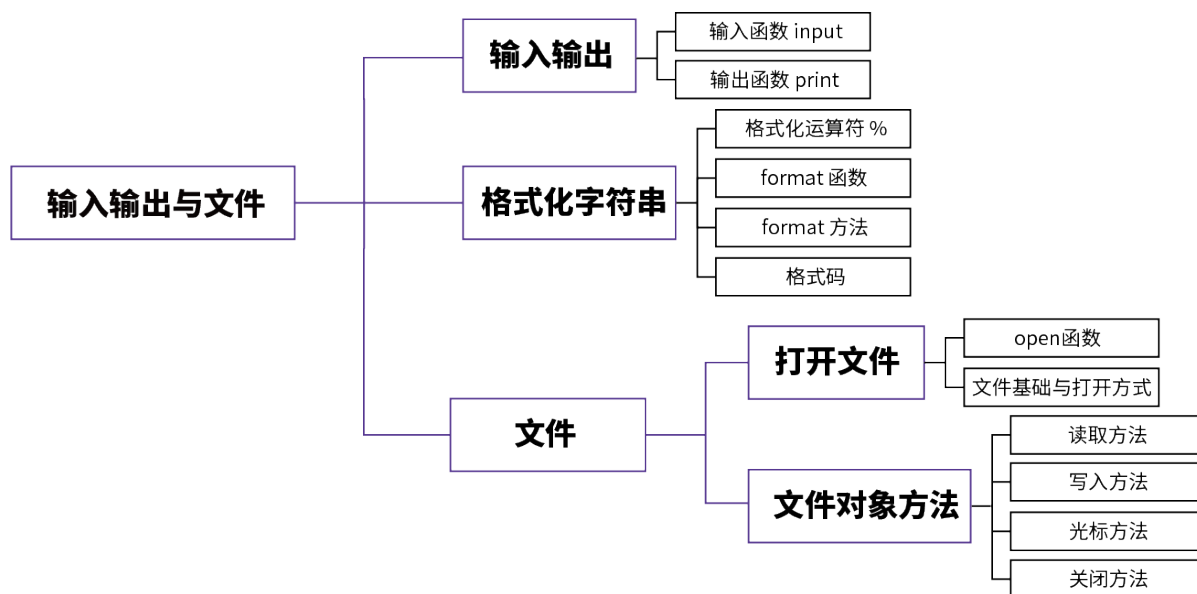


第 5 讲 输入、输出与文件

知识脉络



一 输入与输出函数

1. 输入函数

```
input(string)    # 从键盘（控制台）输入一个字符串，并返回这个字符串
```

· string: 输出提示字符串，执行该函数时会在控制台中输出该字符串提示用户进行操作

① 即使输入的是数字，返回的也是字符串形式的数字，要用 `int()` 转换

② 如果要一次性输入多个值：用空格或指定字符分开，返回的字符串调用 `split()` 方法

2. 输出函数

```
print(*args, end='\n', sep=' ')
```

· 向控制台输出调用时输入的所有参数 `args`，参数间用字符 `sep` 分开，以字符 `end` 结尾

· 要输出的参数会先自动转换成字符串（类似 `str()` 函数），再输出

· `split` 的默认值为空格，`end` 的默认值为回车，这两个参数必须以关键字参数指定

二 格式化字符串

1. 格式化运算符 %

包含占位符`%格式码`的字符串 % 参数序列

```
>>> 'Happy Birthday %d, %s!' % (23, 'John Johnson')
'Happy Birthday 23, John Johnson!'
```

· 将参数序列中的元素按顺序放入字符串中占位符`%`的位置，并按照格式码转换为指定的格式

2. format 函数

```
format(a, 格式码)    # 将参数 a 按照格式码转换为对应格式的字符串
```

```
format(53.2345, "10.2e")    # 5.32e+01
```

3. format 方法

包含占位符{索引或关键字: 格式码}的字符串.format(参数表)

① 参数表中参数为位置参数

- 占位符内可以填入索引，此时参数将按索引填入占位符的位置（默认是从左到右依次填入）

```
>>> 'my name is {1}, age {0}'.format('Mary', 18)
'my name is 18, age Mary'
```

② 参数表中参数为关键字参数

- 占位符内需要填入关键字，此时参数将按关键字填入对应占位符的位置

```
>>> 'my name is {name}, age {age}'.format(name='Mary', age=18)
'my name is Mary, age 18'
```

4. 格式码

<填充字符><对齐方式><宽度><.精度><格式>

① 宽度、填充字符、对齐方式

将参数转换为字符串后，若字符串长度低于宽度，则用填充字符填补至长度等于宽度

填充字符的位置决定于对齐方式：

对齐方式	结合性
>	右对齐
<	左对齐
^	居中对齐

```
>>> '{0:*>10}'.format(10)
'*****10'
```

填充字符默认值为空格，对齐方式默认为右对齐，宽度默认不大于字符串长度

② 格式、精度

格式决定参数将以哪种形式转换为字符串

格式	说明	格式	说明
d	整数（十进制）	X	整数（十六进制，大写字母）
b	整数（二进制）	f	浮点数
o	整数（八进制）	%e	浮点数（科学计数法）
x	整数（十六进制，小写字母）	s	字符串

当格式为浮点数时可以设置精度，即保留几位小数

三 文件

- 文件是存储在外存上的 01 序列数据，分为**二进制文件**和**文本文件**（序列可通过文字编码直接对应）
图形文件、计算机程序等属于二进制文件，文本文件可以用文字处理程序阅读

1. 打开文件

```
open(filename, mode, encoding) # 以指定方式打开指定文件，并返回一个文件对象
```

- **filename**: 文件路径（如果只包含了文件名，那么路径默认为与程序文件同一目录）
- **encoding**: 以何种编码方式读写文件（默认 UTF-8）
- **mode**: 打开模式，由两个字母组成。第一个字母代表操作类型：

	模式名称	若文件不存在	若文件已存在
"r"	只读模式(默认)	报错	不能写
"w"	覆盖写模式	新创建	覆盖掉已有内容重新写
"a"	追加模式	新创建	在已有内容后追加内容
"x"	创建写模式	新创建	报错

第二个字母代表文件类型: "t" / 省略 - 文本类型 "b" - 二进制类型

在此基础上可以追加字符"+", 表示文件可以读也可以写

3. 文件对象的方法

第一组：读取

- ① `file.read(size)` # 从文件 `file` 读取的长度为 `size` 的字符串，如果 `size` 省略或为负则读取所有内容
- ② `file.readline()` # 从文件 `file` 读取的整行字符串（读到换行符为止）
- ③ `file.readlines()` # 从文件 `file` 读取所有字符串，以换行符分割成字符串组成的列表

第二组：写入

- ④ `file.write(s)` # 把字符串 `s` 写入文件
- ⑤ `file.writelines(s)` # 把字符串组成的列表 `s` 写入文件，但换行需要字符串中有换行符

第三组：光标

- ⑥ `file.tell()` # 返回当前文件读写的位置
- ⑦ `file.seek(off, whence=0)` # 设置当前文件读写的位置

- 也就是将光标移动至 `whence` 所在的位置向后 `off` 个字节
- `off` 偏移量，光标向后移动的字节数，如果是负数表示向前移动
- `whence` 计算偏移量的起始位置，0: 文件开头，1: 当前位置，2: 文件末尾

第四组：关闭

- ⑧ `file.close()` # 关闭文件，此后文件不能再进行读写操作