

day18-异常

- 常见异常

```
TypeError 类型异常 +
NameError 名字异常
ValueError 值异常 remove()
KeyError 键异常
AttributeError
IndexError
SyntaxError 语法异常
AssertionError 断言异常
OSError
FileNotFoundError
```

- 异常处理

异常 报错 不一定是坏事
实际开发过程中，尽量保证程序的健壮性，尽可能容错性更多。

- 异常处理方式

```
1. try-except
    try:
        测试代码
```

`except 异常类型 (as 变量名) :`

`异常处理代码块`

- `try`执行 测试代码，如果测试代码没有异常，则执行完毕之后，`try-except`结构结束
- `try`执行 测试代码，一旦测试某一行出问题，则之后测试代码不会执行，直接进入`except`结构进行异常捕获，如果异常类型匹配，则捕获成功，执行 异常处理代码块；否则`except`不能捕获成功，则由python异常机制来处理。
- `except 异常类型 as 变量名`
 - 变量名：表示当前异常类型具体的错误提示参数/提示信息

2. `try-多个except`

`try:`

`测试代码`

`except 异常类型1 as tips1:`

`异常类型1处理`

`except 异常类型2 as tips2:`

`异常类型2处理`

`except 异常类型3 as tips3:`

`异常类型3处理`

`...`

- 尝试执行测试代码，一旦测试代码出现问题，则会到多个`except`结构中进行逐一比对，如果某个`except`比对成功，则进入该 异常类型处理代码块；否则，python内置异常捕获机制启动处理

- 注意：常见异常，实际上是一个一个类实现的。而常见类都是继承自Exception。Exception的父类是BaseException。所以在进行多个except结构书写时，需要注意前后顺序问题，尽量父类(范围广的异常)放在后面，从而避免直接放在前面拦截掉后面的异常类型。

- 特殊写法：

- try:

- 测试代码

- except:

- 异常处理代码块

3. try-except-finally

- try:

- 测试代码

- except 异常类型1 as tips1:

- 异常类型1处理

- ...

- finally:

- finally代码块

- 尝试测试代码有问题，except结构成功捕获异常，仍会执行finally代码块
- EXCEPT结构未能成功捕获，仍会执行
- 测试代码无问题，也会执行
- 不管测试代码有无问题，都会执行finally。

本周作业：

- 当天代码整理
- 复习day1-day18。
- 整理思维导图。 day1-10本周完成前10天。