

[笔记][LIKE-Python-2][05]

Python

[笔记][LIKE-Python-2][05]

011. Python输入

012. Python的输出-2.x版本

013. Python的输出-3.x版本

014. Python的格式符(补充)

011. Python输入

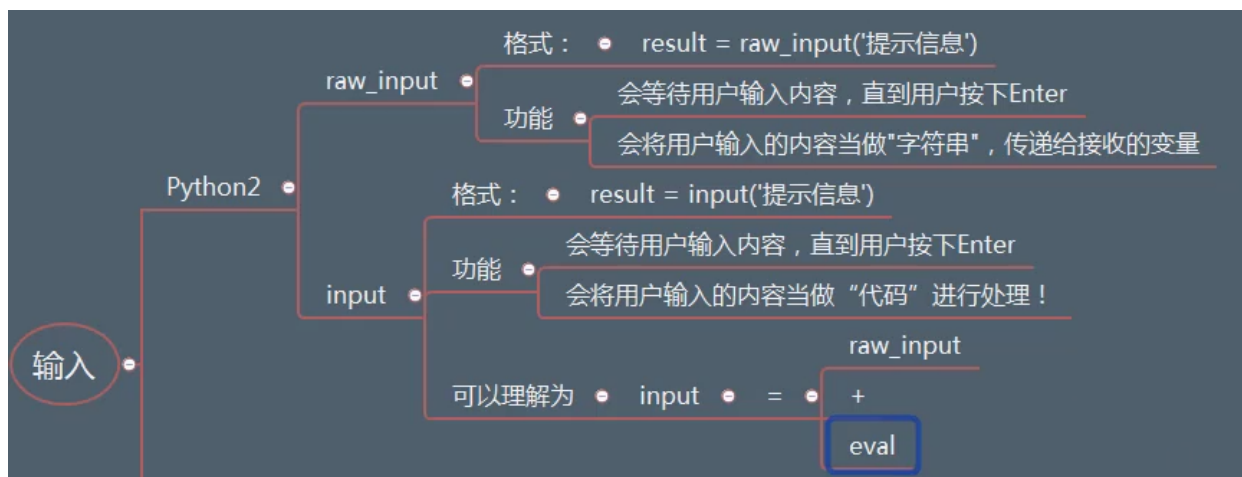


程序的数据来源

- 程序内部写死的
- 从文件中读取
- 网络服务器进行获取
- 接收用户的输入

处理数据的方式

- 写回到文件里面保存
- 发送回服务器
- 打印到控制台，通过一些界面，展示给用户看



```
# -*- coding: utf-8 -*-
```

```
# Python 2.x 版本
```

```
# 函数: 封装了一些功能，只需要写一个函数名字，就可以使用这些功能
```

```
content = raw_input('请输入内容') # 把输入的内容当作字符串处理
```

```
print type(content)
```

```
print content
```

```
content = input('请输入内容') # 把输入的内容当作代码处理。
```

```
print type(content)
```

```
print content
```

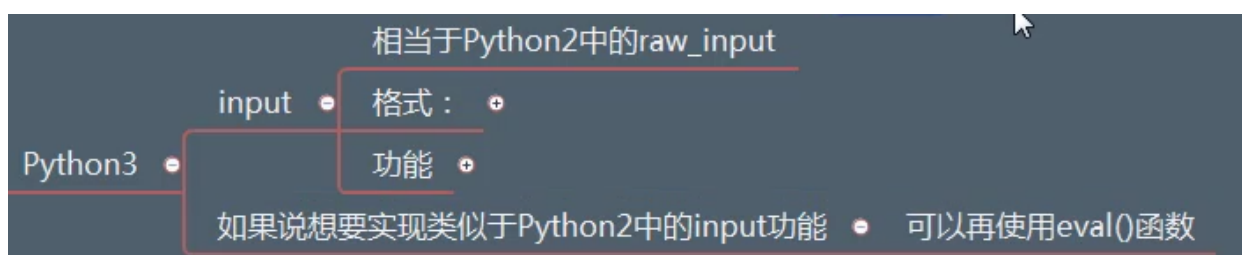
```
# input 相当于 raw_input + eval
```

```
content = raw_input('请输入内容')
```

```
result = eval(content)
```

```
print type(content)
```

```
print content
```



```
# Python 3.x 中的 input 相当于 Python 2.x 中的 raw_input
```

```
content = input('请输入内容')
```

```
print(type(content))
```

```
print(content)
```

```
# 实现 Python 2.x 中的 raw_input 功能
```

```
content = input('请输入内容')
```

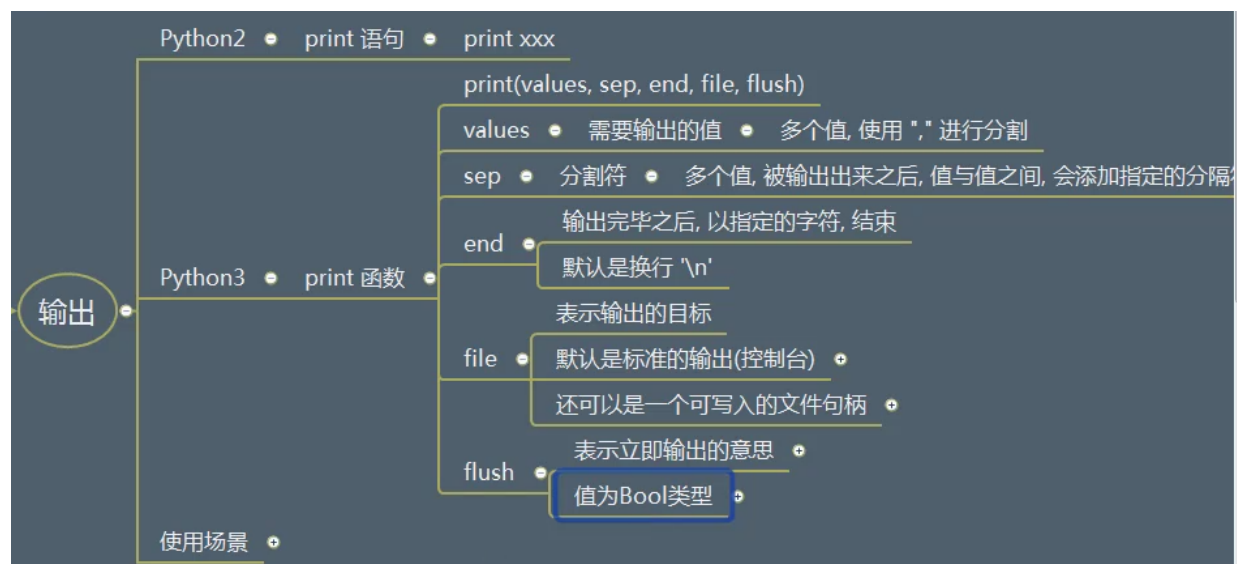
```
result = eval(content)
```

```
print(type(result))
```

```
print(result)
```

012. Python的输出-2.x版本

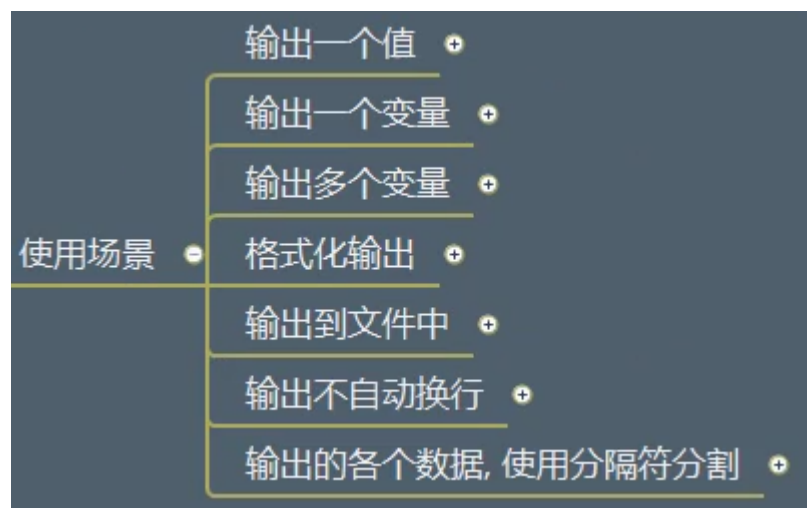
sep 默认是空格



打印输出到控制台的流程：

‘字符串’ – 缓冲区 – 控制台

如果 `flush=True` 的话，立即把缓冲区内容输出到控制台



```
# -*- coding: utf-8 -*-
# Python 2.x

# 输出一个值
print 123

# 输出一个变量
num = 10
print num
```

```
# 输出多个变量
num2 = 66
print num, num2

# 格式化输出
name = '团子'
age = 18
# 我的名字是团子，年龄是18
print '我的名字是', name, '年龄是', age
print '我的名字是%s，年龄是%d' % (name, age)
# 注意：格式化输出不是隶属于print语句，而是属于字符串

# .format 方法，字符串是一个对象，可以调用对象的方法
print '我的名字是{0}，年龄是{1}'.format(name, age)
# 两个都找第0个值
print '我的名字是{0}，年龄是{0}'.format(name, age)

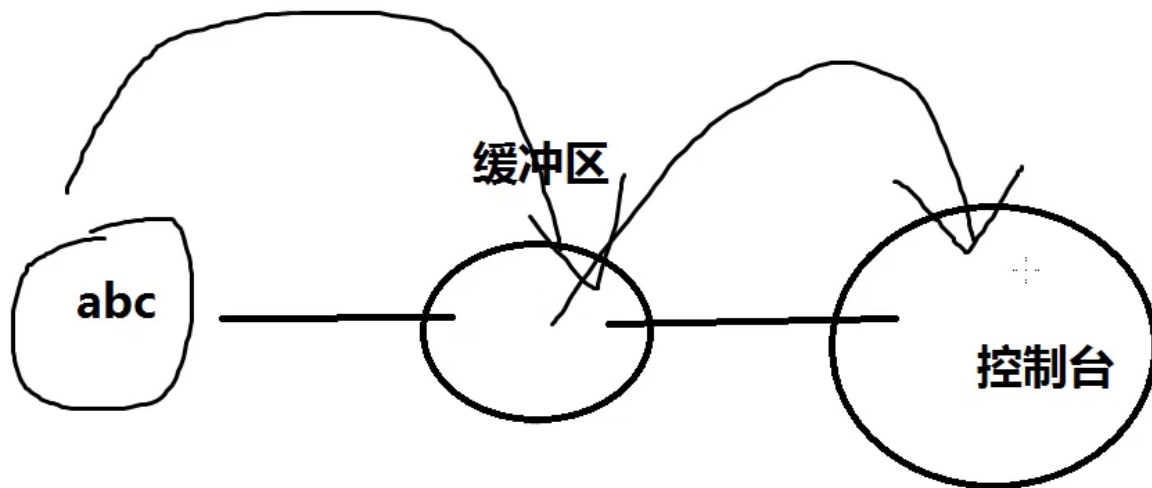
# 输出到文件中
f = open('test.txt', 'w')
# 重定向符号 >> 尖尖
print >> f, '测试字符串'

# 输出不自动换行
print '1', # 不换行
print '2',
print '3',

# 输出的各个数据，使用分隔符分割
# 默认使用空格分隔
# 需求：使用-分割
print '-'.join(['a', 'b', 'c'])
```

013. Python的输出-3.x版本

`flush` 参数



flush=True

什么时候把缓冲区输出到控制台？

有很多判定条件，其中有一个是你输出的内容当中有一个换行
如果有换行，立即输出到控制台

```
# -*- coding: utf-8 -*-
# Python 3.x

# 输出一个值
print(123)

# 输出一个变量
num = 55
print(num)

# 输出多个变量
num2 = 44
print(num, num2)

# 格式化输出
name = '团子'
age = 18
# 我的名字是团子，年龄是18
# '我的名字是%s，年龄是%d' % (name, age)
# 上面那个整体就相当于是一个字符串
# % 是属于字符串拼接的范畴
print('我的名字是%s，年龄是%d' % (name, age))

# .format 方法，字符串是一个对象，可以调用对象的方法
print('我的名字是{0}，年龄是{1}'.format(name, age))
# format 还可以使用名称或者下标的形式，后面补充

# 输出到文件中
# 'w' 是操作权限符
f = open('test.txt', 'w')
```

```

print('测试内容', file=f)
# 默认情况下是标准输出, 即输出到控制台
import sys
print('测试内容', file=sys.stdout)
f.close()

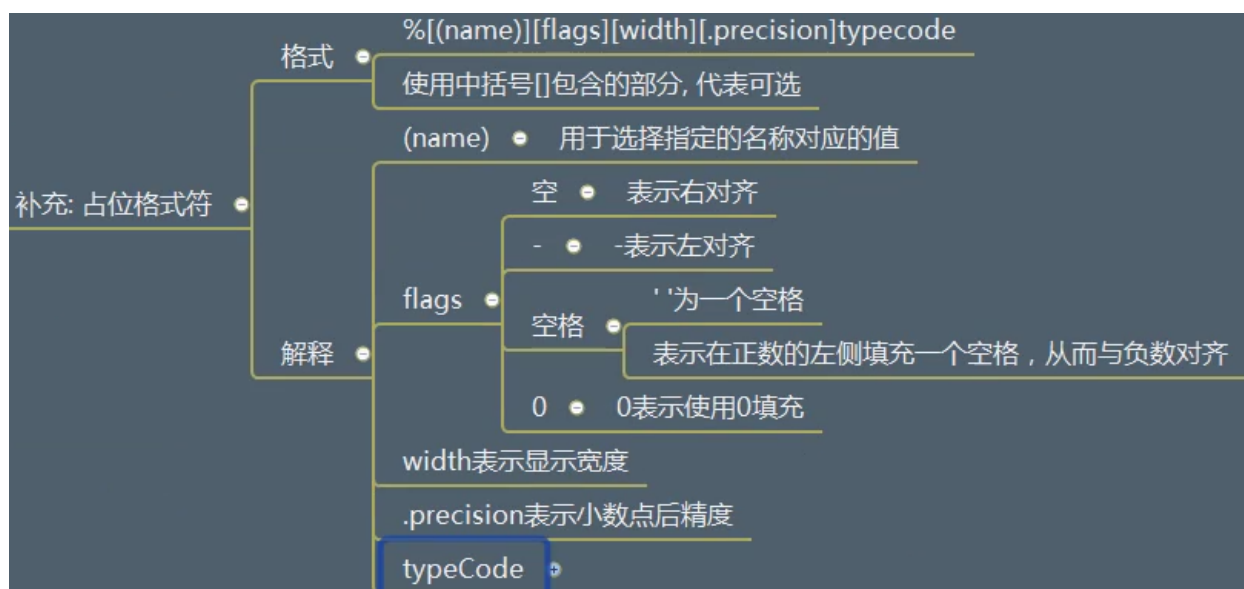
# 输出不自动换行
print('abc', end='')
print('def')

# 输出的各个数据, 使用分隔符分割
# 默认使用空格分隔
# 需求: 使用-分割
print('1', '2', '3', sep='-')

# flush 参数的说明
# print('请输入账号', end='')
print('请输入账号', end='', flush=True)
# 休眠 5s
# 睡眠的过程中什么都不做
from time import sleep
sleep(3)

```

014. Python的格式符(补充)



占位格式符完全体

`%[(键)][对齐方式][显示宽度][.精度]类型码`

typeCode	数值	i/d	#将整数、浮点数转换成十进制表示，并将其格式化到指定位置
		o	#将整数转换成八进制表示，并将其格式化到指定位置
		x	#将整数转换成十六进制表示，并将其格式化到指定位置
		e	#将整数、浮点数转换成科学计数法，并将其格式化到指定位置（小写e）
		E	#将整数、浮点数转换成科学计数法，并将其格式化到指定位置（大写E）
		f	#将整数、浮点数转换成浮点数表示，并将其格式化到指定位置（默认保留小数点后6位）
		F	#同上
		g	#自动调整将整数、浮点数转换成浮点型或科学计数法表示（超过6位数用科学计数法），并
		G	#自动调整将整数、浮点数转换成浮点型或科学计数法表示（超过6位数用科学计数法），并
		s	#获取传入对象的__str__方法的返回值，并将其格式化到指定位置
	字符串	r	#获取传入对象的__repr__方法的返回值，并将其格式化到指定位置
		c	#整数：将数字转换成其unicode对应的值，10进制范围为 0 <= i <= 1114111（py27则
	特殊	%	#当字符串中存在格式化标志时，需要用%%表示一个百分号
注：Python中百分号格式化是不存在自动将整数转换成二进制表示的方式			

	g	#自动调整将整数、浮点数转换成 浮点型或科学计数法表示（超过6位数用科学计数法），并将其格式化到指定位置（如果是科学计数则是e；）
	G	#自动调整将整数、浮点数转换成 浮点型或科学计数法表示（超过6位数用科学计数法），并将其格式化到指定位置（如果是科学计数则是E；）
字符串	s	#获取传入对象的__str__方法的返回值，并将其格式化到指定位置
	r	#获取传入对象的__repr__方法的返回值，并将其格式化到指定位置
	c	#整数：将数字转换成其unicode对应的值，10进制范围为 0 <= i <= 1114111（py27则只支持0-255）；字符：将字符添加到指定位置
特殊	%	#当字符串中存在格式化标志时，需要用 %%表示一个百分号
注：Python中百分号格式化是不存在自动将整数转换成二进制表示的方式		

```
# 格式化输出
name = '团子'
age = 18

# 我的名字是...，年龄是...
print('我的名字是%s，年龄是%d' % (name, age))

# %[(name)][flags][width][.precision]typecode

# typecode 类型码，就是 s， d 这种

# [] 可以省略的

# (name)
# 表示根据指定的名称(key)，查找对应的值，格式化到字符串中
math = 60
english = 100
print('我的数学成绩是%d，英文成绩是%d' % (math, english))
# 需求，调换两个分数
# 方法1：直接在后面调换
print('我的数学成绩是%d，英文成绩是%d' % (english, math))
# 方法2：借助(name)
print('我的数学成绩是%(math)d，英文成绩是%(en)d' % {'en': english, 'math': math})
# jpch89：字符串里面写%(key)类型码，格式符%后面跟着字典（字典外面不用加括号！）

# width表示占用的宽度
print('%10d' % math)
```

```

# flag表示对齐方式, 默认右对齐, 用空格填充
# - 表示左对齐
print('%-10d' % math)
# 空格 这样写可以对齐负数
print('% d' % -math)
print('% d' % math)
# 写两个空格还是一个空格
print('%  d' % math)

mini = 5
sec = 8
print('%02d:%02d' % (mini, sec))

score = 59.6
print('%d' % score) # 59, 截取整数部分, 类似于 int 类型转换的效果
print('%f' % score) # 默认小数点后6位
print('%.2f' % score)

# 类型码 typecode
# i/d 整数、浮点数变成十进制整数
print('%i' % score)
print('%i' % 0b11)
print('%d' % 0o10)
print('%d' % 0x10)

# o 转换成八进制数表示
print('%o' % 8)

# x 转换成十六进制数表示

# e 科学记数法, E 大写E的科学记数法
print('%e' % 16)

# g, G
print('%d' % 101.1) # 无法转换成 101.1
print('%g' % 101.1)
print('%g' % 101)
print('%g' % 1111111) # 超过6位自动转换成科学记数法

# c
print('%c' % 97)

# 需求: 输出 num%
num = 99
print('%d%%' % num) # %% 表示百分号

# 注意: 没有 %b!
print('%b' % 10) # 不支持!

```


完成于 201810281039