

14.7 捕获所有异常¶

问题¶

怎样捕获代码中的所有异常？

解决方案¶

想要捕获所有的异常，可以直接捕获 `Exception` 即可：

```
try:
    ...
except Exception as e:
    ...
    log('Reason:', e)          # Important!
```

这个将会捕获除了 `SystemExit`、`KeyboardInterrupt` 和 `GeneratorExit` 之外的所有异常。如果你还想捕获这三个异常，将 `Exception` 改成 `BaseException` 即可。

讨论¶

捕获所有异常通常是由于程序员在某些复杂操作中并不能记住所有可能的异常。如果你不是很细心的人，这也是编写不易调试代码的一个简单方法。

正因如此，如果你选择捕获所有异常，那么在某个地方（比如日志文件、打印异常到屏幕）打印确切原因就比较重要了。如果你没有这样做，有时候你看到异常打印时可能摸不着头脑，就像下面这样：

```
def parse_int(s):
    try:
        n = int(v)
    except Exception:
        print("Couldn't parse")
```

试着运行这个函数，结果如下：

```
>>> parse_int('n/a')
Couldn't parse
>>> parse_int('42')
Couldn't parse
>>>
```

这时候你就会挠头想：“这咋回事啊？”假如你像下面这样重写这个函数：

```
def parse_int(s):
    try:
        n = int(v)
    except Exception as e:
        print("Couldn't parse")
        print('Reason:', e)
```

这时候你能获取如下输出，指明了有个编程错误：

```
>>> parse_int('42')
Couldn't parse
Reason: global name 'v' is not defined
>>>
```

很明显，你应该尽可能将异常处理器定义的精准一些。不过，要是你必须捕获所有异常，确保打印正确的诊断信息或将异常传播出去，这样不会丢失掉异常。