

[笔记][Learning Python][7. 异常和工具]

Python

[笔记][Learning Python][7. 异常和工具]

33. 异常基础

34. 异常编写细节

34.5 assert 语句

34.6 with/as 上下文管理器

33. 异常基础

34. 异常编写细节

34.5 assert 语句

参考文章：https://blog.csdn.net/you_are_my_dream/article/details/53328293

除了 `assert` 语句之外，python还提供内置的只读变量 `__debug__`，除非解释器运行在最优模式中(通过使用 `-O` 选项指定)，否则它的值为 `True`。

在最优模式中，`if __debug__` 语句以及所有相关语句都将从程序中完全删除。

个人补充：

编写 `test.py` 内容如下：

```
assert False
print('执行完毕啦')
```

在控制台运行如下：

```
PS D:\code\learningpython\C33> python test.py
Traceback (most recent call last):
  File "test.py", line 1, in <module>
    assert False
AssertionError
PS D:\code\learningpython\C33> python -O test.py
执行完毕啦
```

可见如果使用 `-O` 选项开启优化模式，`Python` 解释器根本不会编译 `assert` 语句到字节码。

34.6 with/as 上下文管理器

`PyCharm` 中自动补全的 `__enter__` 方法定义是：

```
def __enter__(self, exc_type, exc_val, exc_tb):
```

以后都这么写吧！