第 1 章 初识 Python

必背必记

1、print()—输出

print()函数的基本用法如下：

```
print(输出内容)
```

其中，输出内容可以是数字和字符串（使用引号括起来），此类内容将直接输出，也可以是包含运算符的表达式，此类内容将计算结果输出。例如：

```
a = 100                                  # 变量 a，值为 100
b = 5                                    # 变量 b，值为 5
print(9)                                 # 输出数字 9
print(a)                                 # 输出变量 a 的值 100
print(a*b)                               # 输出 a*b 的结果 500
```

通过 print()函数输出字符串时，如果想要换行，可以使用换行符“\n”。

2、.py

.py 是 Python 文件的扩展名。

3、IDLE 提供的常用快捷键

快 捷 键	说 明	适 用 于
F1	打开 Python 帮助文档	Python 文件窗口和 Shell 窗口均可用
F5	运行程序	仅 Python 文件窗口可用
Alt + /	自动补全前面曾经出现过的单词，如果之前有多个单词具有相同前缀，可以连续按下该快捷键，在多个单词中循环选择	Python 文件窗口和 Shell 窗口均可用
Alt + 3	注释代码块	仅 Python 文件窗口可用
Alt + 4	取消代码块注释	仅 Python 文件窗口可用
Alt + g	转到某一行	仅 Python 文件窗口可用
Ctrl + Z	撤销一步操作	Python 文件窗口和 Shell 窗口均可用
Ctrl + Shift + Z	恢复上一次的撤销操作	Python 文件窗口和 Shell 窗口均可用
Ctrl + S	保存文件	Python 文件窗口和 Shell 窗口均可用
Ctrl +]	缩进代码块	仅 Python 文件窗口可用
Ctrl + [	取消代码块缩进	仅 Python Shell 窗口可用

英语词汇

背记有法，让英语不再成为编程学习的拦路虎！

Python
含义：巨蛇，大蟒
热度：★★★★

Linux
词义：一个个人电脑上免费的 UNIX 操作系统
热度：★

File
含义：文件
热度：★★★★

New File
含义：新建文件
热度：★★★★

Run
含义：运行
热度：★★★★

Run Module
含义：运行程序
热度：★★★★

PyCharm
含义：一款 Python 开发工具
热度：★

Django
含义：Python 的 Web 开发框架
热度：★★

第 2 章 Python 语言基础

必背必记

1、转义字符

Python 中的字符串还支持转义字符。所谓转义字符是指使用反斜杠“\”对一些特殊字符进行转义。

转义字符	说　明
\	续行符
\n	换行符
\0	空
\t	水平制表符，用于横向跳到下一制表位
\"	双引号
\'	单引号
\\	一个反斜杠
\f	换页
\0dd	八进制数，dd 代表字符，如\012 代表换行
\xhh	十六进制数，hh 代表字符，如\x0a 代表换行

2、数据类型转换函数

在 Python 中，提供了如下表 所示的函数进行数据类型的转换。

函　数	作　用
int(x)	将 x 转换成整数类型
float(x)	将 x 转换成浮点数类型
complex(real [,imag])	创建一个复数
str(x)	将 x 转换为字符串
repr(x)	将 x 转换为表达式字符串
eval(str)	计算在字符串中的有效 Python 表达式，并返回一个对象
chr(x)	将整数 x 转换为一个字符
ord(x)	将一个字符 x 转换为它对应的整数值
hex(x)	将一个整数 x 转换为一个十六进制字符串
oct(x)	将一个整数 x 转换为一个八进制的字符串

3、赋值运算符

赋值运算符主要用来为变量等赋值。使用时，可以直接把基本赋值运算符“=”右边的值赋给左边的变量，也可以进行某些运算后再赋值给左边的变量。在 Python 中常用的赋值运算符如下表所示。

运 算 符	说　明	举　例	展 开 形 式
=	简单的赋值运算	x=y	x=y
+=	加赋值	x+=y	x=x+y
-=	减赋值	x-=y	x=x-y
*=	乘赋值	x*=y	x=x*y
/=	除赋值	x/=y	x=x/y
%=	取余数赋值	x%=y	x=x%y
**=	幂赋值	x**=y	x=x**y
//=	取整除赋值	x//=y	x=x//y

4、比较（关系）运算符

比较运算符，也称关系运算符，用于对变量或表达式的结果进行大小、真假等比较，如果比较结果为真，则返回 True，如果为假，则返回 False。比较运算符通常用在条件语句中作为判断的依据。Python 中的比较运算符如下表所示。

运 算 符	作　用	举　例	结　果
>	大于	'a' > 'b'	False
<	小于	156 < 456	True
==	等于	'c' == 'c'	True
!=	不等于	'y' != 't'	True
>=	大于或等于	479 >= 426	True
<=	小于或等于	62.45 <= 45.5	False

5、逻辑运算符

逻辑运算符是对真和假两种布尔值进行运算，运算后的结果仍是一个布尔值，Python 中的逻辑运算符主要包括 and（逻辑与）、or（逻辑或）、not（逻辑非）。

运 算 符	含　义	用　法	结 合 方 向
and	逻辑与	op1 and op2	从左到右
or	逻辑或	op1 or op2	从左到右
not	逻辑非	not op	从右到左

6、运算符的优先级

所谓运算符的优先级，是指在应用中哪一个运算符先计算，哪一个后计算，与数学的四则运算应遵循的“先乘除，后加减”是一个道理。

下表按从高到低的顺序列出了运算符的优先级。同一行中的运算符具有相同优先级，此时它们的结合方向决定求值顺序。

运　算　符	说　　明
**	幂
~、+、-	取反、正号和负号
*、/、%、//	算术运算符
+、-	算术运算符
<<、>>	位运算符中的左移和右移
&	位运算符中的位与
^	位运算符中的位异或
|	位运算符中的位或
<、<=、>、>=、!=、==	比较运算符

7、input()函数

在 Python 中，使用内置函数 input()可以接收用户的键盘输入。input()函数的基本用法如下：

```
variable = input(" 提示文字 ")
```

其中，variable 为保存输入结果的变量，双引号内的文字用于提示要输入的内容。

8、print()函数

默认的情况下，在Python中，使用内置的print()函数可以将结果输出到IDLE或者标准控制台上。其基本语法格式如下：

```
Print(输出内容)
```

其中，输出内容可以是数字和字符串（字符串需要使用引号括起来），此类内容将直接输出，也可以是包含运算符的表达式，此类内容将计算结果输出。

英语词汇

背记有法，让英语不再成为编程学习的拦路虎！

height
含义：高度
热度：★

weight
含义：重量
热度：★

keyword
含义：关键字
热度：★★

width
含义：宽度
热度：★

print
含义：打印，输出
热度：★★★★★

false
含义：假，错误的
热度：★★★★★

true
含义：真，正确的
热度：

title
含义：标题
热度：★

type
含义：类型
热度：★★★★★

Input
含义：输入
热度：★★★★★

typeerror
含义：类型错误
热度：★

float
含义：浮动
热度：★

file
含义：文件
热度：★

add
含义：添加
热度：★★★★★

none
含义：没有一个
热度：★

avg
含义：平均值
热度：★★

close
含义：关闭
热度：★★

第 3 章 流程控制语句

必背必记

1、if 语句

if 语句的基本用法如下：

```
if 表达式:
    语句块
```

其中，表达式可以是一个单纯的布尔值或变量，也可以是比较表达式或逻辑表达式（例如：a > band a != c），如果表达式为真，则执行“语句块”；如果表达式的值为假，就跳过“语句块”，继续执行后面的语句。

2、if...else 语句

if…else 语句的基本用法如下：

```
if 表达式:
    语句块 1
else:
    语句块 2
```

使用 if…else 语句时，表达式可以是一个单纯的布尔值或变量，也可以是比较表达式或逻辑表达式，如果满足条件，则执行 if 后面的语句块，否则，执行 else 后面的语句块。在使用 else 语句时，else 一定不可以单独使用，它必须和保留字 if 一起使用。

3、if...elif...else 语句

if…elif…else 语句的基本用法如下：

```
if 表达式 1:
    语句块 1
elif 表达式 2:
    语句块 2
elif 表达式 3:
    语句块 3
...
else:
    语句块 n
```

使用 if…elif…else 语句时，表达式可以是一个单纯的布尔值或变量，也可以是比较表达式或逻辑表达式，如果表达式为真，执行语句；而如果表达式为假，则跳过该语句，进行下一个 elif 的判断，只有在所有表达式都为假的情况下，才会执行 else 中的语句。

4、while 循环

while 语句的基本用法如下：

```
while 条件表达式:
    循环体
```

当条件表达式的返回值为真时，则执行循环体中的语句，执行完毕后，重新判断条件表达式的返回值，直到表达式返回的结果为假时，退出循环。

5、for 循环

for 语句的基本用法如下：

```
for 迭代变量 in 对象:
    循环体
```

其中，迭代变量用于保存读取出的值；对象为要遍历或迭代的对象，该对象可以是任何有序的序列对象，如字符串、列表和元组等；循环体为一组被重复执行的语句。

6、break 语句

在 while 语句中使用 break 语句的形式如下：

```
while 条件表达式 1:
    执行代码
    if 条件表达式 2:
        break
```

其中，条件表达式 2 用于判断何时调用 break 语句跳出循环。

在 for 语句中使用 break 语句的形式如下：

```
for 迭代变量 in 对象:
    if 条件表达式:
        break
```

其中，条件表达式用于判断何时调用 break 语句跳出循环。

7、continue 语句

在 while 语句中使用 continue 语句的形式如下：

```
while 条件表达式 1:
    执行代码
    if 条件表达式 2:
        continue
```

其中，条件表达式 2 用于判断何时调用 continue 语句跳出循环。

在 for 语句中使用 continue 语句的形式如下：

```
for 迭代变量 in 对象:
    if 条件表达式:
        continue
```

其中，条件表达式用于判断何时调用 continue 语句跳出循环。

英语词汇

背记有法，让英语不再成为编程学习的拦路虎！

if
含义：如果
热度：★★★★★

else
词义：否则
热度：★★★★★

while
含义：当……的时候
热度：★★★★★

for
含义：适合于
★★★★★

热度：

break
含义：中断
热度：★★★★★

continue
含义：继续
热度：★★★★★

pass
含义：通过
热度：★★

第 4 章 序列的应用

必背必记

1、切片

切片操作是访问序列中元素的另一种方法，它可以访问一定范围内的元素。通过切片操作可以生成一个新的序列。实现切片操作的语法格式如下：

```
sname[start : end : step]
```

参数说明：

sname：表示序列的名称。

start：表示切片的开始位置（包括该位置），如果不指定，则默认为0。

end：表示切片的截止位置（不包括该位置），如果不指定，则默认为序列的长度。

step：表示切片的步长，如果省略，则默认为 1，当省略该步长时，最后一个冒号也可以省略。

2、检查某个元素是否是序列的成员

在Python中，可以使用in关键字检查某个元素是否为序列的成员，即检查某个元素是否包含在某个序列中。语法格式如下：

```
value in sequence
```

其中，value 表示要检查的元素，sequence 表示指定的序列。

3、序列相关内置函数

函　　数	作　　用
list()	将序列转换为列表
str()	将序列转换为字符串
sum()	计算元素和
reversed()	反向序列中的元素
enumerate()	将序列组合为一个索引序列，多用在 for 循环中
sorted()	对元素进行排序

4、列表的创建和删除

（1）同其他类型的 Python 变量一样，创建列表时，也可以使用赋值运算符“=”直接将一个列表赋值给变量，语法格式如下：

```
listname = [element 1, element 2, element 3, …, element n]
```

（2）创建空列表

在 Python 中，也可以创建空列表，例如，要创建一个名称为 emptylist 的空列表，可以使用下面的代码：

```
emptylist = []
```

（3）创建数值列表

在 Python 中，可以使用 list() 函数直接将 range()函数循环出来的结果转换为列表。

list()函数的基本语法如下：

```
list(data)
```

其中，data 表示可以转换为列表的数据，其类型可以是 range 对象、字符串、元组或者其他可迭代类型的数据。

（4）删除列表

对于已经创建的列表，不再使用时，可以使用 del 语句将其删除。语法格式如下：

```
Del listname
```

其中，listname 为要删除列表的名称。

5、遍历列表

（1）直接使用 for 循环实现

直接使用 for 循环遍历列表，只能输出元素的值，语法格式如下：

```
for item in listname:
    # 输出 item
```

其中，item 用于保存获取到的元素值，要输出元素内容时，直接输出该变量即可；listname 为列表名称。

（2）使用 for 循环和 enumerate()函数实现

使用 for 循环和 enumerate()函数可以实现同时输出索引值和元素内容，语法格式如下：

```
for index，item in enumerate(listname):
    # 输出 index 和 item
```

参数说明：

index：用于保存元素的索引。

item：用于保存获取到的元素值，要输出元素内容时，直接输出该变量即可。

listname 为列表名称。

6、添加列表元素语法格式如下：

```
listname.append(obj)
```

上面介绍的是向列表中添加一个元素，如果想要将一个列表中的全部元素添加到另一个列表中，可以使用列表对象的 extend()方法实现。extend()方法的语法如下：

```
listname.extend(seq)
```

其中，listname 为要添加元素的列表名称，obj 为要添加到列表末尾的对象。seq 为要添加的列表。语句执行后，seq 的内容将追加到 listname 的后面。

7、对列表进行统计和计算

（1）获取指定元素出现的次数

使用列表对象的 count()方法可以获取指定元素在列表中的出现次数。基本语法格式如下：

```
listname.count(obj)
```

（2）获取指定元素首次出现的下标

使用列表对象的 index()方法可以获取指定元素在列表中首次出现的位置（即索引）。

基本语法格式如下：

```
listname.index(obj)
```

参数说明：

listname：表示列表的名称。

obj：表示要判断是否存在的对象，这里只能进行精确匹配，即不能是元素值的一部分。

（3）统计数值列表的元素和

在 Python 中，提供了 sum()函数用于统计数值列表中各元素的和。语法格式如下：

```
sum(iterable[,start])
```

参数说明：

iterable：表示要统计的列表。

start：表示统计结果是从哪个数开始（即将统计结果加上 start 所指定的数），是可选参数，如果没有指定，默认值为 0。

8、对列表进行排序

（1）使用列表对象的 sort()方法

列表对象提供了 sort()方法用于对原列表中的元素进行排序。排序后原列表中的元素顺序将发生改变。列表对象的 sort()方法的语法格式如下：

```
listname.sort(key=None,reverse=False)
```

参数说明：

listname：表示要进行排序的列表。

key：表示指定从每个元素中提取一个用于比较的键（例如，设置“key=str.lower”表示在排序时不区分字母大小写）。

reverse：可选参数，如果将其值指定为 True，则表示降序排列；如果为 False，则表示升序排列，默认为升序排列。

（2）使用内置的 sorted()函数实现

在 Python 中，提供了一个内置的 sorted()函数，用于对列表进行排序。使用该函数进行排序后，原列表的元素顺序不变。storted()函数的语法格式如下：

```
sorted(iterable,key=None,reverse=False)
```

参数说明：

iterable：表示要进行排序的列表名称。

key：表示指定从每个元素中提取一个用于比较的键（例如，设置“key=str.lower”表示在排序时不区分字母大小写）。

reverse：可选参数，如果将其值指定为 True，则表示降序排列；如果为 False，则表示升序排列，默认为升序排列。

9、列表推导式

（1）生成指定范围的数值列表，语法格式如下：

```
list = [Expression for var in range]
```

参数说明：

list：表示生成的列表名称。

Expression：表达式，用于计算新列表的元素。

var：循环变量。

range：采用 range()函数生成的 range 对象。

（2）根据列表生成指定需求的列表，语法格式如下：

```
newlist = [Expression for var in list]
```

参数说明：

newlist：表示新生成的列表名称。

Expression：表达式，用于计算新列表的元素。

var：变量，值为后面列表的每个元素值。

list：用于生成新列表的原列表。

（3）从列表中选择符合条件的元素组成新的列表，语法格式如下：

```
newlist = [Expression for var in list if condition]
```

参数说明：

newlist：表示新生成的列表名称。

Expression：表达式，用于计算新列表的元素。

var：变量，值为后面列表的每个元素值。

list：用于生成新列表的原列表。

condition：条件表达式，用于指定筛选条件。

10、元组的创建和删除

（1）使用赋值运算符直接创建元组

同其他类型的 Python 变量一样，创建元组时，也可以使用赋值运算符“=”直接将一个元组赋值给变量。语法格式如下：

```
tuplename = (element 1,element 2,element 3,…,element n)
```

其中，tuplename 表示元组的名称，可以是任何符合 Python 命名规则的标识符；elemnet 1、elemnet 2、elemnet 3、elemnet n 表示元组中的元素，个数没有限制。

（2）创建数值元组

在 Python 中，可以使用 tuple()函数直接将 range()函数循环出来的结果转换为数值元组。tuple()函数的基本语法如下：

```
tuple(data)
```

其中，data 表示可以转换为元组的数据，其类型可以是 range 对象、字符串、元组或者其他可迭代类型的数据。

（3）删除元组

对于已经创建的元组，不再使用时，可以使用 del 语句将其删除。语法格式如下：

```
del tuplename
```

其中，tuplename 为要删除元组的名称。

11、集合的创建

（1）直接使用“{}”创建集合

```
setname = {element 1,element 2,element 3,…,element n}
```

参数说明：

setname：表示集合的名称，可以是任何符合 Python 命名规则的标识符。

elemnet 1,elemnet 2,elemnet 3,…,elemnet n：表示集合中的元素，个数没有限制

（2）使用 set()函数创建

在 Python 中，可以使用 set()函数将列表、元组等其他可迭代对象转换为集合。set()函数的语法格式如下：

```
setname = set(iteration)
```

参数说明：

setname：表示集合名称。

iteration：表示要转换为集合的可迭代对象，可以是列表、元组、range 对象等，也可以是字符串。如果是字符串，返回的集合将是包含全部不重复字符的集合。

12、集合的添加和删除

（1）向集合中添加元素

向集合中添加元素可以使用 add()方法实现，语法格式如下：

```
setname.add(element)
```

参数说明：

setname：表示要添加元素的集合。

element：表示要添加的元素内容，只能使用字符串、数字及布尔类型的 True 或者 False 等，

不能使用列表、元组等可迭代对象。

（2）从集合中删除元素

在 Python 中，可以使用 del 命令删除整个集合，也可以使用集合的 pop()方法或者 remove()方法删除一个元素，或者使用集合对象的 clear()方法清空集合，即删除集合中的全部元素，使其变为空集合。

```
mr.remove('零基础学 Python')                    # 移除指定元素
print('使用 remove()方法移除指定元素后：',mr)
mr.pop()                                        # 删除一个元素
print('使用 pop()方法移除一个元素后：',mr)
mr.clear()                                      # 清空集合
print('使用 clear()方法清空集合后：',mr)
```

英语词汇

背记有法，让英语不再成为编程学习的拦路虎！

sorted
含义：分类，整理；挑选；[计算机]（根据指令的模式）把…分类；把…归类；排序的
热度：★

reversed
词义：推翻；（使）反转;（使）颠倒；（使）翻转；颠倒的
热度：★

enumerate
含义：列举，枚举，数
热度：★

import
含义：输入；进口；重要性；意义
热度：★★

append
含义：附加;添加;贴上
热度：★★

extend
含义：延伸；扩大；推广；延长；伸展；给予；发出；伸出；增加
热度：★★

price
含义：价格
热度：★

sale
含义：拍卖；卖，出卖；销售额，销售；销路
热度：★

set
含义：设置；放置，安置；集合；一套，一副
热度：★★★★★

del
含义：微分算子，倒三角形
热度：★★

remove
含义：去除；开除；脱掉，拿下
热度：★★

clear
含义：完全地；清晰地；;整整；扫除，除去；消除；使清楚；使干净
热度：★★

第 5 章 字符串及正则表达式

必背必记

1、len()函数

Len()函数的基本用法如下：

```
len(string)
```

其中，string 用于指定要进行长度统计的字符串。

2、切片法截取字符串

切片法截取字符串的语法格式如下：

```
string[start : end : step]
```

参数说明：

string：表示要截取的字符串。

start：表示要截取的第一个字符的索引（包括该字符），如果不指定，则默认为 0。

end：表示要截取的最后一个字符的索引（不包括该字符），如果不指定则默认为字符串的长度。

step：表示切片的步长，如果省略，则默认为 1，当省略该步长时，最后一个冒号也可以省略。

3、分割字符串

split()方法的语法格式如下：

```
str.split(sep, maxsplit)
```

参数说明：

str：表示要进行分割的字符串。

sep：用于指定分隔符，可以包含多个字符，默认为 None，即所有空字符（包括空格、换行“\n”、制表符“\t”等）。

maxsplit：可选参数，用于指定分割的次数，如果不指定或者为 -1，则分割次数没有限制，否则返回结果列表的元素个数，个数最多为 maxsplit+1。

返回值：分隔后的字符串列表。

4、合并字符串

join()方法的语法格式如下：

```
strnew = string.join(iterable)
```

参数说明：

strnew：表示合并后生成的新字符串。

string：字符串类型，用于指定合并时的分隔符。

iterable：可迭代对象，该迭代对象中的所有元素（字符串表示）将被合并为一个新的字符串。string 作为边界点分割出来。

5、常用的格式化字符

格式化字符	说 明
%s	字符串（采用 str()显示）
%c	单个字符
%d 或者%i	十进制整数
%x	十六进制整数
%f 或者%F	浮点数
%r	字符串（采用 repr()显示）
%o	八进制整数
%e	指数（基底写为 e）
%E	指数（基底写为 E）
%%	字符%

6、format()方法中常用的格式化字符

格式化字符	说 明
s	对字符串类型格式化
d	十进制整数
c	将十进制整数自动转换成对应的 Unicode 字符
e 或者 E	转换为科学计数法表示再格式化
g 或者 G	自动在 e 和 f 或者 E 和 F 中切换
b	将十进制整数自动转换成二进制表示再格式化
o	将十进制整数自动转换成八进制表示再格式化
x 或者 X	将十进制整数自动转换成十六进制表示再格式化
f 或者 F	转换为浮点数（默认小数点后保留 6 位）再格式化
%	显示百分比（默认显示小数点后 6 位）

7、使用 encode()方法编码

encode()方法的语法格式如下：

```
str.encode([encoding="utf-8"][,errors="strict"])
```

参数说明：

str：表示要进行转换的字符串。

encoding="utf-8"：可选参数，用于指定进行转码时采用的字符编码，默认为 UTF-8，如果想使用简体中文，也可以设置为 gb2312。当只有这一个参数时，也可以省略前面的“encoding=”，直接写编码。

errors="strict"：可选参数，用于指定错误处理方式，其可选择值可以是 strict（遇到非法字符就抛出异常）、ignore（忽略非法字符）、replace（用“?”替换非法字符）或 xmlcharref replace（使用 XML 的字符引用）等，默认值为 strict。

8、使用 decode()方法解码

decode()方法的语法格式如下：

```
bytes.decode([encoding="utf-8"][,errors="strict"])
```

参数说明：

bytes：表示要进行转换的二进制数据，通常是 encode()方法转换的结果。

encoding="utf-8"：可选参数，用于指定进行解码时采用的字符编码，默认为 UTF-8，如果想使用简体中文，也可以设置为 gb2312。当只有这一个参数时，也可以省略前面的“encoding=”，直接写编码。

9、常用元字符

代 码	说 明
.	匹配除换行符以外的任意字符
\w	匹配字母或数字或下划线或汉字
\s	匹配任意的空白符
\d	匹配数字
\b	匹配单词的开始或结束
^	匹配字符串的开始
$	匹配字符串的结束

10、常用限定符

限 定 符	说 明	举 例
?	匹配前面的字符零次或一次	colou?r，该表达式可以匹配 colour 和 color
+	匹配前面的字符一次或多次	go+gle，该表达式可以匹配的范围从 gogle 到 goo…gle
*	匹配前面的字符零次或多次	go*gle，该表达式可以匹配的范围从 ggle 到 goo…gle
{n}	匹配前面的字符 n 次	go{2}gle，该表达式只匹配 google
{n,}	匹配前面的字符最少 n 次	go{2,}gle，该表达式可以匹配的范围从 google 到 goo…gle
{n,m}	匹配前面的字符最少 n 次，最多 m 次	employe{0,2}，该表达式可以匹配 employ、employe 和 employee 3 种情况

英语词汇

背记有法，让英语不再成为编程学习的拦路虎！

string
含义：字符串
热度：★★★★★

split
词义：分割
热度：★★

join
含义：连接
热度：★★

count
含义：计数
热度：★★

find
含义：查找
热度：★★

index
含义：索引
热度：★★

lower
含义：低级的
热度：★★

upper
含义：高级的
热度：★★

strip
含义：去除
热度：★★

lstrip
含义：只去掉左边的
热度：★★

rstrip
含义：只去掉右边的
热度：★★

format
含义：格式化
热度：★★

encode
含义：编码
热度：★★

decode
含义：解码
热度：★★

match
含义：匹配
热度：★★

search
含义：搜索
热度：★★

sub
含义：代替
热度：★★

第 6 章 函数

必背必记

1、创建函数

创建函数使用 def 关键字实现，具体的语法格式如下：

```
def functionname([parameterlist]):
    ['''comments''']
    [functionbody]
```

参数说明：

functionname：函数名称，在调用函数时使用。

parameterlist：可选参数，用于指定向函数中传递的参数。如果有多个参数，各参数间使用逗号“,”分隔。如果不指定，则表示该函数没有参数，在调用时也不指定参数。

'''comments''' ：可选参数，表示为函数指定注释，注释的内容通常是说明该函数的功能、要传递的参数的作用等，可以为用户提供友好提示和帮助的内容。

functionbody ：可选参数，用于指定函数体，即该函数被调用后，要执行的功能代码。如果函数有返回值，可以使用 return 语句返回。

2、调用函数

调用函数也就是执行函数，基本语法格式如下：

```
functionname([parametersvalue])
```

参数说明：

functionname：函数名称，要调用的函数名称必须是已经创建好的。

parametersvalue：可选参数，用于指定各个参数的值。如果需要传递多个参数值，则各参数值间使用逗号“,”分隔。如果该函数没有参数，则直接写一对小括号即可。

3、为参数设置默认值

定义带有默认值参数的函数的语法格式如下：

```
def functionname(…,[parameter1 = defaultvalue1]):
    [functionbody]
```

参数说明：

functionname：函数名称，在调用函数时使用。

parameter1 = defaultvalue1：可选参数，用于指定向函数中传递的参数，并且为该参数设置默认值为 defaultvalue1。

functionbody ：可选参数，用于指定函数体，即该函数被调用后，要执行的功能代码。

4、返回值

return 语句的语法格式如下：

```
return [value]
```

参数说明：

value：可选参数，用于指定要返回的值，可以返回一个值，也可返回多个值。

5、匿名函数

匿名函数是指没有名字的函数，在 Python 中，使用 lambda 表达式创建匿名函数，其语法格式如下：

```
result = lambda [arg1 [,arg2,……,argn]]:expression
```

参数说明：

result：用于调用 lambda 表达式。

[arg1 [,arg2,……,argn]]：可选参数，用于指定要传递的参数列表，多个参数间使用逗号“,”分隔。

expression：必选参数，用于指定一个实现具体功能的表达式。如果有参数，那么在该表达式中将应用这些参数。

英语词汇

背记有法，让英语不再成为编程学习的拦路虎！

TypeError
含义：类型错误
热度：★★

parameter
词义：参数
热度：★★

global
含义：总体的
热度：★★

第 7 章 面向对象程序设计

必背必记

1、定义类

在 Python 中，类的定义使用 class 关键字来实现，语法如下：

```
class ClassName:
    '''类的帮助信息'''                    # 类文档字符串
    statement                             # 类体
```

参数说明：

ClassName：用于指定类名，一般使用大写字母开头，如果类名中包括两个单词，第二个单词的首字母也大写，这种命名方法也称为“驼峰式命名法”，这是惯例。当然，也可根据自己的习惯命名，但是一般推荐按照惯例来命名。

statement：类体，主要由类变量（或类成员）、方法和属性等定义语句组成。如果在定义类时，没想好类的具体功能，也可以在类体中直接使用 pass 语句代替。

2、创建类的实例

class 语句本身并不创建该类的任何实例。所以在类定义完成以后，可以创建类的实例，即实例化该类的对象。创建类的实例的语法如下：

```
classname(parameterlist)
```

其中，ClassName 是必选参数，用于指定具体的类；parameterlist 是可选参数，当创建一个类时，没有创建__init__()方法，或者__init__()方法只有一个 self 参数时，parameterlist 可以省略。

3、创建类的成员并访问

所谓实例方法是指在类中定义的函数。该函数是一种在类的实例上操作的函数。 实例方法的第一个参数必须是 self，并且必须包含一个 self 参数。创建实例方法的语法格式如下：

```
def functionName(self,parameterlist):
    block
```

参数说明：

functionName：用于指定方法名，一般使用小写字母开头。

self：必要参数，表示类的实例，其名称可以是 self 以外的单词，使用 self 只是一个惯例而已。

parameterlist：用于指定除 self 参数以外的参数，各参数间使用逗号“,”进行分隔。

block：方法体，实现的具体功能。

实例方法创建完成后，可以通过类的实例名称和点（.）操作符进行访问，语法格式如下：

```
instanceName.functionName(parametervalue)
```

参数说明：

instanceName：为类的实例名称。

functionName：为要调用的方法名称。

parametervalue：表示为方法指定对应的实际参数，其值的个数与创建实例方法中 parameterlist 的个数相同。

4、创建用于计算的属性

通过@property 创建用于计算的属性的语法格式如下：

```
@property
def methodname(self):
    block
```

参数说明：

methodname：用于指定方法名，一般使用小写字母开头。该名称最后将作为创建的属性名。

self：必要参数，表示类的实例。

block：方法体，实现的具体功能。在方法体中，通常以 return 语句结束，用于返回计算结果。

5、继承的基本语法

通过继承不仅可以实现代码的重用，还可以通过继承来理顺类与类之间的关系。在 Python 中，可以在类定义语句中，类名右侧使用一对小括号将要继承的基类名称括起来，从而实现类的继承。具体的语法格式如下：

```
class ClassName(baseclasslist):
    '''类的帮助信息'''                    #类文档字符串
    statement                             #类体
```

参数说明：

ClassName：用于指定类名。

baseclasslist：用于指定要继承的基类，可以有多个，类名之间用逗号“,”分隔。如果不指定，将使用所有 Python 对象的根类 object。

''' 类的帮助信息 ''' ：用于指定类的文档字符串，定义该字符串后，在创建类的对象时，输入类名和左侧的括号“(”后，将显示该信息。

statement：类体，主要由类变量（或类成员）、方法和属性等定义语句组成。如果在定义类时，没想好类的具体功能，也可以在类体中直接使用 pass 语句代替。

6、方法重写

基类中定义的 harvest()方法，无论派生类是什么水果都显示“水果……”，如果想

要针对不同水果给出不同的提示，可以在派生类中重写 harvest()方法。例如，在创建派生类 Orange 时，重写 harvest()方法的代码如下：

```
01 class Orange(Fruit):                                    # 定义橘子类（派生类）
02     color = "橙色"
03     def __init__(self):
04         print("\n 我是橘子")
05     def harvest(self, color):
06         print("橘子是：" + color + "的！")               # 输出的是形式参数 color
07         print("橘子已经收获……")
08         print("橘子原来是：" + Fruit.color + "的！");# 输出的是类属性 color
```

英语词汇

背记有法，让英语不再成为编程学习的拦路虎！

parameter
含义：参数;参量
热度：★★

init
含义：初始化
热度：★★

self
含义：自己;本人;本性;私利
同一的;纯净的;单一的
热度：★

function
含义：功能，作用;应变量，
函数;职务
热度：★★

block
含义：块;街区;阻止;阻塞;限制
热度：★

protected
含义：防护;保护
热度：★★★

private
含义：私有的，私人的;秘密的
热度：★★★★★

harvest
含义：收割，收成
热度 ★★★

第 8 章 模块

必背必记

1、使用 import 语句导入模块

import 语句的基本语法格式如下：

```
import modulename [as alias]
```

其中，modulename 为要导入模块的名称；[as alias]为给模块起的别名，通过该别名也可以使用模块。

2、使用 from...import 语句导入模块

from…import 语句的语法格式如下：

```
from modelname import member
```

参数说明：

modelname：模块名称，区分字母大小写，需要和定义模块时设置的模块名称的大小写保持一致。

member：用于指定要导入的变量、函数或者类等。可以同时导入多个定义，各个定义之间使用逗号“,”分隔。如果想导入全部定义，也可以使用通配符星号“*”代替。

3、Python 常用的内置标准模块

模 块 名	描 述
sys	与 Python 解释器及其环境操作相关的标准库
time	提供与时间相关的各种函数的标准库
os	提供了访问操作系统服务功能的标准库
calendar	提供与日期相关的各种函数的标准库
urllib	用于读取来自网上（服务器上）的数据的标准库
json	用于使用 JSON 序列化和反序列化对象
re	用于在字符串中执行正则表达式匹配和替换
math	提供算术运算函数的标准库
decimal	用于进行精确控制运算精度、有效数位和四舍五入操作的十进制运算
shutil	用于进行高级文件操作，如复制、移动和重命名等
logging	提供了灵活的记录事件、错误、警告和调试信息等日志信息的功能
tkinter	使用 Python 进行 GUI 编程的标准库

英语词汇

背记有法，让英语不再成为编程学习的拦路虎！

Modules
含义：模块
热度：★★

import
词义：输入
热度：★★★

random
含义：随机的
热度：★★

第 9 章 异常处理及程序调试

必背必记

1、Python 中常见的异常

异 常	描 述
NameError	尝试访问一个没有声明的变量引发的错误
IndexError	索引超出序列范围引发的错误
IndentationError	缩进错误
ValueError	传入的值错误
KeyError	请求一个不存在的字典关键字引发的错误
IOError	输入输出错误（如要读取的文件不存在）
ImportError	当 import 语句无法找到模块或 from 无法在模块中找到相应的名称时引发的错误
AttributeError	尝试访问未知的对象属性引发的错误
TypeError	类型不合适引发的错误
MemoryError	内存不足
ZeroDivisionError	除数为 0 引发的错误

2、try...except 语句

在 Python 中，提供了 try…except 语句捕获并处理异常，具体的语法格式如下：

```
try:
    block1
except [ExceptionName [as alias]]:
    block2
```

参数说明：

block1：表示可能出现错误的代码块。

ExceptionName [as alias]：可选参数，用于指定要捕获的异常。其中，ExceptionName 表示要捕获的异常名称，如果在其右侧加上 as alias，则表示为当前的异常指定一个别名，通过该别名，可以记录异常的具体内容。

3、try...except...finally 语句

完整的异常处理语句应该包含 finally 代码块，其语法格式如下：

```
try:
    block1
except [ExceptionName [as alias]]:
    block2
finally:
```

```
    block3
```

对于 try…except…finally 语句的理解并不复杂，它只是比 try…except 语句多了一个 finally 语句，如果程序中有一些在任何情形中都必须执行的代码，那么就可以将它们放在 finally 代码块中。

4、使用 raise 语句抛出异常

如果某个函数或方法可能会产生异常，但不想在当前函数或方法中处理这个异常，则可以使用 raise 语句在函数或方法中抛出异常。raise 语句的语法格式如下：

```
raise [ExceptionName[(reason)]]
```

其中，ExceptionName[(reason)]为可选参数，用于指定抛出的异常名称以及异常信息的相关描述。如果省略，就会把当前的错误原样抛出。

5、使用 assert 语句调试程序

Python 提供了 assert 语句来调试程序，assert 语句的基本语法如下：

```
assert expression [,reason]
```

参数说明：

expression：条件表达式，如果该表达式的值为真时，什么都不做，如果为假时，则抛出 AssertionError 异常。

reason：可选参数，用于对判断条件进行描述，为了以后更好地知道哪里出现了问题。

英语词汇

背记有法，让英语不再成为编程学习的拦路虎！

try
含义：试图
热度：★★★

except
词义：除……之外
热度：★★★

finally
含义：最后
热度：★★★

raise
含义：提高
热度：★★

assert
含义：维护
热度：★★

第 10 章 文件及目录操作

必背必记

1、创建和打开文件

在 Python 中，想要操作文件需要先创建或者打开指定的文件并创建文件对象，可以通过内置的 open()函数实现。open()函数的基本语法格式如下：

```
file = open(filename[,mode[,buffering]])
```

参数说明：

file：被创建的文件对象。

filename：要创建或打开文件的文件名称，需要使用单引号或双引号括起来。如果要打开的文件和当前文件在同一个目录下，那么直接写文件名即可，否则需要指定完整路径。例如，要打开当前路径下的名称为 status.txt 的文件，可以使用“status.txt”。

mode：可选参数，用于指定文件的打开模式。默认的打开模式为只读（即 r）。

buffering：可选参数，用于指定读写文件的缓冲模式，值为 0 表达式不缓存；值为 1 表示缓存；如果大于 1，则表示缓冲区的大小。默认为缓存模式。

2、关闭文件

关闭文件可以使用文件对象的 close()方法实现。close()方法的语法格式如下：

```
file.close()
```

其中，file 为打开的文件对象。

3、写入文件内容

Python 的文件对象提供了 write()方法，可以向文件中写入内容。write()方法的语法格式如下：

```
file.write(string)
```

其中，file 为打开的文件对象；string 为要写入的字符串。

4、删除文件

Python 没有内置删除文件的函数，但是在内置的 os 模块中提供了删除文件的函数 remove()，该函数的基本语法格式如下：

```
os.remove(path)
```

其中，path 为要删除的文件路径，可以使用相对路径，也可以使用绝对路径。

5、重命名文件和目录

os 模块提供了重命名文件和目录的函数 rename()，基本语法格式如下：

```
os.rename(src,dst)
```

其中，src 用于指定要进行重命名的目录或文件；dst 用于指定重命名后的目录或文件。

6、stat()函数返回的对象的常用属性

属 性	说 明
st_mode	保护模式
st_ino	索引号
st_nlink	硬链接号（被连接数目）
st_size	文件大小，单位为字节
st_mtime	最后一次修改时间
st_dev	设备名
st_uid	用户 ID
st_gid	组 ID
st_atime	最后一次访问时间
st_ctime	最后一次状态变化的时间（系统不同返回结果也不同，例如，在 Windows 操作系统下返回的是文件的创建时间）

英语词汇

背记有法，让英语不再成为编程学习的拦路虎！

file
含义：文件
热度：★★

open
词义：打开
热度：★★

close
含义：关闭
热度：★★

with
含义：伴随
热度：★★

write
含义：写
热度：★★

read
含义：读取
热度：★★

seek
含义：寻找
热度：★★

exist
含义：存在
热度：★★

mkdir
含义：创建一个新的子目录
热度：★★

walk
含义：走
热度：★★

rename
含义：重新命名
热度：★★

stat
含义：静
热度：★★

第 11 章 使用 Python 操作数据库

必背必记

1、连接对象

（1）获取连接对象

如何获取连接对象呢？这就需要使用connect()函数。该函数有多个参数，具体使用哪个参数，取决于使用的数据库类型。例如，需要访问Oracle数据库和MySQL数据库，则必须同时下载Oracle和MySQL数据库模块。这些模块在获取连接对象时，都需要使用connect()函数。connect()函数常用的参数及说明如下表所示。

参　数	说　明
dsn	数据源名称，给出该参数表示数据库依赖
user	用户名
password	用户密码
host	主机名
database	数据库名称

（2）连接对象的方法

Connect()函数返回连接对象。这个对象表示目前和数据库的会话，连接对象支持的方法如下表所示。

方 法 名	说　明
close()	关闭数据库连接
commit()	提交事务
rollback()	回滚事务
cursor()	获取游标对象，操作数据库，如执行 DML 操作，调用存储过程等

2、游标对象

游标对象（Cursor Object）代表数据库中的游标，用于指示抓取数据操作的上下文，主要提供执行SQL语句、调用存储过程、获取查询结果等方法。

如何获取游标对象呢？通过使用连接对象的cursor()方法，可以获取到游标对象。游标对象的属性如下所示：

description：数据库列类型和值的描述信息。

rowcount：回返结果的行数统计信息，如SELECT、UPDATE、CALLPROC等。

游标对象的方法如下表所示。

方 法 名	说 明
callproc(procname,[,parameters])	调用存储过程，需要数据库支持
close()	关闭当前游标
execute(operation[, parameters])	执行数据库操作，SQL 语句或者数据库命令
executemany(operation,seq_of_params)	用于批量操作，如批量更新
fetchone()	获取查询结果集中的下一条记录
fetchmany(size)	获取指定数量的记录
fetchall()	获取结果集的所有记录
nextset()	跳至下一个可用的结果集
arraysize	指定使用 fetchmany()获取的行数，默认为 1
setinputsizes(sizes)	设置在调用 execute*()方法时分配的内存区域大小
setoutputsize(sizes)	设置列缓冲区大小，对大数据列（如 LONGS 和 BLOBS）尤其有用

3、操作SQLite

1、新增用户数据信息

为了向数据表中新增数据，可以使用如下SQL语句：

```
Insert into 表名(字段名1, 字段名2, …, 字段名n)
values(字段名1, 字段名2,…, 字段名n)
```

2、查看用户数据信息

查找user表中的数据可以使用如下SQL语句：

```
select   字段名1,字段名2,字段名3,…from 表名  where 查询条件
```

查看用户信息的代码与插入数据信息大致相同，不同点在于使用的SQL语句不同。此外，查询数据时通常使用如下3种方式：

fetchone()：获取查询结果集中的下一条记录。

fetchmany(size)：获取指定数量的记录。

fetchall()：获取结构集的所有记录。

3、修改用户数据信息

修改user表中的数据可以使用如下SQL语句：

```
update 表名 set 字段名 = 字段值  where 查询条件
```

4、删除用户数据信息

查找user表中的数据可以使用如下SQL语句：

```
delete from 表名  where 查询条件
```

5、操作MySQL 数据表

在向表中插入数据时，可以使用excute()方法添加一条记录，也可以使executemany()方法批量添加多条记录，executemany()方法的语法格式如下：

```
executemany(operation,seq_of_params)
```

operation：操作的SQL语句。

seq_of_params：参数序列。

英语词汇

背记有法，让英语不再成为编程学习的拦路虎！

Connection
词义：连接；联系，关系；连接点
热度：★★★★★

insert
词义：插入；嵌入；添加；加插
热度：★★★★★

into
词义：进入…中
热度：★★★★

execute
词义：执行；完成；履行
热度：★★★★

result
词义：结果
热度：★★★★★

update
词义：更新，使现代化；校正，修正
热度：★★★★★

root
词义：根，根源;原因，本质
热度：★

第 12 章 GUI 界面编程

必背必记

1、流行的 GUI 工具包

工 具 包	描 述
wxPython	wxPython 是 Python 语言的一套优秀的 GUI 图形库，允许 Python 程序员很方便地创建完整的、功能键全的 GUI 用户界面
Kivy	Kivy 是一个开源工具包能够让使用相同源代码创建的程序能跨平台运行。它主要关注创新型用户界面开发，如多点触摸应用程序
Flexx	Flexx 是一个纯 Python 工具包，用来创建图形化界面应用程序，可使用 Web 技术进行界面的渲染
PyQt	PyQt 是 Qt 库的 Python 版本，支持跨平台
Tkinter	Tkinter（也叫 Tk 接口）是 Tk 图形用户界面工具包标准的 Python 接口。Tk 是一个轻量级的跨平台图形用户界面（GUI）开发工具
Pywin32	Windows Pywin32 允许你像 VC 一样的形式来使用 Python 开发 win32 应用
PyGTK	PyGTK 让你用 Python 轻松创建具有图形用户界面的程序
pyui4win	pyui4win 是一个开源的采用自绘技术的界面库

2、使用 wx.Frame 框架

wx.Frame 构造器的语法格式如下：

```
wx.Frame(parent, id=-1, title="", pos=wx.DefaultPosition, size=wx.DefaultSize,
    style=wx.DEFAULT_FRAME_STYLE, name="frame")
```

参数说明：

parent：框架的父窗口。如果是顶级窗口，这个值是 None。

id：关于新窗口的 wxPython ID 号。通常设为-1，让 wxPython 自动生成一个新的 ID。

title：窗口的标题。

pos：一个 wx.Point 对象，它指定这个新窗口的左上角在屏幕中的位置。在图形用户界面程序中，通常(0,0) 是显示器的左上角。这个默认值(−1,−1) 将让系统决定窗口的位置。

size：一个 wx.Size 对象，它指定这个窗口的初始尺寸。这个默认值 (−1,−1) 将让系统决定窗口的初始尺寸。

style：指定窗口的类型的常量。可以使用或运算来组合它们。

name：框架内在的名字。可以使用它来寻找这个窗口。

3、StaticText 文本类

wx.StaticText 类的构造函数语法格式如下：

```
wx.StaticText(parent, id, label, pos=wx.DefaultPosition,size=wx.DefaultSize,
    style=0, name="staticText")
```

参数说明：

parent：父窗口部件。

id：标识符。使用-1 可以自动创建一个唯一的标识。

label：显示在静态控件中的文本内容。

pos：一个 wx.Point 或一个 Python 元组，它是窗口部件的位置。

size：一个 wx.Size 或一个 Python 元组，它是窗口部件的尺寸。

style：样式标记。

name：对象的名字。

4、TextCtrl 输入文本类

wx.TextCtrl 类的构造函数语法格式如下：

```
wx.TextCtrl(parent, id, value = "", pos=wx.DefaultPosition, size=wx.DefaultSize,
style=0,validator=wx.DefaultValidator name=wx.TextCtrlNameStr)
```

参数说明：

style：单行 wx.TextCtrl 的样式，取值及说明如下：

wx.TE_CENTER：控件中的文本居中。

wx.TE_LEFT：控件中的文本左对齐。

wx.TE_NOHIDESEL：文本始终高亮显示，只适用于 Windows。

wx.TE_PASSWORD：不显示所键入的文本，以星号（*）代替显示。

wx.TE_PROCESS_ENTER ：如果使用改参数，那么当用户在控件内按下<Enter>键时，一个文本输入事件将被触发。否则，按键事件由该文本控件或该对话框管理。

wx.TE_PROCESS_TAB：如果指定了这个样式，那么通常的字符事件在按下<Tab>键时创建（一般意味一个制表符将被插入文本）。否则，tab 由对话框来管理，通常是控件间的切换。

wx.TE_READONLY：文本控件为只读，用户不能修改其中的文本。

wx.TE_RIGHT：控件中的文本右对齐。

value：显示在该控件中的初始文本。

validator：常用于过滤数据以确保只能键入要接受的数据。

5、Button 按钮类

wx.Button 的构造函数的语法如下：

```
wx.Button(parent, id, label, pos, size=wxDefaultSize, style=0, validator,
name="button")
```

wx.Button 的参数与 wx.TextCtrl 的参数基本相同，其中参数 label 是显示在按钮上的文本。

6、wxPython 的 sizer 说明

sizer 名称	描 述
BoxSizer	在一条水平或垂直线上的窗口部件的布局。当尺寸改变时，控制窗口部件的行为上很灵活。通常用于嵌套的样式，可用于几乎任何类型的布局

GridSizer	一个十分基础的网格布局。当你要放置的窗口部件都是同样的尺寸且整齐地放入一个规则的网格中可以使用它
FlexGridSizer	对 GridSizer 稍微做了些改变，当窗口部件有不同的尺寸时，可以有更好的结果
GridBagSizer	GridSizer 系列中最灵活的成员。使得网格中的窗口部件可以随意放置
StaticBoxSizer	一个标准的 Box Sizer。带有标题和环线

英语词汇

背记有法，让英语不再成为编程学习的拦路虎！

frame
含义：框架
热度：★★

parent
词义：父，上层
热度：★★

title
含义：标题
热度：★★

size
含义：尺寸
热度：★★

name
含义：名字
热度：★★

label
含义：标签
热度：★★

button
含义：按钮
热度：★★

add
含义：添加
热度：★★

第 13 章 Pygame 游戏编程

必背必记

1、Pygame 常用模块

Pygame做游戏开发的优势在于不需要过多考虑与底层开发相关的内容，而可以把工作重心放在游戏逻辑上。例如，Pygame中集成了很多和底层开发相关的模块，如访问显示设备、管理事件、使用字体等。Pygame常用模块如下表所示。

模 块 名	功 能
pygame.cdrom	访问光驱
pygame.cursors	加载光标
pygame.display	访问显示设备
pygame.draw	绘制形状、线和点
pygame.event	管理事件
pygame.font	使用字体
pygame.image	加载和存储图片
pygame.joystick	使用游戏手柄或者类似的东西
pygame.key	读取键盘按键
pygame.mixer	声音
pygame.mouse	鼠标
pygame.movie	播放视频
pygame.music	播放音频
pygame.overlay	访问高级视频叠加
pygame.rect	管理矩形区域
pygame.sndarray	操作声音数据
pygame.sprite	操作移动图像
pygame.surface	管理图像和屏幕
pygame.surfarray	管理点阵图像数据
pygame.time	管理时间和帧信息
pygame.transform	缩放和移动图像

2、display 模块的常用方法如下表所示

方 法 名	功 能
pygame.dispaly.init	初始化 display 模块
pygame.dispaly.quit	结束 display 模块

pygame.dispaly.get_init	如果 display 模块已经被初始化，则返回 True
pygame.dispaly.set_mode	初始化一个准备显示的界面
pygame.dispaly.get_surface	获取当前的 Surface 对象
pygame.dispaly.flip	更新整个待显示的 Surface 对象到屏幕上
pygame.dispaly.update	更新部分内容显示到屏幕上，如果没有参数，则与 flip 功能相同

3、Surface 对象的常用方法如下表所示

方 法 名	功 能
pygame.Surface.blit	将一个图像画到另一个图像上
pygame.Surface.convert	转换图像的像素格式
pygame.Surface.convert_alpha	转化图像的像素格式，包含 alpha 通道的转换
pygame.Surface.f ill	使用颜色填充 Surface
pygame.Surface.get_rect	获取 Surface 的矩形区域

英语词汇

display
含义：显示；陈列；展开，伸展
热度：★★★

event
词义：事件，大事；活动，经历；结果
热度：★★★

screen
词义：屏幕
热度：★★

image
词义：影像；肖像；概念，意向；镜像，映像
热度：★★

load
词义：加载；装载
热度：★★

surface
词义：表面；外观，外表
热度：★

flip
词义：轻弹，轻击；按（开关）
热度：★

speed
词义：速度；快速
热度：★★

colliderect
词义：碰撞的
热度：★★

第 14 章 网络爬虫开发

必背必记

1、urllib 中的子模块

模 块 名 称	说 明
urllib.request	该模块定义了打开 URL（主要是 HTTP）的方法和类，例如，身份验证、重定向、cookie 等等
urllib.error	该模块中主要包含异常类，基本的异常类是 URLError
urllib.parse	该模块定义的功能分为两大类：URL 解析和 URL 引用
urllib.robotparser	该模块用于解析 robots.txt 文件

2、Scrapy 爬虫框架

Scrapy 框架是一套比较成熟的 Python 爬虫框架，简单轻巧，并且非常方便，可以高效率地爬取 Web 页面并从页面中提取结构化的数据。

3、Crawley 爬虫框架

Crawley 也是 Python 开发出的爬虫框架，该框架致力于改变人们从互联网中提取数据的方式。

4、PySpider 爬虫框架

相对于 Scrapy 框架而言，PySpider 框架还是新秀。PySpider 框架采用 Python 语言编写，分布式架构，支持多种数据库后端，强大的 WebUI 支持脚本编辑器，任务监视器，项目管理器以及结果查看器。

英语词汇

背记有法，让英语不再成为编程学习的拦路虎！

requests
含义：请求
热度：★★

error
词义：误差；错误；过失
热度：★★

parse
词义：解析，理解
热度：★★

post
词义：岗位；邮件；标杆；张贴；公布；邮递；布置；快速行进
热度：★★

params
含义：参数；参数个数
热度：★★

header
词义：头部
热度：★★

timeout
含义：超时
热度：★★

Soup
含义：加速；增加马力
热度：★★

widget
含义：装饰物；小机械；小部件；未定名的主要新产品
热度：★★

checkbox
含义：复选框
热度：★★

第 15 章 Web 编程

必背必记

1、HTTP

HTTP（HyperText Transfer Protocol），即超文本传输协议，是互联网上应用最为广泛的一种网络协议。HTTP 是利用 TCP 在 Web 服务器和客户端之间传输信息的协议。

2、HTML 简介

HTML 是用来描述网页的一种语言。HTML 指的是超文本标记语言(Hyper Text Markup Language)，它不是一种编程语言，而是一种标记语言。

3、CSS 简介

CSS 是 Cascading Style Sheets（层叠样式表）的缩写。CSS 是一种标记语言，用于为 HTML 文档定义布局。

4、JavaScript 简介

JavaScript 是一种可以嵌入在 HTML 代码中由客户端浏览器运行的脚本语言。在网页中使用 JavaScript 代码，不仅可以实现网页特效，还可以响应用户请求实现动态交互的功能。

5、CGI 简介

CGI（Common Gateway Interface），即通用网关接口，它是一段程序，运行在服务器上。

6、WSGI 简介

WSGI（Web Server Gateway Interface），即服务器网关接口，是 Web 服务器和 Web 应用程序或框架之间的一种简单而通用的接口。

英语词汇

背记有法，让英语不再成为编程学习的拦路虎！

general
含义：一般的，普通的；综合的；大体的；一般；常规
热度：★★

request
词义：请求；需要；要求，请求
热度：★★

method
词义：方法；条理；类函数；使用体验派表演方法的
热度：★★

code
词义：代码，密码；编码；法典
热度：★★

address
词义：地址；写姓名地址
热度：★★

Web
含义：网
热度：★★

style
词义：类型
热度：★★

link
含义：联系
热度：★★

第 16 章 常用 Web 框架

必背必记

1、常用的 HTTP 方法

方 法 名	说 明
GET	浏览器通知服务器：将获取页面上的信息并发给浏览器
HEAD	浏览器通知服务器：获取信息，但是只关心消息头。应用应像处理GET请求一样来处理它，但是不分发实际内容。在Flask中你完全不需要人工干预，底层的Werkzeug库已经替你处理好了
POST	浏览器通知服务器：在URL上发布新信息。并且，服务器必须确保数据已存储且仅存储一次。这是 HTML表单通常发送数据到服务器的方法
PUT	类似POST，但是服务器可能触发了存储过程多次，多次覆盖掉旧值。考虑到传输中连接可能会丢失，在这种情况下浏览器和服务器之间的系统可能安全地第二次接收请求，而不破坏其他东西。因为POST只触发一次，所以使用POST是不可能的
DELETE	删除给定位置的信息
OPTIONS	给客户端提供一个捷径来弄清这个 URL支持哪些HTTP方法。从Flask 0.6开始，实现了自动处理

2、Jinja2 提供的部分常用过滤器

名 称	说 明
safe	渲染值时不转义
capitalize	把值的首字母转换成大写，其他字母转换成小写
lower	把值转换成小写形式
upper	把值转换成大写形式
title	把值中每个单词的首字母都转换成大写
trim	把值的首尾空格去掉
striptags	渲染之前把值中所有的 HTML 标签都删掉

3、Django 项目中的文件及说明

文 件	说 明
manage.py	Django 程序执行的入口

db.sqlite3	SQLite 的数据库文件，Django 默认使用这种小型数据库存取数据，非必须
templates	Django 生成的 HTML 模板文件夹，我们也可以在每个 app 中使用模板文件夹
demo	Django 生成的和项目同名的配置文件夹
settings.py	Django 总的配置文件，可以配置 App、数据库、中间件、模板等诸多选项
urls.py	Django 默认的路由配置文件
wsgi.py	Django 实现的 WSGI 接口的文件，用来处理 Web 请求

4、Django 项目中 App 目录的文件及说明

文　件	说　明
migrations	执行数据库迁移生成的脚本
admin.py	配置 Django 管理后台的文件
apps.py	单独配置添加的每个 App 的文件
models.py	创建数据库数据模型对象的文件
tests.py	用来编写测试脚本的文件
views.py	用来编写视图控制器的文件

5、Django 数据模型中常见字段类型及说明

字 段 类 型	说　明
AutoField	一个 id 自增的字段，但创建表过程 Django 会自动添加一个自增的主键字段
BinaryField	一个保存二进制源数据的字段
BooleanField	一个布尔值的字段，应该指明默认值，管理后台中默认呈现为 CheckBox 形式
NullBooleanField	可以为 None 值的布尔值字段
CharField	字符串值字段，必须指明参数 max_length 值，管理后台中默认呈现为 TextInput 形式
TextField	文本域字段，对于大量文本应该使用 TextField。管理后台中默认呈现为 TextArea 形式
DateField	日期字段，代表 Python 中 datetime.date 的实例。管理后台默认呈现 TextInput 形式
DateTimeField	时间字段，代表 Python 中 datetime.datetime 实例。管理后台默认呈现 TextInput
EmailField	邮件字段，是 CharField 的实现，用于检查该字段值是否符合邮件地址格式
FileField	上传文件字段，管理后台默认呈现 ClearableFileInput 形式
ImageField	图片上传字段，是 FileField 的实现。管理后台默认呈现 ClearableFileInput 形式
IntegerField	整数值字段，在管理后台默认呈现 NumberInput 或者 TextInput 形式
FloatField	浮点数值字段，在管理后台默认呈现 NumberInput 或者 TextInput 形式
SlugField	只保存字母数字和下划线和连接符，用于生成 url 的短标签
UUIDField	保存一般统一标识符的字段，代表 Python 中 UUID 的实例，建议提供默认值 default
ForeignKey	外键关系字段，需提供外检的模型参数，和on_delete参数（指定当该模型实例删除的

	时候，是否删除关联模型），如果要外键的模型出现在当前模型的后面，需要在第一个参数中使用单引号

英语词汇

背记有法，让英语不再成为编程学习的拦路虎！

return
含义：回转，返回
热度：★★★

content
含义：内容；满足；目录；容量
热度：★

safe
含义：安全的;保险的
热度：★

include
含义：包括；包含；计入
热度：★★

models
含义：模式；模型(model 的名词复数)
热度：★

install
含义：安装；安顿，安置
热度：★