

- 前言
 - 开始
 - 1. 准备工作
 - 2. 第一次测试
 - 3. 修改
 - 4. 第二次测试
 - 5. 第三次测试
 - 后记
-

title: unlimited-errors date: 2022-06-23 index_img: <https://dice-forum.s3.ap-northeast-1.amazonaws.com/2022-05-28/1653713607-498189-screenshot-2022-05-28-12-41-00-92.jpg> categories:

- [技术,Lua] tags:
 - Dice
-

{% pdf /pdf/ssJSKFJDJ.pdf %}

{% note secondary %} 目前正在努力ing... 利用 `error()` 与 `pcall()` 进行lua自定义报错
{% endnote %}

前言

使用《署名—非商业性使用—相同方式共享4.0 协议国际版》（CC BY-NC-SA 4.0）进行授权<https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode.zh-Hans> 作者：简律纯 联系方式：qq:A2C29K9 更新日期：2022/5/28

{% note warning %} **WARNING** ⚠ 本帖是作为利用basicFunction自带的chat指令并结合修改chat指令内tryCatch部分的函数名来进行报错排查的示例帖。

简言之让报错更好看一点，更容易理解一点。{% endnote %}

taboo前些日子（好久了）评论了basicFunction的tryCatch函数，他说，

骰娘自检可以安排到行程上了

虽然和写tryCatch的初衷完全不一样，但是返校后在考试过程中突然发现这会是一个很不错的想法，于是单独写一篇帖子介绍它。

开始

1. 准备工作

首先我们在basicFunction的chat指令函数里的tryCatch部分写上这些代码，图片中的抓取报错函数示例是 `read_file()` 函数，输入的参数是 `path`，文件路径；返回值为文件内容。

```
976 elseif(fitems[1]=="tryCatch") then
977     local pc=tryCatch(read_file,fitems[2])
978     if(string.find(pc,"nil")) then
979         return "笨蛋，有输入要read的文件路径吗？"
980     else
981         return pc
982     end
```

这段代码的意思是如果报错信息里有nil这个字符串那返回自定义报错回复内容，否则返回read_file函数的返回内容。

2. 第一次测试

接着输入 `#basicFunction:tryCatch`，不带任何参数，它等效于 `#basicFunction:read_file` 接着的确报错了，但是因为 `read_file()` 函数的 `path` 参数为空，报错信息中有nil字符串，于是被替换成了预先写好的自定义报错回复内容。接着我使用 `#basicFunction:read_file myDiary.txt` 查看了此文件，由于这个文件并不存在，于是krypton返回了read_file里写好的自定义回复。

取消 ssJSKFJDJ(13)





简律纯 12:28



#basicFunction:tryCatch

@简律纯 笨蛋，有输入要read的文件路径吗？

管理员 简律纯



#basicFunction:tryCatch
myDiary.txt



管理员 krypton/[IBU]信使

简律纯 12:28



#basicFunction:tryCatch myDiary.txt

@简律纯 没有错误发生，返回值如下：
没有该文件或文件内容为空哦



3.修改

那么我们使用 `#basicFunction:write_file myDiary.txt [内容]` 来创建这个文件并且写下点什么~烂人诅咒~

取消 ssJSKFJDJ(13)

管理员 简律纯



#basicFunction:read_file
myDiary.txt



管理员 krypton/[IBU]信使

简律纯 12:29



#basicFunction:read_file myDiary.txt

@简律纯 没有该文件或文件内容为空哦

管理员 简律纯



#basicFunction:write_file
myDiary.txt
2022年5月14日
#总有一天，我会成长，而你永远是
是个烂人。



管理员 krypton/[IBU]信使

简律纯 12:29



#basicFunction:write_file myDiary.txt

@简律纯 本大小姐已将文本写入啦



4.第二次测试

已经写好了此文件，我们再用 `#basicFunction:read_file myDiary.txt` 查看一下，来确保有此文件并且写入了内容。

取消 ssJSKFJDJ(13)



5.第三次测试

一切顺利，这时候我们再输入正确的指令来试试抓取报错

取消 ssJSKFJDJ(13)

12:30

管理员 简律纯



#basicFunction:tryCatch
myDiary.txt



管理员 krypton/[IBU]信使

简律纯 12:30



#basicFunction:tryCatch myDiary.txt

@简律纯 没有错误发生，返回值如下：

--myDiary.txt

2022年5月14日

#总有一天，我会成长，而你永远是个烂人。



因为这次输入了参数，即 `#basicFunction:tryCatch myDiary.txt` 等效于 `#basicFunction:read_file myDiary.txt`，所以顺理成章的返回了里面的文本内容。

后记

附上tryCatch函数的内容，如下：

```
lua-----[15]:
-- 功能：监听错误报告如有则反馈
-- 参数：函数名
-- 返回：错误信息
-- 说明：lua有两个函数可以捕获异常，分别是pcall(fun[,arg1,...])和xpcall(fun,errfun,
[,arg1,...])，对比这两个函数,xpcall多了异常处理，这里仅采用pcall()做示例
tryCatch = function(fun,arg1,arg2,arg3,arg4,arg5)
    local ret,errMessage=pcall(fun,arg1,arg2,arg3,arg4,arg5);
    wrong=ret and "false" or "true"
    --print("是否错误:\n"..错误.." \n\n出错信息:\n" .. (errMessage or "无"));
    if wrong=="true" then--错误提示
        local ret,errMessage=pcall(fun,arg1,arg2,arg3,arg4,arg5)
        return "\n错误详情: \n"..errMessage
    else--无错误正常执行
        ret,back=pcall(fun,arg1,arg2,arg3,arg4,arg5);
        return "没有错误发生，返回值如下:\n"..back
    end
end
```