

ZOE's MindMap

17-17 | v3.0 | 20180410

公众号: 数林觅风
<https://woaielf.github.io/>
17 测试 | 调试

IPython

交互式调试器

- 调用「调试器」
 - 引发异常的栈帧
 - 直接跳到
 - %debug
- 直接调用「调试器」
 - 出现异常后
 - %pdb
- 任意 Python 代码
 - 可执行
- 对象 & 数据
 - 各个「栈帧」中
 - 可查看
- 调试器中
 - 命令列表
 - h(elp)
 - 恢复程序执行
 - c(ontinue)
 - 退出调试器
 - q(uit)
 - 设置该行断点
 - *
 - b(reak) *
 - 函数调用
 - 单步进入
 - s(tep)
 - 前进下一行
 - 执行当前行
 - n(ext)
 - 显示当前函数参数
 - a(rgs)
 - u(p)
 - 切换「栈级别」
 - d(own)
- 调试器命令
 - 命令列表
 - h(elp)
 - 恢复程序执行
 - c(ontinue)
 - 退出调试器
 - q(uit)
 - 设置该行断点
 - *
 - b(reak) *
 - 函数调用
 - 单步进入
 - s(tep)
 - 前进下一行
 - 执行当前行
 - n(ext)
 - 显示当前函数参数
 - a(rgs)
 - u(p)
 - 切换「栈级别」
 - d(own)

调试方法

- assert
 - 必须为「真」
 - 判定
 - 则引发异常
 - 若为假
 - 断言
 - 表达式
 - assert expression[, arguments]
 - 使用
 - assert n != 0, 'n is zero!'
 - e.g.
- logging
 - 允许配置「记录信息」级别
 - logging.info()
 - import logging
 - logging.basicConfig(level=logging.INFO)
 - 代码
 - debug/info/warning/error
 - 级别
- pdb
 - 让程序以「单步方式」运行
 - 以参数 -m pdb 启动
 - 单步执行代码
 - n
 - 查看变量
 - P
 - 输入命令
 - 退出程序
 - 结束调试
 - q
 - import pdb
 - 在可能出错的地方
 - 设置一个断点
 - pdb.set_trace()
 - 放一个
 - 查看变量
 - P
 - 继续运行
 - c
 - 命令

测试「函数」

- 测试/验证
 - verification/validation
 - 验证程序的「功能」是否达到要求
- 测试
 - debug
 - 也许功能达到要求
 - 但运行时出现异常
- vs
 - 单元测试
 - 用于核实
 - 函数的某个方面
 - 没有问题
 - 测试用例
 - 一组「单元测试」
 - 一起核实
 - 函数在各种情形下的行为
 - 都符合要求
 - 全覆盖式测试
 - 一整套「单元测试」
 - 涵盖了
 - 各种可能的函数使用方式
- 编写「测试用例」
 - import unittest
 - 导入「待测试函数」
 - 创建继承 unittest.TestCase 的「测试类」
 - 编写一系列方法
 - 测试「函数行为」的不同方面
 - 必须以「test_」打头
 - 将自动运行
 - 断言方法
 - 用来核实
 - 得到的结果
 - 是否与期望的结果一致
 - unittest
 - .assertEqual()
 - 比较是否「相等」
 - 运行测试
 - unittest.main()
- 添加新测试
 - 直接编写 test_ 函数即可
- 「单元测试」结果
 - 通过
 - .
 - 不能通过
 - E
 - 断言失败
 - F

测试「类」

- unittest.TestCase 类
 - 断言方法
 - 6 个核实方法
 - assertEqual(a, b)
 - a == b
 - assertNotEqual(a, b)
 - a != b
 - assertTrue(x)
 - x 为 True/False
 - assertFalse(x)
 - x 为 True/False
 - assertIn(item, list)
 - item 在/不在 list 中
 - assertNotIn(item, list)
 - item 在/不在 list 中
 - 类似于函数的测试
 - def setUp(self):
 - 创建一系列实例 & 设置属性
 - 先运行
 - .
 - 再运行
 - 以 test_ 打头的方法
 - 在测试方法中直接使用实例



公众号: 数林觅风