

# Lesson5 文件操作和异常处理

## 一. 文件基本操作

### --打开文件

使用 `open()` 函数打开文件，可以指定文件路径和打开模式。

'r': 读取模式（默认）

'w': 写入模式（会覆盖文件内容）

'a': 追加模式（在文件末尾添加内容）

'b': 二进制模式（与其他模式结合使用，如 'rb', 'wb'）

```
In [5]: # 打开文件
file = open('example.txt', 'r')

file.close()
```

### --读取文件

使用 `read()`, `readline()` 或 `readlines()` 方法读取文件内容。

```
In [1]: # 打开文件并读取内容
file = open('example.txt', 'r')

# 读取整个文件内容
content = file.read()
print(content)

# 关闭文件
file.close()
```

```
Hello, world!
This is a test file.
```

### --写入文件

使用 `write()` 方法向文件写入内容。

```
In [7]: # 打开文件并写入内容
file = open('example.txt', 'w')

# 写入内容
file.write("Hello, world!\n")
file.write("This is a test file.")

# 关闭文件
file.close()
```

使用 'a' 模式打开文件，并使用 `write()` 方法追加内容。

```
In [ ]: # 打开文件并追加内容
file = open('example.txt', 'a')
```

```
# 追加内容
file.write("\nAppending a new line.")

# 关闭文件
file.close()
```

with 语句用于简化资源管理，确保资源在使用后能够自动关闭或释放。

```
In [2]: # 创建并写入文件
with open('students.txt', 'w') as file:
    file.write("Name,Age,Grade\n")
    file.write("Alice,20,A\n")
    file.write("Bob,22,B\n")

# 读取文件内容
with open('students.txt', 'r') as file:
    content = file.read()
    print("File content:\n", content)

# 追加文件内容
with open('students.txt', 'a') as file:
    file.write("Charlie,23,C\n")

# 逐行读取文件内容
with open('students.txt', 'r') as file:
    print("Reading line by line:")
    for line in file:
        print(line.strip())
```

```
File content:
Name,Age,Grade
Alice,20,A
Bob,22,B
```

```
Reading line by line:
Name,Age,Grade
Alice,20,A
Bob,22,B
Charlie,23,C
```

---

## 二. 异常处理

--语法错误

```
In [4]: while True:
        print('Hello world')
        break
```

Hello world

--异常与异常处理

即便 Python 程序的语法是正确的，在运行它的时候，也有可能发生错误。运行期检测到的错误被称为异常。大多数的异常都不会被程序处理，都以错误信息的形式展现。

异常处理：try/except

```
In [5]: # 异常处理基本格式
# try:
#     可能发生异常的代码
# except ExceptionType:
#     处理异常的代码

# 示例：处理除零异常
try:
    result = 10 / 0
except ZeroDivisionError:
    print("Error: Cannot divide by zero!")
```

Error: Cannot divide by zero!

```
In [ ]: # 示例：处理多种异常
try:
    num = int(input("Enter a number: "))
    result = 10 / num
except ValueError:
    print("Error: Invalid input! Please enter a valid number.")
except ZeroDivisionError:
    print("Error: Cannot divide by zero!")
```

else 块

```
In [ ]: # 示例：使用 else 块，当没有异常发生时执行
try:
    num = int(input("Enter a number: "))
    result = 10 / num
except ValueError:
    print("Error: Invalid input! Please enter a valid number.")
except ZeroDivisionError:
    print("Error: Cannot divide by zero!")
else:
    print(f"Result: {result}")
```

finally 块

```
In [ ]: # 示例：使用 finally 块，无论是否发生异常都会执行
try:
    num = int(input("Enter a number: "))
    result = 10 / num
except ValueError:
    print("Error: Invalid input! Please enter a valid number.")
except ZeroDivisionError:
    print("Error: Cannot divide by zero!")
else:
    print(f"Result: {result}")
finally:
    print("Execution completed.")
```

---

作者: Yming

邮箱: yuemingn@student.unimelb.edu.au

版权声明: 本代码仅用于个人学习目的，未经许可，不得用于商业用途。

