用 fun（）函数后，函数内输出的值是 100。在函数外，局部变量失效，此时输出函数体外赋予的值 200。

6.7　Python 的内置函数

在 Python 中，系统提供了多种内置函数，也称为内建函数，这些内置函数在使用时不需要引用，可以直接调用。常用的内置函数主要包括数学运算函数、类型转换函数、字符串处理函数及其他函数，如表 6-1 所示。

表 6-1　Python 常用内置函数汇总

函数名	功能	函数名	功能
abs（）	求绝对值	all（）	判断参数中的所有数据是否都为 True
any（）	判断参数中是否存在任意一个为 True 的数据	complex（）	创建一个复数
pow（）	幂函数	delattr（）	删除对象的属性
bool（）	将参数转换成逻辑型数据	dir（）	没有参数时，返回当前范围内的变量、方法和定义的类型列表；带参数时，返回参数的属性和方法列表
chr（）	返回对应 ASCII 码的字符	enumerate（）	返回一个可以枚举的对象
bin（）	将十进制数转换成二进制数	float（）	将参数转换为浮点数
bytes（）	将参数转换成字节型数据	frozenset（）	创建一个不可修改的集合
hasattr（）	判断对象是否具备特定的属性	dict（）	创建一个空的字典类型的数据
id（）	返回对象的内存地址	divmod（）	分别求商和余数
int（）	将参数转换成整数	eval（）	计算字符串参数中表达式的值
issubclass（）	检查一个类是否是另一个类的子类	format（）	格式化输出字符串
list（）	构造列表数据	getattr（）	获取对象的属性
max（）	求最大值	hex（）	返回参数的十六进制
next（）	返回一个可迭代数据结构中的下一项	input（）	获取用户输入的内容
open（）	打开文件	isinstance（）	检查对象是否是类的实例
range（）	根据需要生成一个指定的范围	len（）	返回对象长度

（续表）

函数名	功能	函数名	功能
round（）	对参数进行四舍五入	map（）	将参数中的所有数据用指定的函数遍历
setattr（）	设置对象的属性	min（）	返回给定元素中的最小值
str（）	构造字符串类型的数据	oct（）	将参数转换成八进制
super（）	调用父类的方法	ord（）	求参数字符的 ASCII 码
type（）	显示对象所属的类型	print（）	输出函数
reversed（）	反转，逆序对象	sorted（）	对参数进行排序
set（）	创建一个集合类型的数据	sum（）	求和函数
tuple（）	构造元组类型的数据	zip（）	将两个可迭代对象中的数据逐一配对

下面列举一些内置函数示例。

1. 与数学运算相关的内置函数

（1）abs（）函数

求取绝对值。函数格式如下：

```
abs(x)
```

abs（）函数的参数 x 可以是一个整数、长整数或浮点数。如果参数是复数，则返回它的模。示例：

```
>>> abs(-5.2)
5.2
>>> abs(3+4j)
5.0
```

（2）max（）函数

求取最大值。函数格式如下：

```
max(x,y,z,…)
```

max（）函数的参数可以是多个值，返回其中最大的数值。示例：

```
>>> max(-1,2,4,3)
4
```

(3) min () 函数

求取最小值。函数格式如下：

```
min(x,y,z,…)
```

min () 函数的参数可以是多个值，返回其中最小的数值。示例：

```
>>> min(- 1,2,4,3)
- 1
```

(4) pow () 函数

获取乘方数。函数格式如下：

```
pow(x,y)
```

这个函数返回 x 的 y 次幂。示例：

```
>>> pow(3,4)
81
```

(5) round () 函数

获取指定位数的小数。函数格式如下：

```
round (x,y)
```

其中 x 代表浮点数，y 代表要保留的位数。示例：

```
>>> round(2.6235,2)
2.62
```

(6) divmod () 函数

获取商和余数。函数格式如下：

```
divmod(x,y)
```

示例：

```
>>> divmod(9,2)
(4,1)
```

2. 类型转换相关的内置函数

类型转换函数是将一种类型的变量强行转化为另一种类型的变量，常见转换函数如下。

（1）int（）函数

将参数转换为 int 型。函数格式如下：

```
int(str)
```

示例：

```
> > > int(12.1)
12
```

（2）float（）函数

将 int 型或字符型转换为浮点型。函数格式如下：

```
float(int/str)
```

示例：

```
> > > float(3)
3.0
```

（3）str（）函数

转换为字符型。函数格式如下：

```
str(int)
```

示例：

```
> > > str(10)
'10'
```

（4）bool（）函数

转换为布尔类型，0 转化为 False，非 0 转化为 True，函数格式如下：

```
bool(int)
```

示例：

```
>>> bool(0)
False
>>> bool(1)
True
```

(5) hex () 函数

转换为十六进制。函数格式如下：

```
hex(int)
```

示例：

```
>>> hex(10)
'0xa'
```

(6) bin () 函数

转换为二进制。函数格式如下：

```
bin(int)
```

示例：

```
>>> bin(59)
'0b111011'
```

(7) chr () 函数

转换数字为相应的 ASCII 码字符。函数格式如下：

```
chr(int)
```

示例：

```
>>> chr(65)
'A'
```

(8) ord () 函数

转换 ASCII 码字符为相应的数字。函数格式如下：

```
ord(str)
```

示例：

```
>>> ord('Z')
90
```

3. 字符串中字符大小写变换函数（见表 6-2）

表 6-2　Python 字符串中字符大小写变换函数

函数	用法	功能
lower ()	str. lower ()	将字符串 str 中的所有字母转换为小写字母
upper ()	str. upper ()	将字符串 str 中的所有字母转换为大写字母
swapcase ()	str. swapcase ()	将字符串 str 中的所有字母转换为大小写字母互换
capitalize ()	str. capitalize ()	将字符串 str 中的首字母大写，其余字母小写
title ()	str. title ()	将字符串 str 中的每个单词的首字母大写，其余字母小写

【例 6-11】字符串大小写字母转换

程序代码：

```
str= "I am a girl"
print(str.lower())
print(str.upper())
print(str.swapcase())
print(str.capitalize())
print(str.title())
```

运行结果：

```
i am a girl
I AM A GIRL
i AM A GIRL
I am a girl
I Am A Girl
```

4. 指定字符串输出方式相关函数（见表 6-3）

表 6-3　字符串输出方式函数

函数	语法	功能
ljust ()	str. ljust (width, [fillchar])	将 str 左对齐输出，字符串的总宽度为 width，不足的部分以 fillchar 指定的字符串填充，默认使用空格填充

（续表）

函数	语法	功能
rjust（）	str. rjust（width，[fillchar]）	将 str 右对齐输出，字符串的总宽度为 width，不足的部分以 fillchar 指定的字符串填充，默认使用空格填充
center（）	str. center（width，[fillchar]）	将 str 居中对齐输出，字符串的总宽度为 width，不足的部分以 fillchar 指定的字符串填充，默认使用空格填充
zfill（）	str. zfill（width）	将字符串的宽度填充为 width，并且右对齐，不足的部分用 0 补足

【例 6-12】指定字符串输出方式

程序代码：

```
str= "very good!"
print(str.ljust(16,"- "))
print(str.rjust(16,"- "))
print(str.center(16,"- "))
print(str.capitalize())
print(str.zfill(16))
```

运行结果：

```
verygood! - - - - - -
- - - - - - very good!
- - - verygood! - - -
very good!
000000very good!
```

5. 搜索和替换字符串处理函数（见表 6-4）

表 6-4　Python 搜索和替换字符串处理函数

函数	语法	功能
find（）	str. find（substr [，start，[end]]）	从 str 字符串的 start 至 end 的范围内检索是否存在 substr，如果存在，则返回出现子串 substr 的第一个字母的位置，如果 str 中没有 substr，则返回－1
index（）	str. index（substr [，start，[end]]）	与 find（）函数相同，只是在 str 中没有 substr 时，返回一个运行时的错误

（续表）

函数	语法	功能
rfind ()	str. rfind (substr [, start, [end]])	从 str 字符串右侧起来的 start 至 end 的范围内检索是否存在 substr，如果存在，则返回出现子串 substr 的第一个字母的位置，如果 str 中没有 substr，则返回 -1
rindex ()	str. rindex (substr [, start, [end]])	与 rfind () 函数相同，如果没有 substr 时，返回一个运行时的错误
replace ()	str. replace (oldstr, newstr [, count])	把 str 中的 oldstr 替换为 newstr，count 为替换次数

【例 6-13】搜索和替换字符串处理函数示例

```
str= "very good!"
print(str.find('g'))
print(str.index('o'))
print(str.rfind('g'))
print(str.rindex('o'))
print(str.replace("","- "))
```

运行结果：

```
5
6
5
7
- v- e- r- y-  - g- o- o- d- ! -
```


6. 字符串的分割与组合函数（见表 6-5）

表 6-5　分割与组合函数

函数	语法	功能
split ()	str. split ([sep, [maxsplit]])	以 sep 为分隔符，把 str 分割成一个列表，其中 maxsplit 表示分割的次数
splitines ()	str. splitines (keepends)	把 str 按照行分割符分为一个列表，其中参数 keepends 为 bool 值，当取 true 时，每行后面会保留行分隔符
join ()	str. join (seq)	把 seq 代表的字符串序列，用 str 连接起来

【例 6-14】字符串的分割与组合函数示例

程序代码：

```
str = "very good!"
list = str.split(' ')
print(list)
str1 = "- "
print(str1.join(list))
```

运行结果：

```
['very','good! ']
very- good!
```

7. 相关操作函数

(1) type () 函数

功能：返回一个对象的类型。type () 函数语法：

```
type(x)
```

示例：

```
>>> x= 2
>>> type(x)
<class'int'>
>>> y= 'good'
>>> type(y)
<class'str'>
```

(2) help () 函数

功能：调用系统内置的帮助。help () 函数语法：

```
help(para)
```

示例 1：

```
>>> help('input')
Help on built- in function input in modulebuiltins:
input(prompt= None,/)
    Read a string from standard input. The trailing newline is stripped.
```

```
        The prompt string, if given, is printed to standard output without a
        trailing newline before reading input.
        If the user hits EOF (* nix: Ctrl- D, Windows: Ctrl- Z+ Return), raise EOFEr-
        ror.
    On * nix systems, readline is used if available.
```

示例 2：

```
> > > help('print')
Help on built- in function print in modulebuiltins:
print(...)
print(value,...,sep= ' ',end= '\n',file= sys. stdout,flush= False)
    Prints the values to a stream, or tosys. stdout by default.
    Optional keyword arguments:
    file: a file- like object (stream); defaults to the currentsys. stdout.
    sep:string inserted between values, default a space.
    end:string appended after the last value, default a newline.
    flush:whether to forcibly flush the stream.
```

6.8 实验 1：使用函数计算规则图形的面积

```
# 圆形
```

6.8 实验1：使用函数计算规则图形的面积

#圆形

```
def findArea(r):
    s= 3.14* r* r
    print ("圆的面积：",s)          # 输出局部变量 s 的值
    return s
findArea(2)                          # 调用 findArea 函数
```

#长方形

```
def findArea(w,h):
    s= w* h
    print ("长方形的面积：",s)
    return s
findArea(2,3)
```

6.9 实验2：利用递归函数调用方式，将所输入的5个字符，以相反顺序打印出来

```
def func(s):
    try:
        print(s[- 1])
        func(s[0:len(s)- 1])
    except Exception:
        pass
func('12345')
```

运行结果：

```
5
4
3
2
1
```