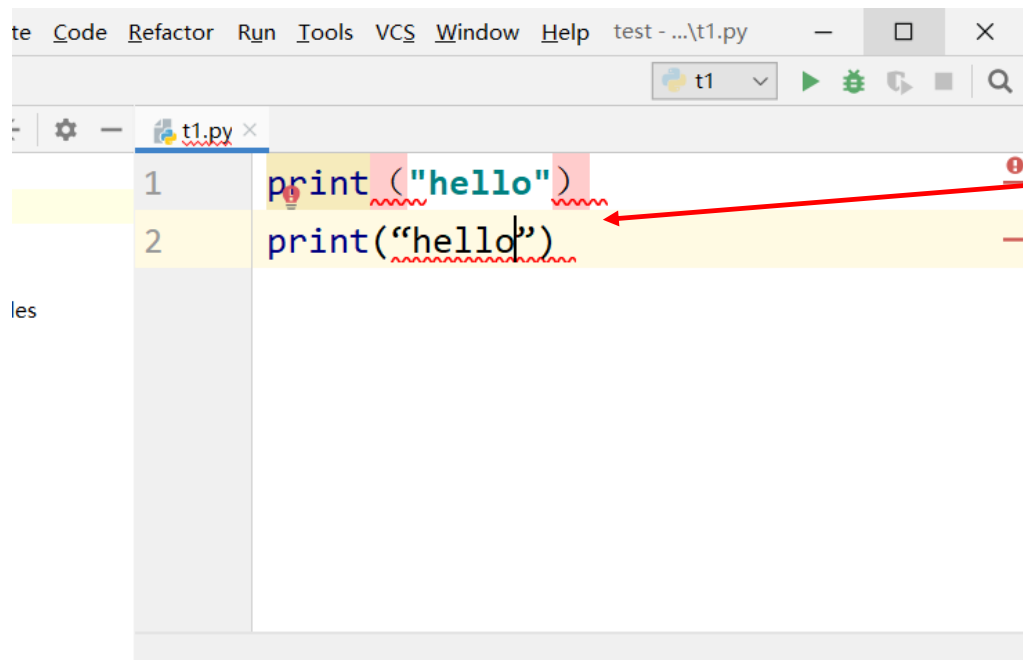


最简单的Python程序

```
print("Hello,World")
```

程序中的符号

- 程序中的所有字符都必须是英文(半角)字符，不能是中文(全角)字符，注意 . , (, " 之类的，都必须是英文字符！



输入了中文的括号和双引号，导致pyCharm以红线提示错误！



北京大学
PEKING UNIVERSITY

信息科学技术学院

Python语言基本要素



北京大学
PEKING UNIVERSITY

信息科学技术学院 郭炜

符号和注释



瑞士布里茨恩湖

程序中的符号

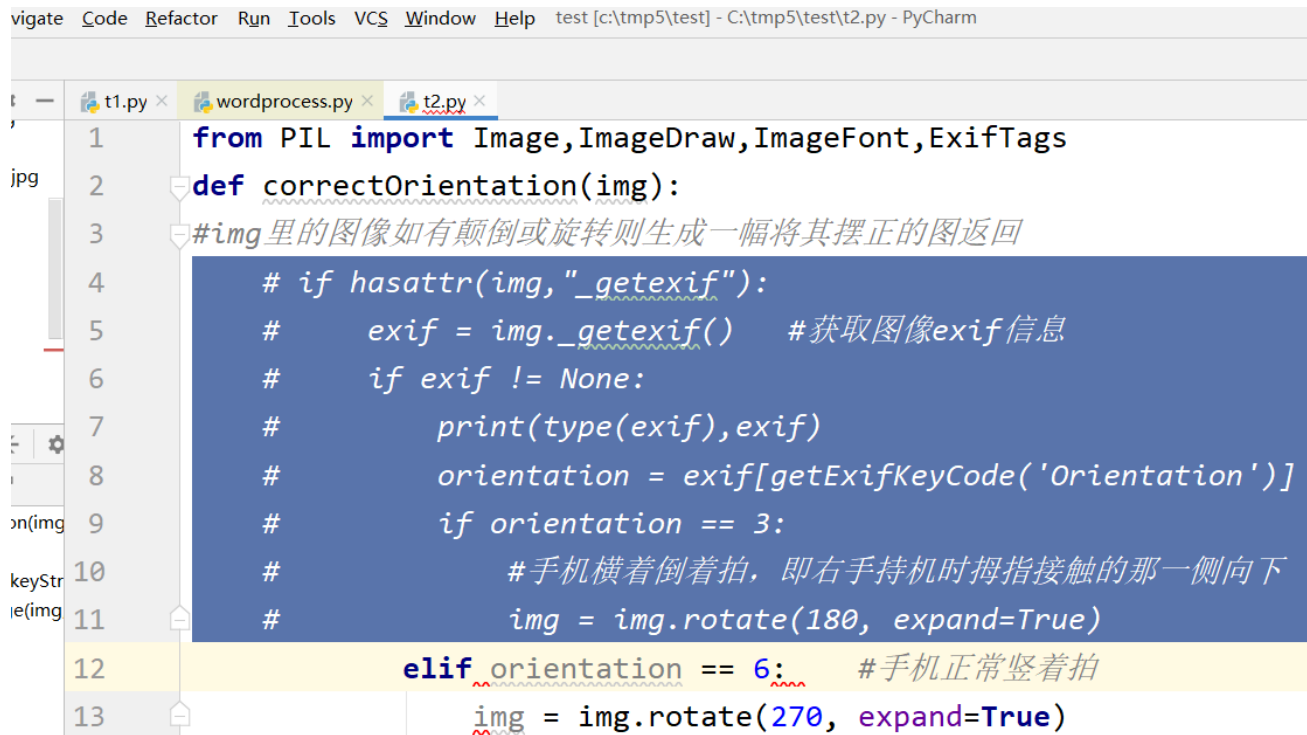
- 程序中的所有字符都必须是英文字符，不能是中文的字符，**注意， : . ' , (" 之类的，都必须是英文字符！不能是中文的！不能在输入中文状态下输入！** 中文的标点符号看起来一般比英文的更宽。
- 除非就是要输出中文文字，才会在“字符串”中使用中文

注释

- 并非需要执行的指令，只是为了编程者方便理解程序之用
- 注释： "#" 开头
a = b # 让a和b的值相同

注释

- Pycharm中， 选中若干行，`Ctrl + '/'` 可将这些行都变成注释/都由注释变为非注释



```
1 from PIL import Image, ImageDraw, ImageFont, ExifTags
2 def correctOrientation(img):
3     #img 里的图像如有颠倒或旋转则生成一幅将其摆正的图返回
4     # if hasattr(img, "_getexif"):
5     #     exif = img._getexif() #获取图像exif信息
6     #     if exif != None:
7     #         print(type(exif), exif)
8     #         orientation = exif[getExifKeyCode('Orientation')]
9     #         if orientation == 3:
10     #             #手机横着倒着拍，即右手持机时拇指接触的那一侧向下
11     #             img = img.rotate(180, expand=True)
12 elif orientation == 6: #手机正常竖着拍
13     img = img.rotate(270, expand=True)
```



北京大学
PEKING UNIVERSITY

信息科学技术学院

变量



新疆喀拉峻鳄鱼湾

变量

➤ 变量有名字，可以用来存储数据。其值可变。

```
a = 12
b = a      # 让b的值变得和a一样
print(a+b)
a = "hello"
print(a)
```

输出：

24
hello

变量的命名办法

- 变量由大小写字母、数字和下划线构成, 中间不能有空格, 长度不限, 不能以数字开头

```
name    _doorNum    x1    y    z    a2    A  
number_of_students MYTYPE
```

- 变量名最好能够体现变量的含义
- 多单词的变量名, 最好第一个单词小写, 后面单词首字母大写

```
dateOfBirth numOfDogs    bookPrice
```

变量的命名办法

➤ 变量名是大小写相关的

a和 A, name 和 Name 是不同的变量。

➤ 有些名字Python预留了，不可用作变量的名字

```
and as assert break class continue def
del elif else except  exec for finally from
global if import in is lambda not or pass
print raise return try while with yield
.....
```

如果不小心用了，python会报错



北京大学
PEKING UNIVERSITY

信息科学技术学院

赋值语句



新疆布尔津五彩滩

赋值语句

形式：

变量 = 表达式

将变量的值变得和“表达式”的值一样

变量、数、字符串.....都是“表达式”

赋值语句

```
a = "he"  
print(a)           # >>he  
b = 3+2  
a = b              # a的值变得和b一样  
print(b)           # >>5  
print(a)           # >>5  
b = b + a          # b的值改为原来b的值加a  
print(b)           # >>10
```

赋值语句

```
a,b = "he",12
print(a,b)           # >>he 12
a,b = b,a            # 交换 a,b的值
print(a,b)           # >>12 he
c,a,b = a,b,a
print(a,b,c)          # >>he 12 12
a = b = c = 10
print(a,b,c)          #>>10 10 10
```



北京大学
PEKING UNIVERSITY

信息科学技术学院

字符串初步



新疆塞里木湖

字符串

➤ 可以且**必须**用单引号、双引号或三单引号括起来

```
x = "I said: 'hello'"
```

```
print(x)          # >>I said:'hello'
```

```
print('I said:"hello"') #>>I said:"hello"
```

```
print('I said: 'he said "hello"'.') #>>I said: 'he said  
"hello"'.
```

```
print("this \
```

is \

`good") #>>this is good` 字符串太长时，可以分行写

```
print( hello,word)    #错！ 没有用引号括起来
```

字符串

➤ 字符串里面不会包含变量

```
s = 1.75
```

```
print(s) #>>1.75
```

```
print("I am s m tall") #>>I am s m tall
```

字符串中的 s 就是个字符，和前面的变量 s 没有关系!!!

字符串必须用引号括起来，用引号括起来的就是字符串！

```
a = 4
```

```
b = 5
```

```
print("a+b") #>> a+b 不会打出9!!!!
```

字符串

➤ 三双引号字符串中可以包含换行符、制表符以及其他特殊字符。

```
para_str = """多行字符串可以使用制表符
```

```
TAB ( \t )。
```

```
也可以使用换行符 [ \n ]。
```

```
<HTML><HEAD><TITLE>
```

```
Friends CGI Demo</TITLE></HEAD>
```

```
<BODY><H3>ERROR</H3>
```

```
<FORM><INPUT TYPE=button VALUE=Back
```

```
ONCLICK="window.history.back()"></FORM>
```

```
</BODY></HTML>
```

```
"""
```

```
print (para_str)
```

多行字符串可以使用制表符

TAB ()。

也可以使用换行符 [

]

<HTML><HEAD><TITLE>

Friends CGI Demo</TITLE></HEAD>

<BODY><H3>ERROR</H3>

<FORM><INPUT TYPE=button VALUE=Back

ONCLICK="window.history.back()"></FORM>

</BODY></HTML>

字符串

➤ 字符串的下标

有 n 个字符的字符串，其中的每个字符都是长度为1的字符串：

从左到右依次编号为 $0, 1, 2, \dots, n-1$

从右到左依次编号为 $-1, -2, \dots, -n$

编号就是下标

```
a = "ABCD"
```

```
print (a[-1])           #>>D
```

```
print (a[0])            #>>A
```

```
print (a[2])            #>>C
```

字符串不可修改

➤ 不可以修改字符串中的字符

```
a = "ABCD"
```

```
a[2] = 'k'    #错，字符串中的字符不能修改
```


字符串

➤ 用"+"连接字符串

```
a = "ABCD"
```

```
b = "1234"
```

```
a = a + b
```

```
print(a) #>>ABCD1234
```

```
a = a + a[1]
```

```
print(a)
```

字符串

➤ 用"+"连接字符串

```
a = "ABCD"
```

```
b = "1234"
```

```
a = a + b
```

```
print(a) #>>ABCD1234
```

```
a = a + a[1]
```

```
print(a) #>>ABCD1234B
```

字符串

➤ 用 in , not in 判断子串

```
a = "Hello"
```

```
b = "Python"
```

```
print("el" in a)           #>>True
```

```
print("th" not in b)       #>>False
```

```
print("lot" in a)          #>>False
```



北京大学
PEKING UNIVERSITY

信息科学技术学院

字符串和数的 转换



内蒙古锡盟草原

字符串和数的转换

`int(x)` : 把字符串转换成整数

(x不会变成整数, `int(x)` 这个表达式的值是整数)

`float(x)` : 把字符串x转换成小数

`str(x)` : 把x转换成字符串

`eval(x)` : 把字符串x看作一个python表达式, 求其值

初学者程序出现runtime error, 多半是由于做了不合法的转换, 如 `int(x)` 而 x 为 "a12" 或 "12.34",

`float(x)` 而x为 "abc"

字符串和数的转换

```
a = 15
```

```
b = "12"
```

```
c = a + b    #错误的语句,字符串和整数无法相加
```

```
print(a + int(b)) #>>27 b没有变成整数 int(b) 这个表达式的值是个整数
```

```
print(str(a) + b) #>>1512
```

```
c = 1 + float("3.5")
```

```
print(c)          #>>4.5
```

```
print(3+eval("4.5")) #>>7.5
```

```
print(eval("3+2"))  #>>5
```

```
print(eval("3+a"))  #>>18
```

小数到整数的转换

➤ `int(x)` `x`是小数，则去尾取整

`int(3.2)` `#3`

`int(3.9)` `#3`

Python数据类型

<code>int</code>	整数	<code>123456899899</code>
<code>float</code>	小数	<code>3.2 1.5E6</code>
<code>complex</code>	复数	<code>1+2j</code>
<code>str</code>	字符串	<code>"hello"</code>
<code>list</code>	列表	<code>[1,2,'ok',4.3]</code>
<code>tuple</code>	元组	<code>(1,2,'ok',4.3)</code>
<code>bool</code>	布尔	<code>True False</code>
<code>dict</code>	字典	<code>{"tom":20,"jack":30}</code>
<code>set</code>	集合	<code>{"tom",18,71}</code>



北京大学
PEKING UNIVERSITY

信息科学技术学院 郭炜

输入输出初步



新加坡金沙酒店无边泳池

输出语句print

`print(x, y, z...)` # 也可以只输出一项

连续输出多项，以空格分隔，然后换行

`print(x, y, z..., end="")`

连续输出多项，以空格分隔，不换行

`print(1, 2, 3, end="")`

`print("ok")`

`#>>1 2 3ok`

注意：做OpenJudge作业的时候，`print`里面慎用`' '`，因为可能会多出不该有的空格

输入语句 input

格式：

```
x = input(y)
```

x 是变量

y 是字符串，或任何值为字符串的表达式

输出y，并等待输入。敲回车后输入的字符串被赋值给 x

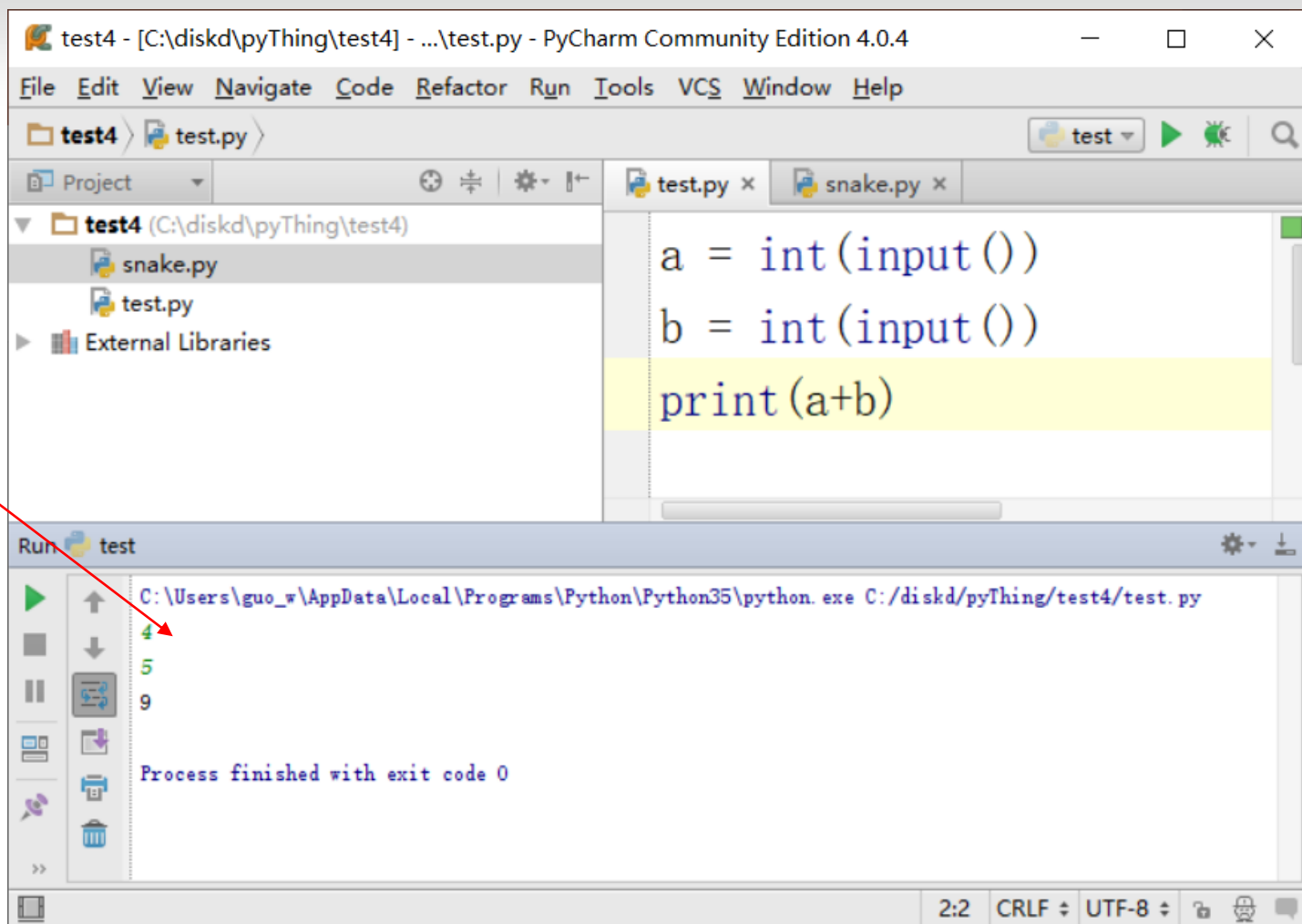
最简单的能处理输入的Python程序

```
s = input("请输入你的名字:")  
print(s + ",你好!")
```

注意：以后做OpenJudge作业的时候，input里面不要写任何东西

```
s = input()
```

4) 如果程序需要输入，则在PyCharm下方输入，输出结果也会出现在下方



输入语句input

`input()` 每次输入一行，如果有多行输入，就用多次`input()`

初学者常犯错误：

如果输入已经结束（后面没有输入数据了），再进行`input()`，就会产生runtime error

不要多行输入只用一次`input`，也不要一行输入用多次`input`



北京大学
PEKING UNIVERSITY

信息科学技术学院 郭炜

列表初步



德国国王湖

列表

➤ 列表可以有0到任意多个元素，元素可以通过下标访问

```
empty = []    #空表
```

```
list1 = ['Google', 'Runoob', 1997, 2000]
```

```
list2 = [1, 2, 3, 4, 5, 6, 7 ]
```

```
print ("list1[0]: ", list1[0])  #>>list1[0]:  Google
```

```
list1[2] = 2001    #更改了列表中下标为2的元素。
```

```
a = 2
```

```
print ("更新后的第三个元素为  :", list1[a])  #变量也能做下标
```

```
#>>更新后的第三个元素为  :  2001
```


列表

➤ 用in 判断列表是否包含某个元素

```
lst = [1,2,3,"4",5]
```

```
print(4 in lst,3 in lst,"4" in lst)#>>False True True
```

实例：输入两个整数求和

```
s = input()
numbers = s.split()
print( int(numbers[0]) + int(numbers[1]))
```

若输入：

3 4

则：

s 为： "3 4"

numbers为： ["3", "4"]

输出： 7

字符串分割成列表

若x是字符串，则 `x.split()` 的值是一个列表，包含字符串x经空格、制表符、换行符分隔得到的所有子串

```
print("34\t\t45\n7".split())    #>>['34', '45', '7']
```

#\t是制表符，\n是换行符号

```
print("ab cd hello ".split())    #>>['ab', 'cd', 'hello']
```

```
s = "12 34"
```

```
lst = s.split()
```

```
print(lst)    #>>['12', '34']
```

单选题 1分



```
print(input().split()[2][1])
```

输入为

abc def 1234 567

输出是：

- ☐ A abc
- ☐ B def
- ☐ C 2
- ☐ D d

此题未设答案

提交

单选题 1分



```
print(input().split()[2][1])
```

输入为

abc def 1234 567

输出是：

- ☐ A abc
- ☐ B def
- ☒ C 2
- ☐ D d

提交

程序顶格书写

➤ 程序每行前面不能留空格！（例外后面讲）

