

## 6.文件读写

Python内置了读写文件的函数：`open`，返回文件对象，

通常的用法需要三个参数：`open(filename, mode, encoding)`。

**-filename:** 包含了你要访问的文件名称的字符串值。

**-mode:** 决定了打开文件的模式(`r`: 只读、`w`: 写入、`a`: 追加; `*b`: 二进制的形式操作)。

**-encoding:** 打开文件的编码格式，默认为`utf8`。

示例：`f = open("file_name","r",encoding="utf8")`以只读的方式打开文件`file_name`，编码格式为`utf8`。

In [1]:

```
f = open("text.txt", 'w') # 打开文件text.txt，当文件不存在时会新建一个。
Str = input("请输入要写入的内容：")
f.write(Str)
f.close()
```

In [2]:

```
f = open("text.txt", 'r')
print(f.read(6)) # 读取六个字符，当前光标后移六个字符
print(f.read()) # 读取光标所在位置至最后
f.close()
```

python

In [4]:

```
# 使用with语句进行文件写入
with open("text1.txt", 'w') as f:
    f.write("python文件操作")
# 使用with语句读取文件内容
with open("text1.txt", 'r') as f:
    print(f.read())
```

python文件操作

## 7.异常捕获

程序出现问题无法执行时大多分为两种情况：

**-编码错误:** 代码编写时（运行前）出现问题。

**-异常:** 在运行时出现错误（比如被除数为零、列表的下标越界、修改字符串等）。

In [5]:

```
print(1/0)
```

```
-----
ZeroDivisionError                                Traceback (most recent call last)
<ipython-input-5-2fc232d1511a> in <module>
----> 1 print(1/0)
```

**ZeroDivisionError:** division by zero

In [6]:

```
# 使用try except 捕获异常
try:
    print(1/0)
except Exception as e:
    print(e)
```

division by zero

## 8 常用工具库

Python中的工具库可以分为两种

-标准库：安装Python以后可以直接使用的。如：`os`、`sys`、`time`等模块。

-第三方库：需要安装以后才可以使用的库。如`numpy`（用于科学计算）、`pandas`（用于数据处理）、`scikit-learn`（机器学习库）

### sys模块的使用

`sys`模块负责程序与Python解释器的交互，提供了一系列的函数和变量，用于操控Python运行时的环境。

In [7]:

```
# sys.exit([n]): 此方法可以是当前程序退出，n为0时表示正常退出，其他值表示异常退出。
import sys
for i in range(100):
    print(i)
    if i ==5:
        sys.exit(0)
```

0  
1  
2  
3  
4  
5

An exception has occurred, use %tb to see the full traceback.

**SystemExit:** 0

```
d:\python\python3.7\lib\site-packages\IPython\core\interactiveshell.py:3327: UserWarning: To exit: use
'exit', 'quit', or Ctrl-D.
  warn("To exit: use 'exit', 'quit', or Ctrl-D.", stacklevel=1)
```

In [8]:

```
# sys.path: 获取模块搜索路径。
sys.path
```

Out[8]:

```
['D:\\python project\\HCIA-AI 3.0 Python前置',
 'd:\\python\\python3.7\\python37.zip',
 'd:\\python\\python3.7\\DLLs',
 'd:\\python\\python3.7\\lib',
 'd:\\python\\python3.7',
 '',
 ...]
```

```
'd:\\python\\python3.7\\lib\\site-packages',
'd:\\python\\python3.7\\lib\\site-packages\\win32',
'd:\\python\\python3.7\\lib\\site-packages\\win32\\lib',
'd:\\python\\python3.7\\lib\\site-packages\\Pythonwin',
'd:\\python\\python3.7\\lib\\site-packages\\IPython\\extensions',
'C:\\Users\\zwx631096\\.ipython']
```

In [9]:

```
# sys.platform: 获取当前系统平台。
sys.platform
```

Out[9]:

```
'win32'
```

## os 模块的使用

os模块负责程序与操作系统的交互，提供了访问操作系统底层的接口。

In [10]:

```
import os
# os.getpid() 获取当前进程id
print("当前进程的ID: ", os.getpid())

# os.getppid():获取当前父进程id
print("当前父进程的ID: ", os.getppid())

# os.getcwd(): 获取当前所在路径
cwd = os.getcwd()
print("当前所在路径为: ", cwd)

# os.chdir(path): 改变当前工作目录
os.chdir("C:\\")
print("修改后当前所在路径为: ", os.getcwd())

# os.listdir(): 返回目录下所有文件
print("当前目录下的文件有: ", os.listdir(cwd))

# os.walk(): 输出当前路径下的所有文件
for root, dirs, files in os.walk(cwd, topdown=False):
    for name in files:
        print(os.path.join(root, name))
    for name in dirs:
        print(os.path.join(root, name))
```

当前进程的ID: 4256

当前父进程的ID: 19372

当前所在路径为: D:\python project\HCIA-AI 3.0 Python前置

修改后当前所在路径为: C:\

当前目录下的文件有: ['.ipynb\_checkpoints', 'Python基础语法.ipynb', 'Python数据结构.ipynb', 'text.txt', 'text1.txt', '函数与对象.ipynb', '控制流语句.ipynb', '文件读写与异常捕获.ipynb']

D:\python project\HCIA-AI 3.0 Python前置\ipynb\_checkpoints\Python基础语法-checkpoint.ipynb

D:\python project\HCIA-AI 3.0 Python前置\ipynb\_checkpoints\Python数据结构-checkpoint.ipynb

D:\python project\HCIA-AI 3.0 Python前置\ipynb\_checkpoints\函数与对象-checkpoint.ipynb

D:\python project\HCIA-AI 3.0 Python前置\ipynb\_checkpoints\控制流语句-checkpoint.ipynb

D:\python project\HCIA-AI 3.0 Python前置\ipynb\_checkpoints\文件读写与异常捕获-checkpoint.ipynb

D:\python project\HCIA-AI 3.0 Python前置\Python基础语法.ipynb

D:\python project\HCIA-AI 3.0 Python前置\Python数据结构.ipynb

D:\python project\HCIA-AI 3.0 Python前置\text.txt

D:\python project\HCIA-AI 3.0 Python前置\text1.txt

D:\python project\HCIA-AI 3.0 Python前置\函数与对象.ipynb

D:\python project\HCIA-AI 3.0 Python前置\控制流语句.ipynb

D:\python project\HCIA-AI 3.0 Python前置\文件读写与异常捕获.ipynb

D:\python project\HCIA-AI 3.0 Python前置\ipynb\_checkpoints

In [12]:

```
# os.path模块: 主要用于获取文件的属性。
```

```

import os
# os.path.abspath(path): 返回绝对路径
print("text.txt的绝对路径为: ", os.path.abspath("text.txt")) # text.txt 为当前文件夹下的一个文件（上一个实验中
将当前路径切换为C:\,需要将路径切换回来）

# os.path.exists(path): 文件存在则返回True,不存在返回False
print("text.txt是否存在: ", os.path.exists("text.txt"))

# os.path.getsize(path): 返回文件大小, 如果文件不存在就返回错误
print("text.txt的文件大小: ", os.path.getsize("text.txt"))

# os.path.isfile(path): 判断路径是否为文件
print("text.txt是否为文件: ", os.path.isfile("text.txt"))

# os.path.isdir(path): 判断路径是否为文件夹
print("text.txt是否为文件夹: ", os.path.isdir("text.txt"))

```

text.txt的绝对路径为: C:\text.txt  
text.txt是否存在: False

```

-----
FileNotFoundError                                Traceback (most recent call last)
<ipython-input-12-83c5896a5b1d> in <module>
      8
      9 # os.path.getsize(path): 返回文件大小, 如果文件不存在就返回错误
--> 10 print("text.txt的文件大小: ", os.path.getsize("text.txt"))
     11
     12 # os.path.isfile(path): 判断路径是否为文件

d:\python\python3.7\lib\genericpath.py in getsize(filename)
     48 def getsize(filename):
     49     """Return the size of a file, reported by os.stat()."""
--> 50     return os.stat(filename).st_size
     51
     52

FileNotFoundError: [WinError 2] 系统找不到指定的文件。: 'text.txt'

```

## time 模块的使用

time模块是Python中处理时间的一个重要模块, 包含了许多的有关时间操作的方法。

In [14]:

```

import time
# time.time(): 用于获取当前时间戳
time_now = time.time()
print("时间戳: ", time_now)

# time.localtime(): 获取时间元组
localtime = time.localtime(time_now)
print("本地时间为: ", localtime)

# time.asctime(): 获取格式化的时间
localtime = time.asctime(localtime)
print("本地时间为: ", localtime)

# time.strftime(format[, t]): 接收时间元组, 并返回以可读字符串表示的当地时间, 格式由参数format决定。
print(time.strftime("%Y-%m-%d %H:%M:%S", time.localtime()))

```

时间戳: 1584458174.6068165  
本地时间为 : time.struct\_time(tm\_year=2020, tm\_mon=3, tm\_mday=17, tm\_hour=23, tm\_min=16, tm\_sec=14, tm\_wday=1, tm\_yday=77, tm\_isdst=0)  
本地时间为 : Tue Mar 17 23:16:14 2020  
2020-03-17 23:16:14

In [ ]:

