PyCon China 2020

PyConChina 2020
PYTHON 中国开发者大会 2020


# 好好写代码：遗失的Python编程原则

赖勇浩

广东天勤科技有限公司

# 自我介绍



- 赖勇浩
  - 从业15年，game->web game->web->data
  - 天勤科技：电商易（电商卖家数据工具）
  - 淘播眼（https://taoboyan.com/）：淘宝直播数据分析工具

# 自我介绍



- 赖勇浩
  - 从业15年，game->web game->web->data
  - 天勤科技：电商易（电商卖家数据工具）
  - 淘播眼（https://taoboyan.com/）：淘宝直播数据分析工具

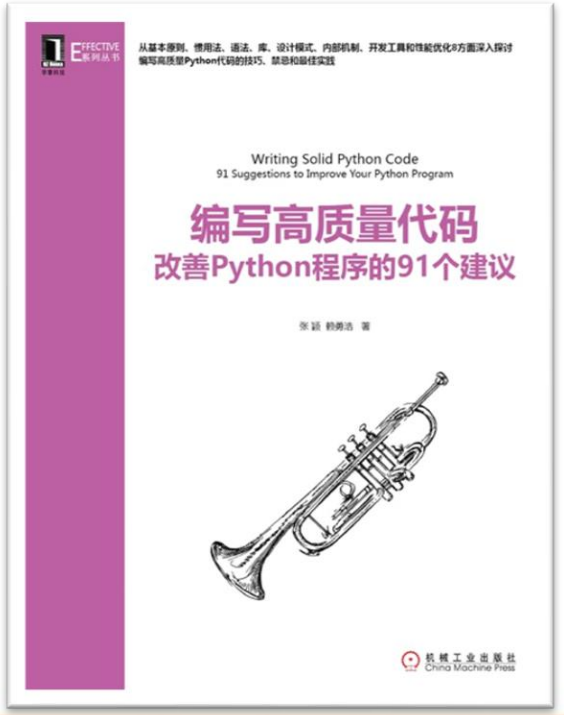
- PyCon China 的老朋友。
  - 2011：Python于web-game的应用
  - 2012：页游开发中的Python组件与模式
  - 2013：论Python与设计模式
  - 2015：Cython的一点使用经验

# 自我介绍



- 赖勇浩
  - 从业15年，game->web game->web->data
  - 天勤科技：电商易（电商卖家数据工具）
  - 淘播眼（https://taoboyan.com/）：淘宝直播数据分析工具

- PyCon China 的老朋友。
  - 2011：Python于web-game的应用
  - 2012：页游开发中的Python组件与模式
  - 2013：论Python与设计模式
  - 2015：Cython的一点使用经验



# 目录



引子 → 工具 → 理论 → 实践 → 小结

# 引子



```
#endpoint list
__endpoints = dict()

#load endpoints info from endpoints.xml file and parse to dict.
__endpoints_file = os.path.join(parent_dir, 'endpoints.xml')
try:
    DOMTree = parse(__endpoints_file)
    root = DOMTree.documentElement
    eps = root.getElementsByTagName('Endpoint')
    for endpoint in eps:
        region_list = []
        product_list = []
        regions = endpoint.getElementsByTagName('RegionId')
        products = endpoint.getElementsByTagName('Product')
        for region in regions:
            region_list.append(region.childNodes[0].nodeValue)
        for product in products:
            name_node = product.getElementsByTagName('ProductName')[0]
            name = name_node.childNodes[0].nodeValue
            domain_node = product.getElementsByTagName('DomainName')[0]
            domain = domain_node.childNodes[0].nodeValue
            product_list.append({name: domain})

        __endpoints[endpoint.getAttribute('name')] = dict(regions=region_list, products=product_list)

except Exception as ex:
    raise ClientException(error_code.SDK_MISSING_ENDPOINTS_FILER, error_msg.get_msg('SDK_MISSING_ENDPOINTS_FILER'))


def find_product_domain(regionid, prod_name):
    """
```

# 引子



- aliyun-python-sdk-core 2.3.1
  - aliyunsdkcore/profile/region_provider.py

# 引子



- aliyun-python-sdk-core 2.3.1
  - aliyunsdkcore/profile/region_provider.py
  - 剖析性能问题，发现xml操作是“耗时大户”，发现了这段代码

# 引子



- aliyun-python-sdk-core 2.3.1
  - aliyunsdkcore/profile/region_provider.py
  - 剖析性能问题，发现xml操作是“耗时大户”，发现了这段代码

```
parent_dir = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
sys.path.insert(0, parent_dir)
```

  - from profile import Profile
    - ImportError: cannot import name 'Profile'

# 引子



- aliyun-python-sdk-core 2.3.1
  - aliyunsdkcore/profile/region_provider.py
  - 剖析性能问题，发现xml操作是“耗时大户”，发现了这段代码

```
parent_dir = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
sys.path.insert(0, parent_dir)
```

  - from profile import Profile
    - ImportError: cannot import name 'Profile'

- 烂代码并不随开发人员年纪、团队规模的增大而减少。

# 工具



工欲善其事，必先利其器。

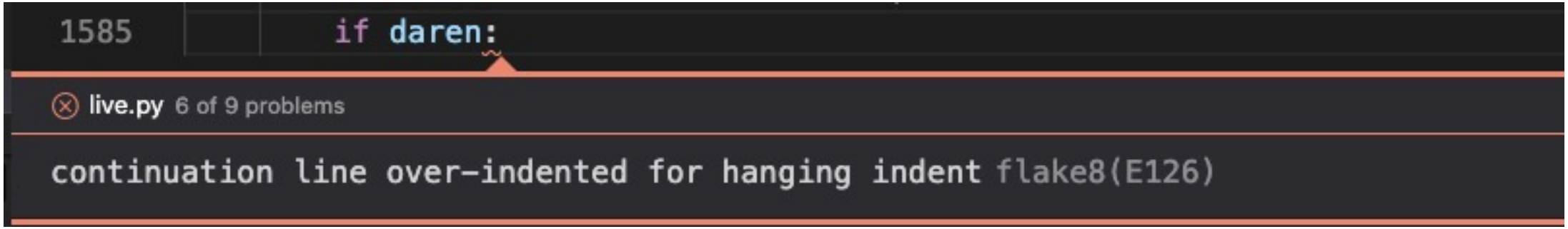
# 工具



pyflakes/flake8/pylance…

# 工具






只要我发现得够早，bug就追不上我。

# 理论



理论指导实践，实践检验理论。

# 理论



- S：单一职责原则 (The Single Responsibility Principle)
  - 每个模块或者类只应该有一项功能。
- O：开闭原则 (The Open/Closed Principle)
  - 实体应对扩展开放并对修改关闭。
- L：里氏替换原则 (The Liskov Substitution Principle)
  - 可以在不破坏系统的情况下，用子类型替换类型。
- I：接口隔离原则 (The Interface Segregation Principle)
  - 不应强制任何客户端依赖于它不使用的方法。
- D：依赖反转原则 (The Dependency Inversion Principle)
  - 高级模块不应该依赖于低级实现。

# 理论



- S：单一职责原则 (The Single Responsibility Principle)
- O：开闭原则 (The Open/Closed Principle)
- L：里氏替换原则 (The Liskov Substitution Principle)
- I：接口隔离原则 (The Interface Segregation Principle)
- D：依赖反转原则 (The Dependency Inversion Principle)

能够帮助开发人员构建更易于维护的系统。

# 理论



- 不要重复你自己原则 (The DRY Principle)
  - 系统中，每一块知识都必须是单一、明确而权威的。
- KISS 原则 (The KISS Principle)
  - 保持简单和直白。
- 你不需要它原则 (YAGNI, You Aren't Gonna Need It)
  - 只有当你需要某些东西的时候，才去实现它们，而不是在你预见的时候。
- Unix 哲学 (The Unix Philosophy)
  - 将小而简单以及定义良好的单元组合在一起，而不是使用大而复杂的多用途程序，可以更轻松地构建系统。
- Python之禅(The Zen of Python)
- ……


我举个栗子

# 实践



淘猫眼 taoboyan.com　首页　直播间旺旺采集　主播实时数据　分析工具　购买套餐　直播讲师招募　视频教程　旗舰版 (2021-09-07到期)　188****6650


**李子柒桂花坚果藕粉藕粉坚果羹营养早餐杭州特产代餐食品方便速食**

天猫店铺：李子柒旗舰店　本店商品投放情况

商品价格：￥59.7

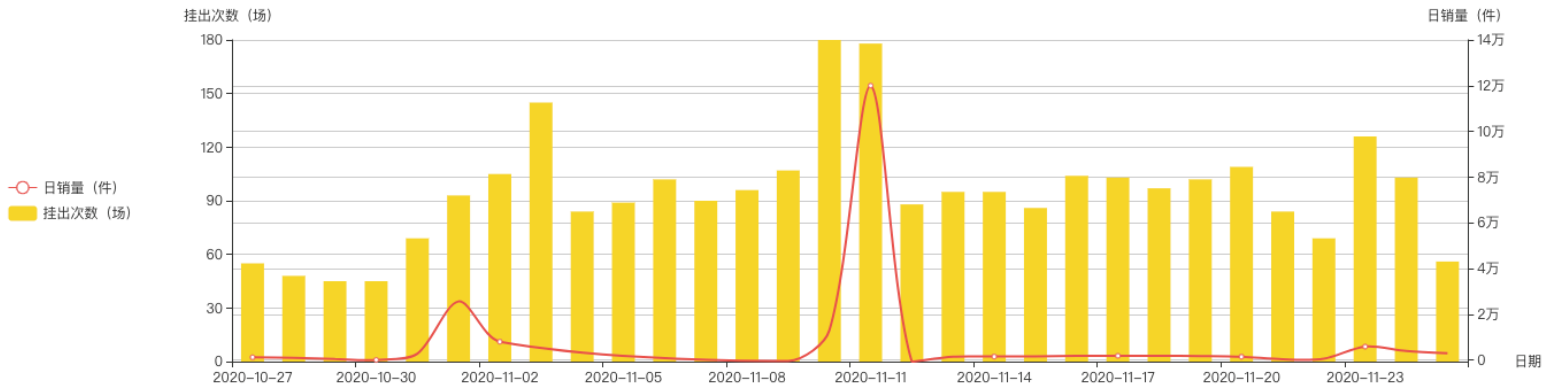
商品品牌：李子柒

商品类目：藕粉　同类商品投放分析

近7天　近15天　近30天

| 销量 | 销售额(估值) | 在售主播数 | 直播挂出次数 |
|---|---|---|---|
| 58.96万 | 3519.86万 | 502 | 178 |
| 近30天 | 近30天 | 近30天 | 近30天 |

直播销量　在售主播　相关回放

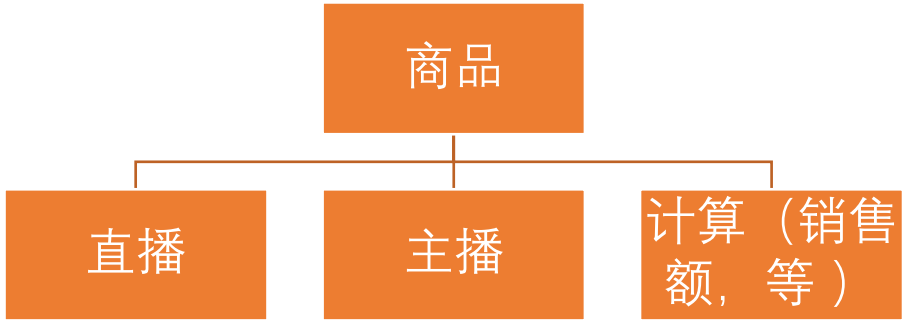



注：1.点击**图中节点**可以查看**当日商品直播详情**。2. 仅2020年7月15日后有日销量数据。3. 日销量数据未严格区分客源渠道，仅供参考。

联系客服

# 实践

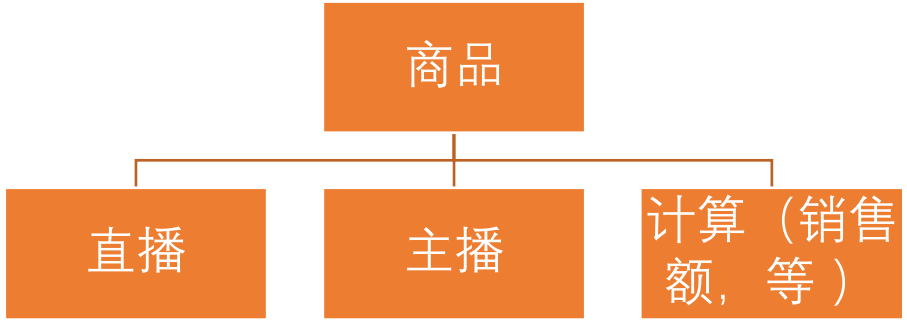



商品
直播
主播
计算（销售额，等）


淘播眼
首页
直播间旺旺采集
主播实时数据
分析工具
购买套餐
直播讲师招募
视频教程
李子柒桂花坚果藕粉藕粉坚果羹营养早餐杭州特产代餐食品方便速食
天猫店铺：李子柒旗舰店
本店商品投放情况
商品价格：¥59.7
商品品牌：李子柒
商品类目：藕粉
同类商品投放分析
近7天
近15天
近30天
销量
58.96万
销售额(估值)
3519.86万
在售主播数
502
直播挂出次数
178
直播销量
在售主播
相关回放
挂出次数（场）
日销量（件）
2020-10-27
2020-10-30
2020-11-02
2020-11-05
2020-11-08
2020-11-11
2020-11-14
2020-11-17
2020-11-20
2020-11-23
日期
联系客服

# 实践







before: 以主播为入口点采集其直播记录，根据直播记录采集商品信息、计算数值。

- after
  - 每个采集进程只做一件事。（单一职责原则）
    - 主播
    - 直播
    - 商品
    - 计算
  - 代码简短清晰
    - 从1000来行减到100多行
  - 性能提升巨大
    - 并发

```
99                  )
100             )
101
102         if fc:
103             darens_coll = self.get_db().get_collectior
104             ret = darens_coll.bulk_write(ops, ordered=
105             if ret.modified_count:
106                 self.log.info(f'updated {ret.modified_
107
108
109 if __name__ == "__main__":
110     mgdfc.run_service_forever()
111
```

再举个栗子

# 引子



```
89          args = {}
90          if shop_result['shop_name'] != '' and shop_result['shop_name'] != doc[Shop['shop_name']]:
91              args[Shop['shop_name']] = shop_result['shop_name']
92          if shop_result['shop_img'] != '' and shop_result['shop_img'] != doc[Shop['shop_logo']]:
93              args[Shop['shop_logo']] = shop_result['shop_img']
94          if shop_result['shop_type'] != '' and shop_result['shop_type'] != doc[Shop['shop_type']]:
95              args[Shop['shop_type']] = shop_result['shop_type']
96          if shop_result['isTmall'] != '' and shop_result['isTmall'] != doc[Shop['is_tmall']]:
97              args[Shop['is_tmall']] = shop_result['isTmall']
98          if shop_result['isTmallGlobal'] != '' and shop_result['isTmallGlobal'] != doc[Shop['is_tmall_global']]:
99              args[Shop['is_tmall_global']] = shop_result['isTmallGlobal']
100         if shop_result['nick'] != '' and shop_result['nick'] != doc[Shop['wangwang']]:
101             args[Shop['wangwang']] = shop_result['nick']
102         if shop_result['uid'] != '' and shop_result['uid'] != doc[Shop['seller_id']]:
103             args[Shop['seller_id']] = shop_result['uid']
104         if shop_result['starts'] != '' and shop_result['starts'] != doc[Shop['starts']]:
105             args[Shop['starts']] = shop_result['starts']
106         if shop_result['phone'] != '' and shop_result['phone'] != doc[Shop['phone']]:
107             args[Shop['phone']] = shop_result['phone']
108         if shop_result['city'] != '' and shop_result['city'] != doc[Shop['city']]:
109             args[Shop['city']] = shop_result['city']
```

```
        args = {}
        if shop_result['shop_name'] != '' and shop_result['shop_name'] != doc[Shop['shop_name']]:
            args[Shop['shop_name']] = shop_result['shop_name']
        if shop_result['shop_img'] != '' and shop_result['shop_img'] != doc[Shop['shop_logo']]:
            args[Shop['shop_logo']] = shop_result['shop_img']
        if shop_result['shop_type'] != '' and shop_result['shop_type'] != doc[Shop['shop_type']]:
            args[Shop['shop_type']] = shop_result['shop_type']
        if shop_result['isTmall'] != '' and shop_result['isTmall'] != doc[Shop['is_tmall']]:
            args[Shop['is_tmall']] = shop_result['isTmall']
        if shop_result['isTmallGlobal'] != '' and shop_result['isTmallGlobal'] != doc[Shop['is_tmall_global']]:
            args[Shop['is_tmall_global']] = shop_result['isTmallGlobal']
        if shop_result['nick'] != '' and shop_result['nick'] != doc[Shop['wangwang']]:
            args[Shop['wangwang']] = shop_result['nick']
        if shop_result['uid'] != '' and shop_result['uid'] != doc[Shop['seller_id']]:
            args[Shop['seller_id']] = shop_result['uid']
        if shop_result['starts'] != '' and shop_result['starts'] != doc[Shop['starts']]:
            args[Shop['starts']] = shop_result['starts']
        if shop_result['phone'] != '' and shop_result['phone'] != doc[Shop['phone']]:
            args[Shop['phone']] = shop_result['phone']
        if shop_result['city'] != '' and shop_result['city'] != doc[Shop['city']]:
            args[Shop['city']] = shop_result['city']
```

# The DRY Principle

# 引子



```
15  shop_fields_map = {
16      'shop_name': Shop['shop_name'],
17      'shop_img': Shop['shop_logo'],
18      'shop_type': Shop['shop_type'],
19      'ismall': Shop['is_tmall'],
20      'isTmallGlobal': Shop['is_tmall_glob
21      'nick': Shop['wangwang'],
22      'uid': Shop['seller
23      'starts': Shop['sta
24      'phone': Shop['phor
25      'city': Shop['city'
26      'prov': Shop['provi
27      'href': Shop['href'
28      'level': Shop['level'],
29      'good-comt': Shop['good_comment'],
30      'collectCount': Shop['collect_count'
31      'itemCount': Shop['item_count'],
32      'totalSold': Shop['total_sold'],
33      'rateSum': Shop['rate_sum'],
34  }
```

```
for result_field_name, db_field_name in shop_fields_map.items():
    field_value = shop_result.get(result_field_name)
    if field_value and field_value != doc.get(db_field_name):
        args[db_field_name] = field_value
```

# 实践

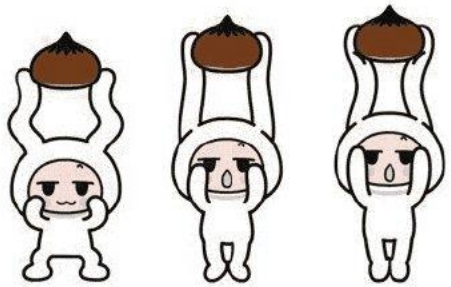


- 语句级
  - 多用assert，但虽用以验证
  - assert(xx, 'yy')假断言
  - 多用with，__enter__()/__exit__()/contextlib
  - 慎用lambda
  - for/while…else…
  - try…except…else…finaly…
- 库级
  - 慎用map()/reduce()/filter()
  - namedtuple是个好东西
  - set/deque/heapq/LifoQueue/PriorityQueue
- 应用级
  - virtualenv
  - pyenv
  - pipenv/peotry

# 实践



- 跟进Python或库的新版本
  - f-strings debugging
    - >>> f"{name=}”
    - "name='karthikeyan’ ”
- RTFM
  - collections.Counter.most_common([n])
- 开发或使用更“现代”的库
- 研发公司内部的业务核心库



多举几个栗子

# 小结



- 低质量代码随时都可能遇到，大厂、老手，也不能幸免。

# 小结



- 低质量代码随时都可能遇到，大厂、老手，也不能幸免。
- 善于利用工具，及早发现劣质代码。

# 小结



- 低质量代码随时都可能遇到，大厂、老手，也不能幸免。
- 善于利用工具，及早发现劣质代码。
- 学习编程原则，提升站位、深化认识，发挥理论的指导作用。

# 小结



- 低质量代码随时都可能遇到，大厂、老手，也不能幸免。
- 善于利用工具，及早发现劣质代码。
- 学习编程原则，提升站位、深化认识，发挥理论的指导作用。
- 在学中做，在做中学，日常CRUD工作中仍然包含很多挑战，平凡工作做出不凡成就，方显水平。

# 小结



- 低质量代码随时都可能遇到，大厂、老手，也不能幸免。
- 善于利用工具，及早发现劣质代码。
- 学习编程原则，提升站位、深化认识，发挥理论的指导作用。
- 在学中做，在做中学，日常CRUD工作中仍然包含很多挑战，平凡工作做出不凡成就，方显水平。
- 持续学习，既跟先进，也跟经典，不骄不躁。

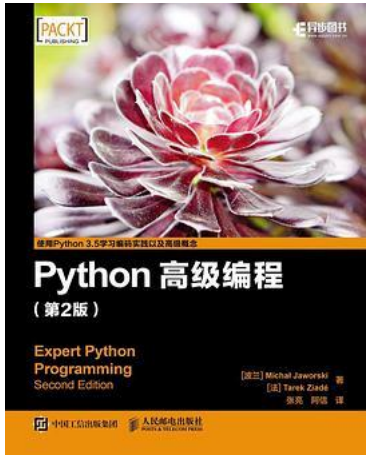
# 小结

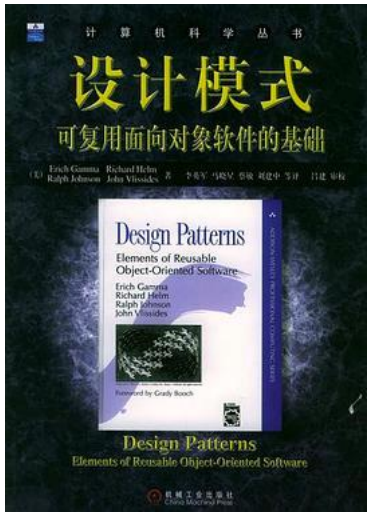


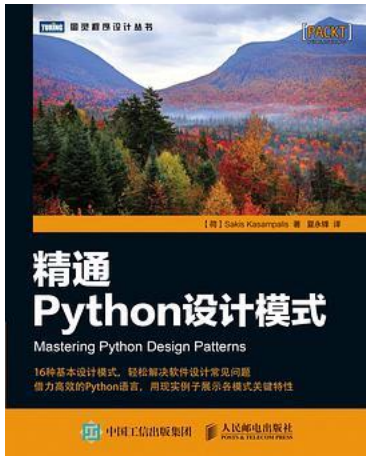
- 低质量代码随时都可能遇到，大厂、老手，也不能幸免。
- 善于利用工具，及早发现劣质代码。
- 学习编程原则，提升站位、深化认识，发挥理论的指导作用。
- 在学中做，在做中学，日常CRUD工作中仍然包含很多挑战，平凡工作做出不凡成就，方显水平。
- 持续学习，既跟先进，也跟经典，不骄不躁。
- 闻过则喜，不耻下问。多总结，多布道。

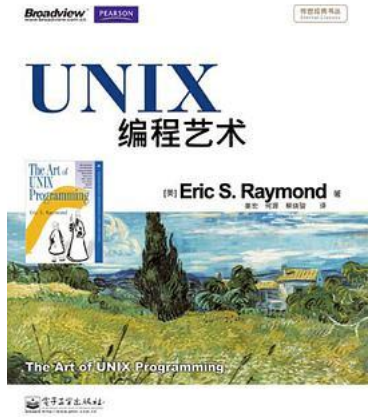
# more...




PACKT
Python 高级编程
（第2版）
Expert Python Programming
Second Edition

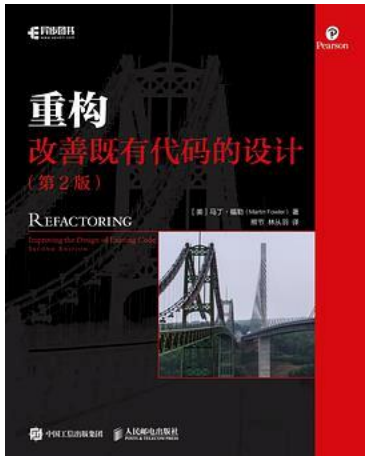

计 算 机 科 学 丛 书
设计模式
可复用面向对象软件的基础
Design Patterns
Elements of Reusable Object-Oriented Software
Design Patterns
Elements of Reusable Object-Oriented Software

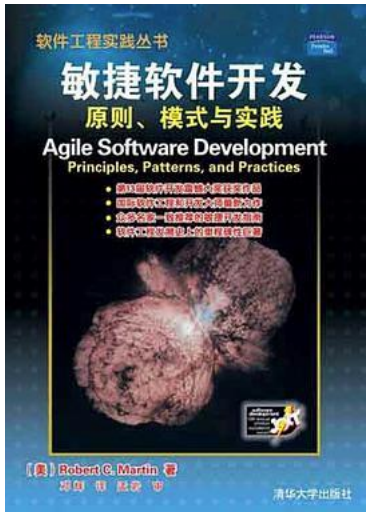

精通
Python设计模式
Mastering Python Design Patterns

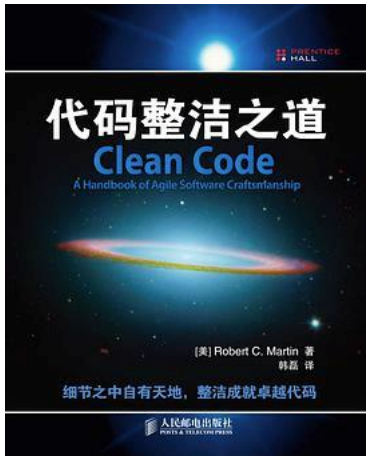

UNIX
编程艺术
Eric S. Raymond
The Art of UNIX Programming


编写可读代码的艺术
The Art of Readable Code
O'REILLY


解析极限编程
——拥抱变化
Extreme Programming explained
EMBRACE CHANGE


重构
改善既有代码的设计
（第2版）
REFACTORING


软件工程实践丛书
敏捷软件开发
原则、模式与实践
Agile Software Development
Principles, Patterns, and Practices
Robert C. Martin
清华大学出版社


代码整洁之道
Clean Code
A Handbook of Agile Software Craftsmanship
[美] Robert C. Martin 著
韩磊 译
细节之中自有天地，整洁成就卓越代码

PyCon China 2020

Python For Good

PyConChina 2020 | PYTHON 中国开发者大会 2020

# THANK YOU.