

# Code Dojo



# What is Test-Driven Development?

---

Double-entry accounting for your code

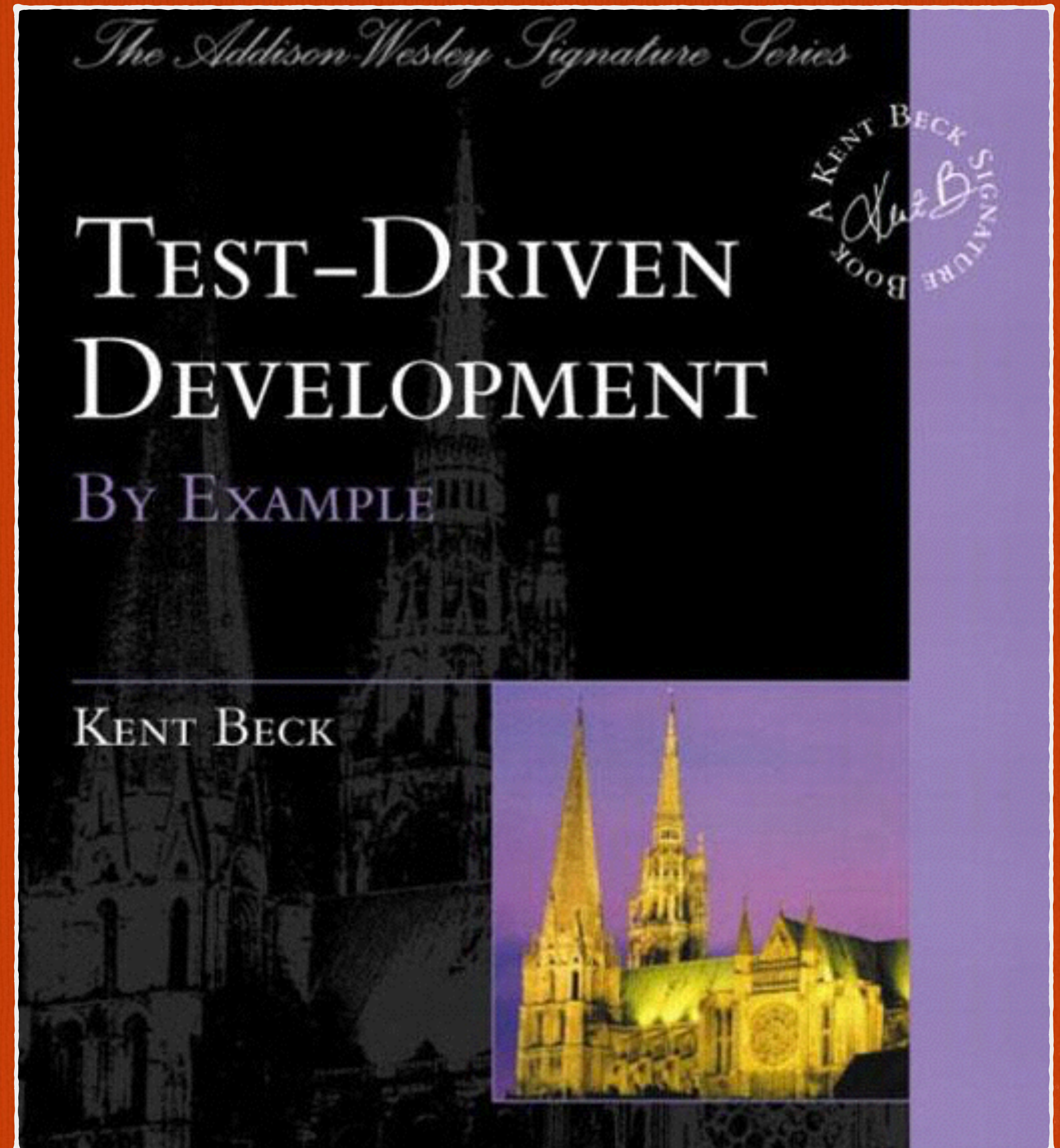
# The Book

---

Test-Driven Development By Example (2002)

By: Kent Beck

<https://www.amazon.ca/Test-Driven-Development-Kent-Beck/dp/0321146530>



# What is Test Driven Development?

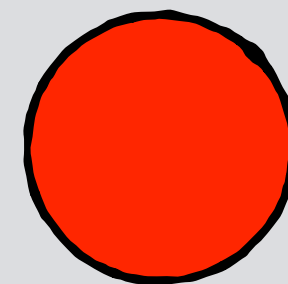
---

- Unit Tests (read: small, low/no dependencies)
- Cycle:
  - Step 1: Write a minimal failing test **RED**
  - Step 2: Write production code to make the test pass **GREEN**
  - Step 3: Refactor **REFACTOR**
  - Step 4: Back to Step 1

# Example: Write a Rectangle Class (Part 1)

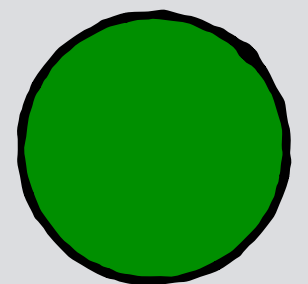
## Test Code

```
class TestRectangle(unittest.TestCase):  
  
    def test_length(self):  
        r = Rectangle(length=10)  
        self.assertEqual(r.length, 10)
```

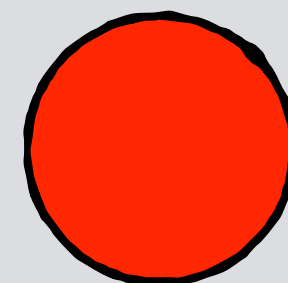


## Production Code

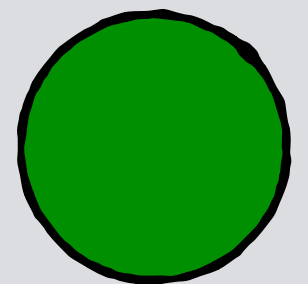
```
class Rectangle(object):  
  
    def __init__(self, length=0.0):  
        self.length = length
```



```
class TestRectangle(unittest.TestCase):  
  
    def test_length(self):  
        r = Rectangle(length=10, width=20)  
        self.assertEqual(r.length, 10)  
  
    def test_width(self):  
        r = Rectangle(length=10, width=20)  
        self.assertEqual(r.width, 20)
```

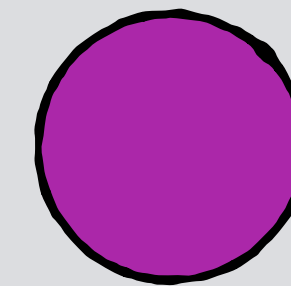


```
class Rectangle(object):  
  
    def __init__(self, length=0.0, width=0.0):  
        self.length = length  
        self.width = width
```



## Example: Write a Rectangle Class (Part 2)

```
class Rectangle(object):  
  
    def __init__(self, length=0.0, width=0.0):  
        self.length = length  
        self.width = width  
  
    @property  
    def length(self):  
        return self._length  
  
    @length.setter  
    def length(self, new_length):  
        self._length = new_length  
  
    @property  
    def width(self):  
        return self._width  
  
    @width.setter  
    def width(self, new_width):  
        self._width = new_width
```



## Example: Write a Rectangle Class (Part 3)

```
class Rectangle(object):

    def __init__(self, length=0.0, width=0.0):
        self.length = length
        self.width = width

    @property
    def length(self):
        return self._length

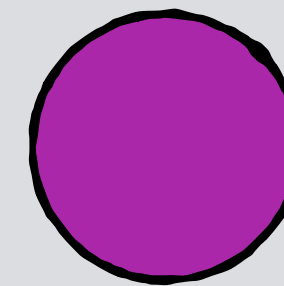
    @length.setter
    def length(self, new_length):
        if not new_length:
            raise ValueError(u"Invalid length. Length must be a real number that is not 0")
        self._length = new_length

    @property
    def width(self):
        return self._width

    @width.setter
    def width(self, new_width):
        if not new_width:
            raise ValueError(u"Invalid length. Length must be a real number that is not 0")
        self._width = new_width

    @property
    def area(self):
        return self.length * self.width

    @property
    def perimeter(self):
        return (2 * self.length) + (2 * self.width)
```



**Video: “Uncle Bob - Expecting  
Professionalism” Question Period**



# What is Code Dojo?

# What is a Code Dojo?

---

- Focus on testing
- Deliberate practice
- Working with others
- Loud, creative, and fun

# What's Gonna Happen?

---

- Tests first!
- Work with a partner
- 1st Round: 15 minutes
- Quick round table and progress update
- 2nd Round: 15 minutes
- Show and Tell

**Break Time**

# **The Problem**

# The Problem

---

## ANAGRAMS

Given a file containing one word per line, print out all the combinations of words that are anagrams; each line in the output contains all the words from the input that are anagrams of each other.

**BONUS:** Find the longest words that are anagrams, and find the set of anagrams containing the most words.

# Example Outputs

---

kinship pinkish

enlist inlets listen silent

boaster boaters borates

fresher refresh

sinks skins

knits stink

rots sort

crepitus cuprites pictures piecrust

paste pates peats septa spate tapes tepas

punctilio unpolitic

sunders undress