# **2019-10-19**

线上训练营以训练为主、学课为辅

一开始会不太适应

因为他需要您主动刻意练习

而不是被动等老师讲课

当您完成项目二

从2小时提高到40分钟

才能真正体会到TDD的用处

----------------

http://codingdojo.org/kata/FizzBuzz/

项目一：小试牛刀——FizzBuzz

任务数：4个

预计完成时间：7天

训练目标：

1.掌握测试驱动开发基本节奏

2.至少练习3遍FizzBuzz题目

3.能在10分钟内完成FizzBuzz题目

----------------

需要您现在：

1.PC端登录训练营地址见微信群

（用手机号验证登陆，已绑定）

2.选择【测试驱动开发实战营】开始训练

3.有问题在群里随时和熊老师同步

4.7天后提交项目一作业

----------------

熊老师在群里全程答疑，请务必保持训练反馈

如何提升你的编程效率

<https://apppptz6pch9932.h5.xiaoeknow.com/content_page/eyJ0eXBlIjoxMiwicmVzb3VyY2VfdHlwZSI6NCwicmVzb3VyY2VfaWQiOiJsXzVkYTk5MDAwY2Y1MDlfTHFDa0lTTTciLCJwcm9kdWN0X2lkIjoiIiwiYXBwX2lkIjoiYXBwcHB0ejZQY0g5OTMyIiwiZXh0cmFfZGF0YSI6MH0=?entry=2&entry_type=2001&state=7c32d174260a68d4101ac9af72ff997a_v8H1Kp&app_id=apppptz6PcH9932>

Hi，报名【TDD-测试驱动开发实战营】的小伙伴

## 您未来21天的被虐清单如下：

项目一：小试牛刀——FizzBuzz

任务数：4个

预计完成时间：7天

训练目标：

1.掌握测试驱动开发基本节奏

2.至少练习3遍FizzBuzz题目

3.能在10分钟内完成FizzBuzz题目

项目二：Args

任务数：11个

预计完成时间7天

训练目标：

1. 准确框定需求范围

2. 把需求拆解成明细的任务列表

3.

项目三：MarsRover

预计完成时间：7天

任务数：3

训练目标：

1. 准确框定需求范围

2. 把需求拆解成明细的任务列表

3. 练熟测试驱动开发的节奏

项目三：MarsRover

预计完成时间：7天

任务数：3

训练目标：

1.反复练习测试驱动开发的节奏

2.练习识别和消除代码坏味道

## **您的第一个任务：**

1.登录训练营地址【plus.jiker.com】

2.选择进入【测试驱动开发实战营开始训练】

3.有问题可在群里随时和熊老师同步

4.7天后提交项目一作业

# 2019.10.20

<https://github.com/Choosue/tdd>

@熊节 Jeff Xiong 算上搭环境，一共用了 40 分钟……请熊老师指导一下，有什么问题？

第一条规则：没有失败的测试就不能写代码

第二条规则：只允许写刚好让测试通过的代码

①项目二Args，一个一般的开发者，完成时间是2-4小时，经过反复训练你会在27分钟内完成。

②实战营不是视频课，只有当你真正开始刻意训练才能收益。

③今天开始训练，我们会在最先的50条打卡记录里选出前3，赠送小米人工智能音响，打卡形式详见实战营。

TDD演示视频1、2: Python老司机实战FizzBuzz

<https://mp.weixin.qq.com/s/c45YeR6Re6FmsA4K7z2c-A>

<https://mp.weixin.qq.com/s/kMaGRKXy_cSVqe0zxQisoA>

# 2019.10.22

Kent Beck的《测试驱动开发》，可以只读第1章

各位同学，如果FizzBuzz做得很顺，5分钟内能完成，可以玩玩这道题：

美国人习惯使用很古怪的英制度量单位。英制度量单位的换算经常不是十进制的，比如说：

\* 1英尺（foot）= 12英寸（inch）

\* 1码（yard）= 3英尺

\*\*任务：写一个程序，用于处理英寸、英尺、码之间的换算。\*\*

你的程序应该能很好地处理下面这些换算关系：

\* 1英尺 应该等于 12英寸

\* 1码 应该等于 3英尺

\* 1码 应该等于 36英寸

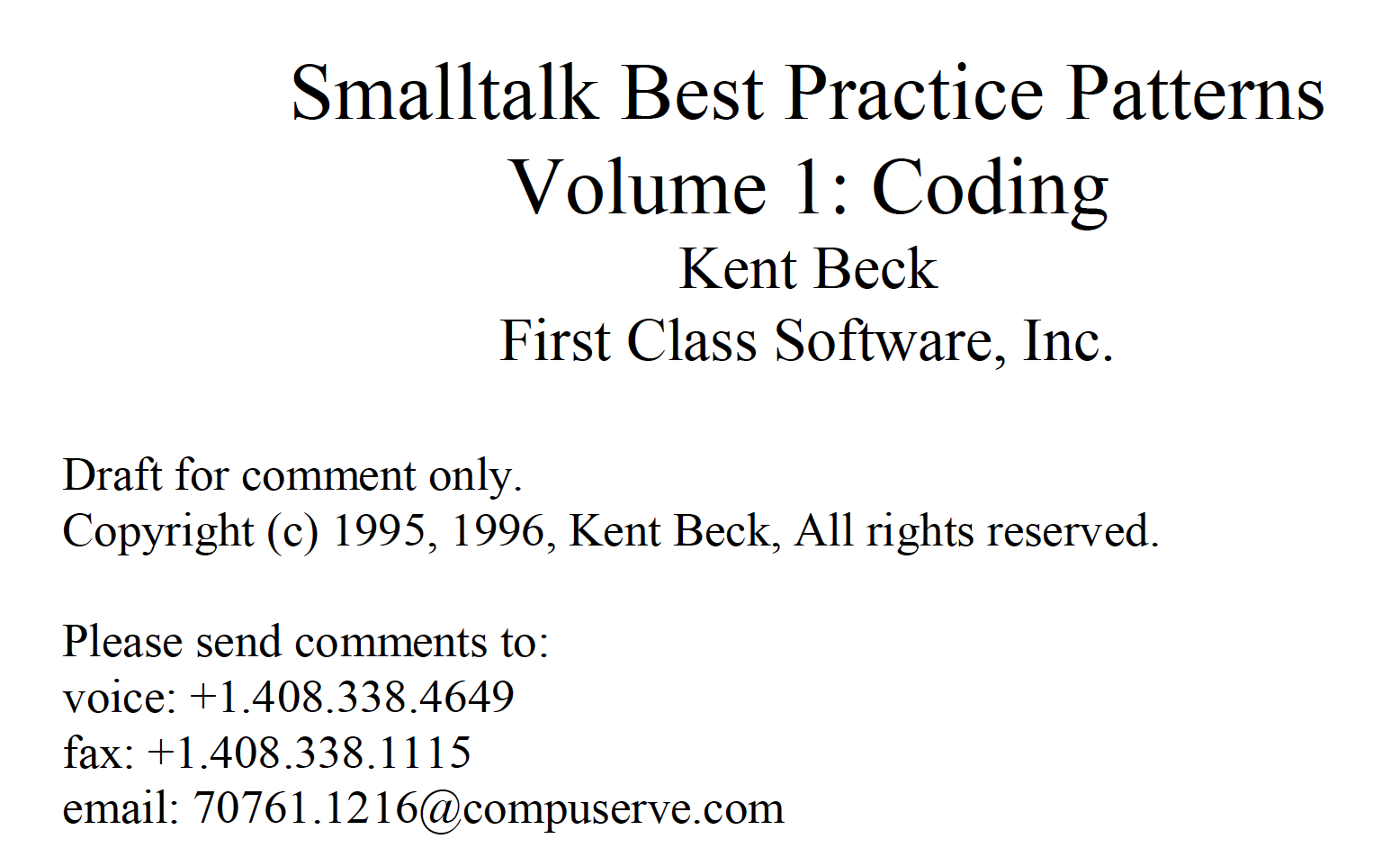
Q：FizzBuzz项目在接到需求变更后，是修改之前的测试好呢？还是新增测试好呢？

熊：如不必要，最好新增测试，放在相关测试用例附近。原有业务逻辑还存在，所以原来的测试也原样存在，一般倾向于一个测试讲一件事，可以运行测试以后不看测试代码，只看测试报告，看是否说清楚了软件需求。

# 2019.10.23

<https://technologyconversations.com/2013/12/24/test-driven-development-tdd-best-practices-using-java-examples-2/>

实现模式的前身



# Summary

　　　1.测试驱动开发，是一种“开发方法”，不是测试方法；

　　　写代码前先写个测试，

　　　没有“开发方法”能“想到”你本来没有想到的事——这事交个测试就好了

　　　2.开发问题颗粒

　　　1）测试红色

　　　2）开发代码->绿色，（不超过2分钟）

　　　两者不超过五分钟；（问题的颗粒大小）

　　　3）重构（20秒）

　　　“颗粒”与“速度”紧密相关

　　　速度快，步子可以大一点；反之亦然；

　　　3.科学上网 + 英文作为主要工作语言

　　　4.

　　　5.

　　　6.

　　　7.

# Question:

　　　1.没有输出的函数如何测试？

　　　2.是不是应该遵循"只允许写刚好让测试通过的代码"

　　　那个步子时而大，时而小，随性而为。但主体原则是按照TDD方式来做的。这个可以根据自己对业务的熟练度来调整，个人认为。

　　　3.

　　　4.

　　　5.

　　　6.

# Task

　　　1.Kata练习

　　　2.

　　　3.

　　　4.

　　　5.

　　　6.