Throw Me the Idol

Uncertainty & Untestable Code

@deejaygraham

sage

Developer!
DeveloperDeveloper
UK Community Events
Noisebridge Hacker Space
https://noisebridge.net
83c.wiese.sf.ca.usa

https://ti.to/ne-bytes/

Change
is the
only
constant

Bob Ross

Google

how do I test my code

how do i test my code

how do i test my **html** code

how do i test my **python** code

how do i test my **javascript** code

how do i test my **postman** code

how **to** test my code

how **can** i test my **html** code

how **can** i test my **php** code

how **can** i test my **java** code

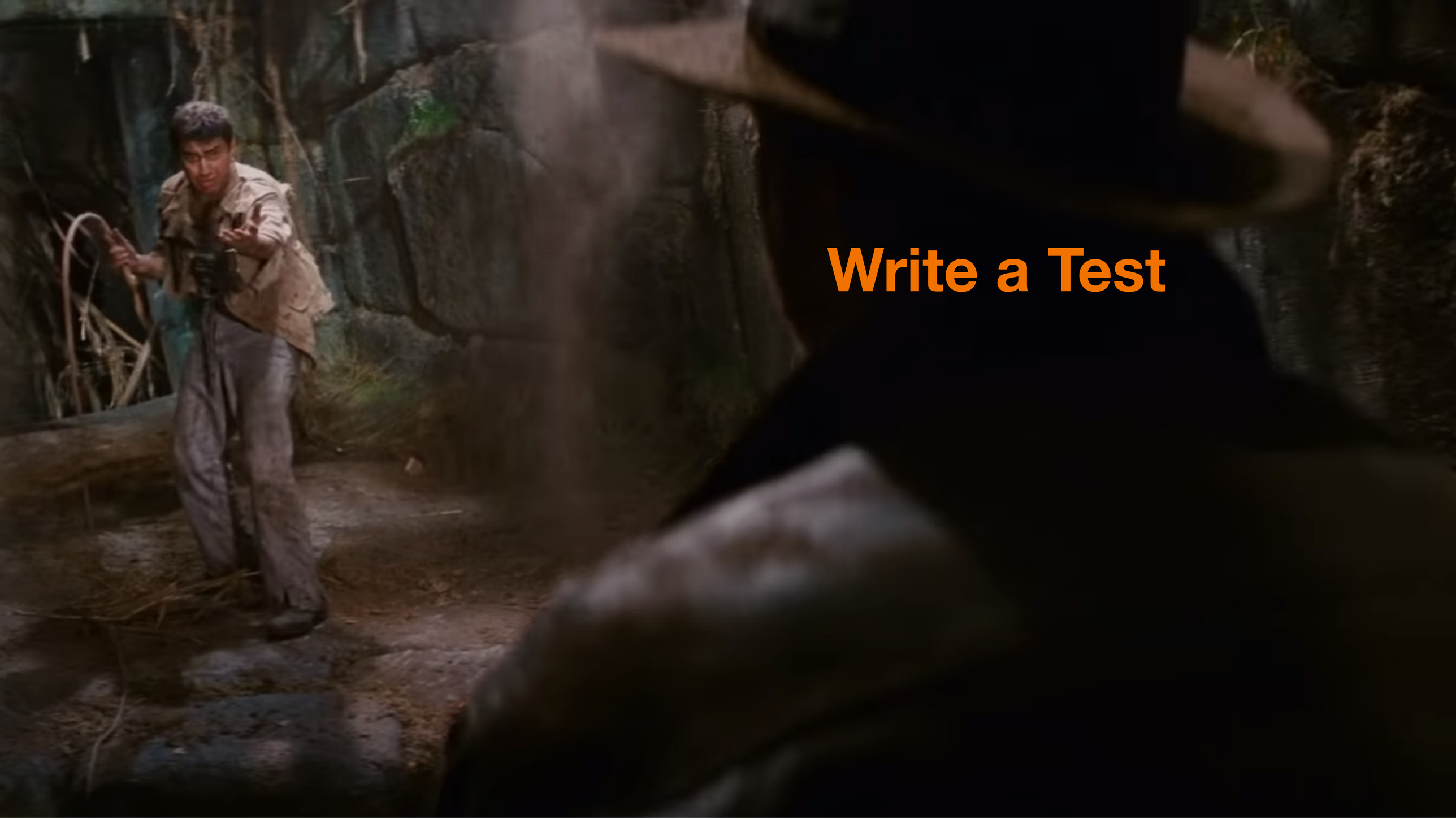
How to draw an owl

1.

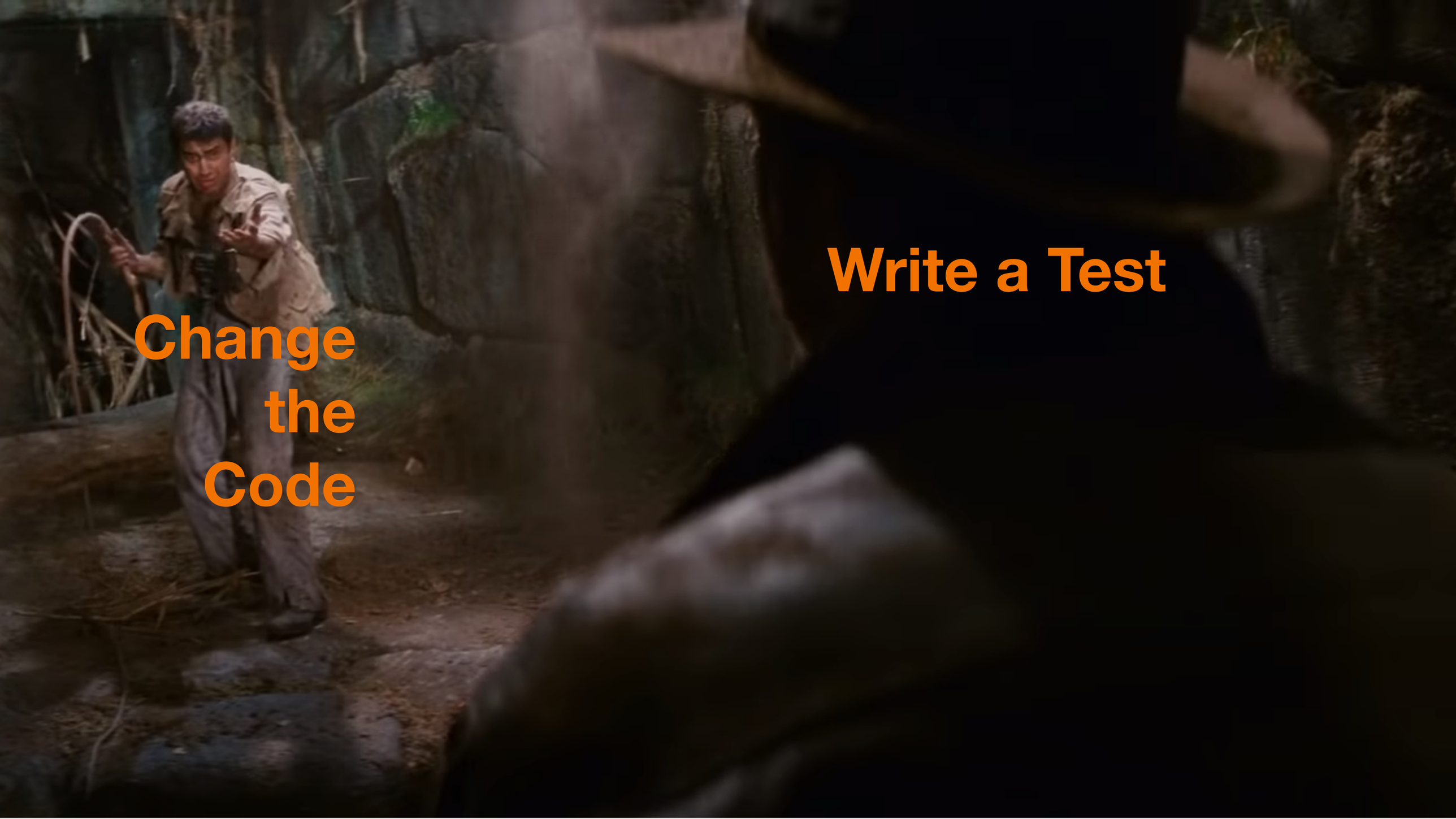
2.

1. Draw some circles

2. Draw the rest of the fucking owl



Write a Test

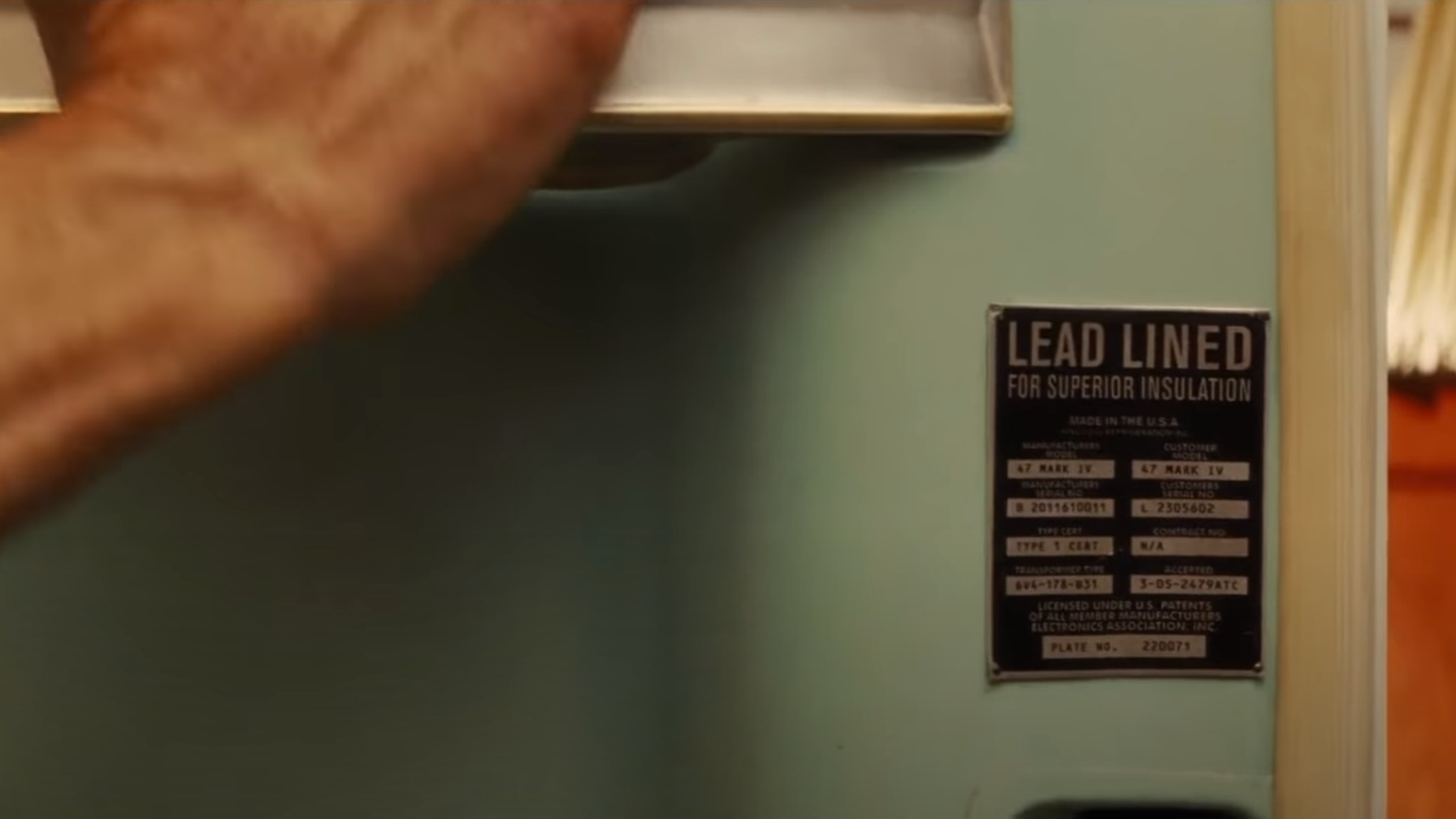

Write a Test
Change the Code

<small>Start</small>

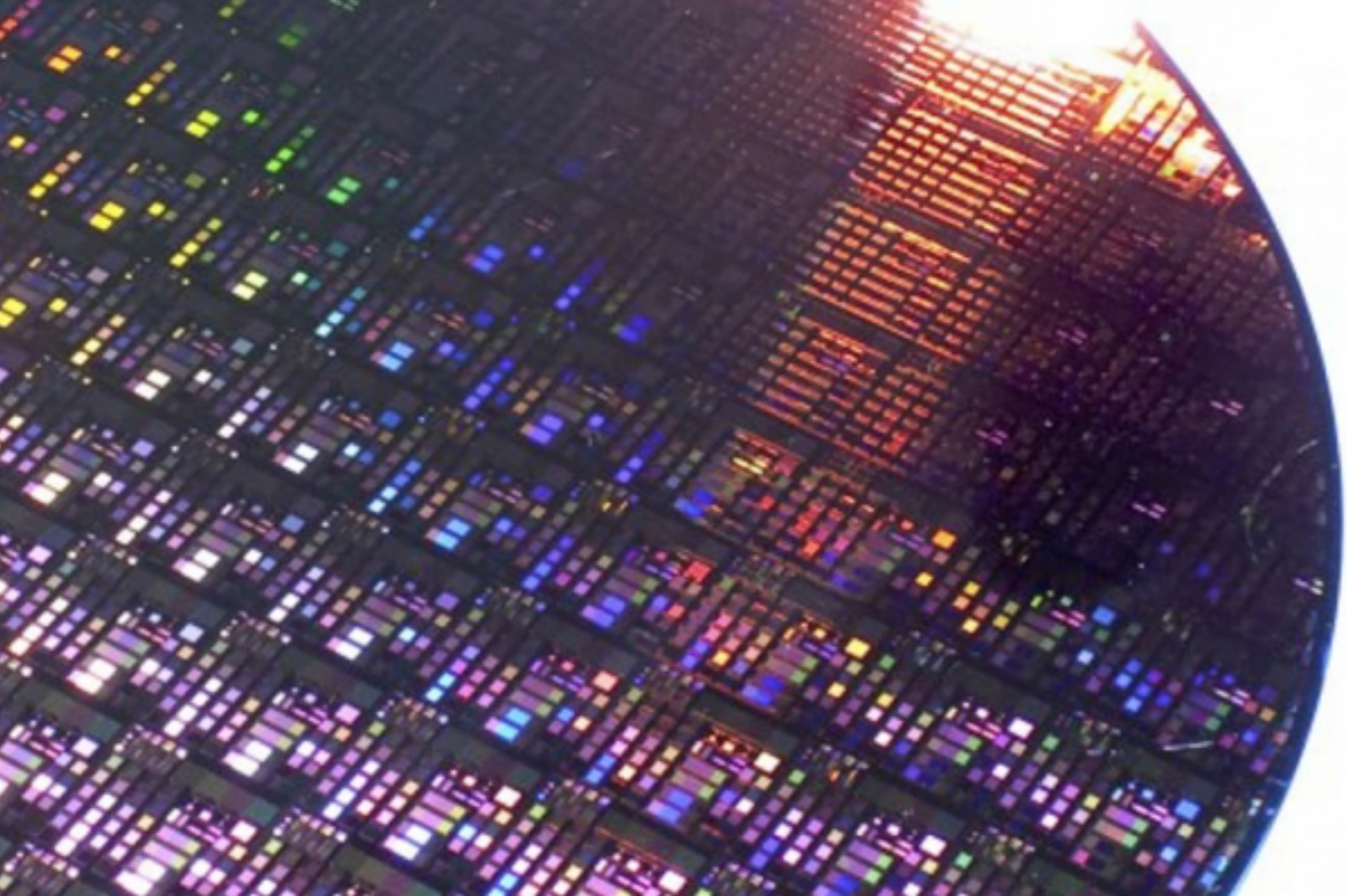
Unit Tests Can't...

Network
Database
File System
Clock & Date

4 3 7 7 0 9 2 9 8 5 1 1 6 2

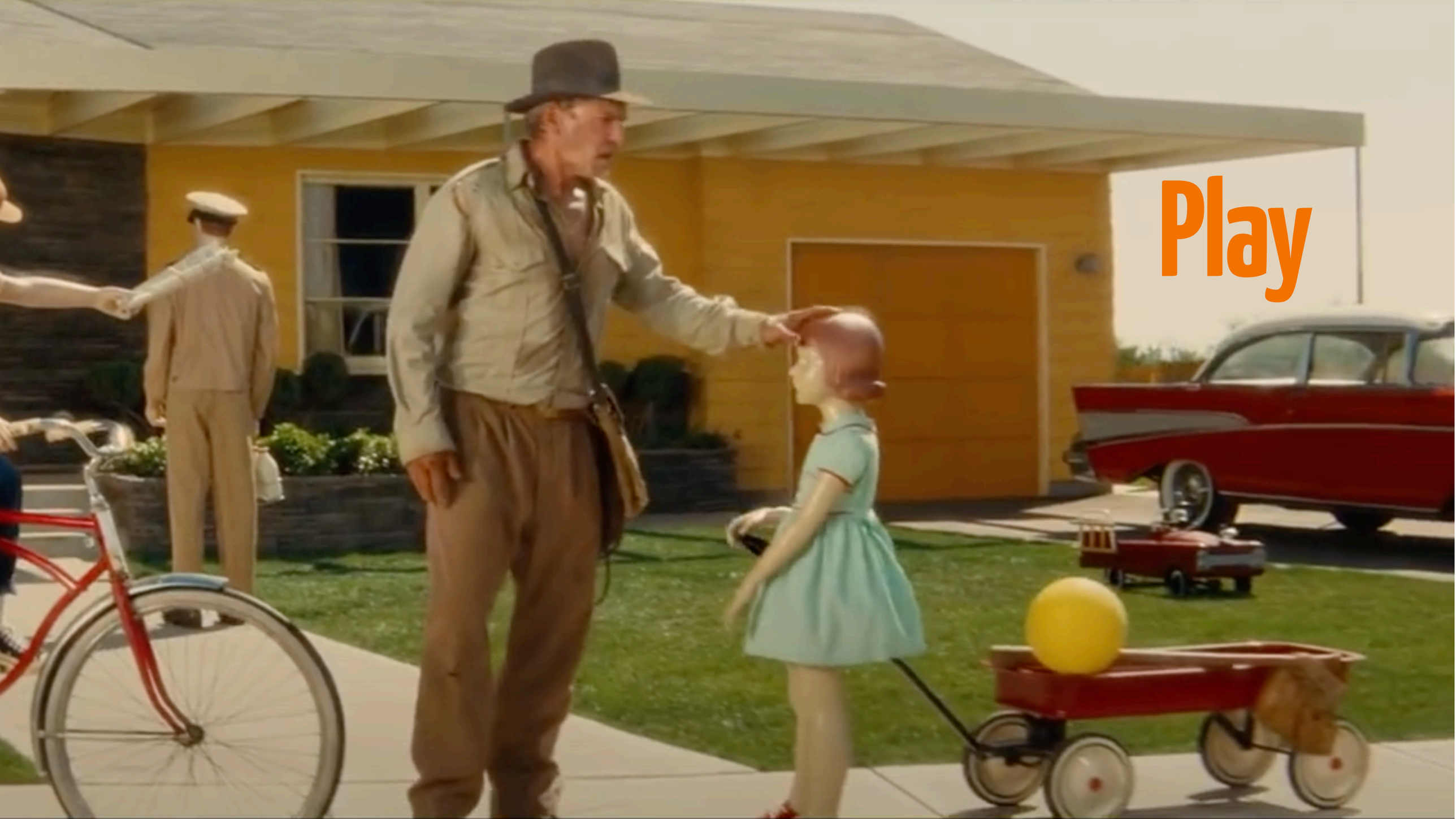

LEAD LINED
FOR SUPERIOR INSULATION
MADE IN THE U.S.A.
MANUFACTURERS MODEL
47 MARK IV
CUSTOMER MODEL
47 MARK IV
MANUFACTURERS SERIAL NO.
B 2011610011
CUSTOMERS SERIAL NO.
L 2305602
TYPE CERT.
TYPE 1 CERT.
CONTRACT NO.
N/A
TRANSFORMER TYPE
6V4-178-B31
ACCEPTED
3-05-2479ATC
LICENSED UNDER U.S. PATENTS
OF ALL MEMBER MANUFACTURERS
ELECTRONICS ASSOCIATION, INC.
PLATE NO. 220071

Fakes



Switches

Debuggers


Play

Text

```
            {
                if (Items[i].Quality > 0)
                {
                    if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
                    {
                        Items[i].Quality = Items[i].Quality - 1;
                    }
                }
            }
            else
            {
                if (Items[i].Quality < 50)
                {
                    Items[i].Quality = Items[i].Quality + 1;

                    if (Items[i].Name == "Backstage passes to a TAFKAL80ETC concert")
                    {
                        if (Items[i].SellIn < 11)
                        {
                            if (Items[i].Quality < 50)
                            {
                                Items[i].Quality = Items[i].Quality + 1;
                            }
                        }

                        if (Items[i].SellIn < 6)
                        {
                            if (Items[i].Quality < 50)
                            {
                                Items[i].Quality = Items[i].Quality + 1;
                            }
                        }
                    }
                }
            }
```

```
{
    IList<Item> Items = new List<Item>{
        new Item {Name = "+5 Dexterity Vest", SellIn = 10, Quality = 20},
        new Item {Name = "Aged Brie", SellIn = 2, Quality = 0},
        new Item {Name = "Elixir of the Mongoose", SellIn = 5, Quality = 7},
        new Item {Name = "Sulfuras, Hand of Ragnaros", SellIn = 0, Quality = 80},
        new Item {Name = "Sulfuras, Hand of Ragnaros", SellIn = -1, Quality = 80},
        new Item
        {
            Name = "Backstage passes to a TAFKAL80ETC concert",
            SellIn = 15,
            Quality = 20
        },
        new Item
        {
            Name = "Backstage passes to a TAFKAL80ETC concert",
            SellIn = 10,
            Quality = 49
        },
        new Item
        {
            Name = "Backstage passes to a TAFKAL80ETC concert",
            SellIn = 5,
            Quality = 49
        },
        // this conjured item does not work properly yet
        new Item {Name = "Conjured Mana Cake", SellIn = 3, Quality = 6}
    };

    var app = new GildedRose(Items);
```

```
public class Item
{
    public string Name { get; set; }
    public int SellIn { get; set; }
    public int Quality { get; set; }

    public override string ToString()
    {
        return this.Name + ", " + this.SellIn + ", " + this.Quality;
    }
}
```

```
for (var i = 0; i < 31; i++)
{
    Console.WriteLine("-------- day " + i + " ---------");
    Console.WriteLine("name, sellIn, quality");
    for (var j = 0; j < Items.Count; j++)
    {
        System.Console.WriteLine(Items[j]);
    }
    Console.WriteLine("");
    app.UpdateQuality();
}
```

```
-------- day 0 --------
name, sellIn, quality
+5 Dexterity Vest, 10, 20
Aged Brie, 2, 0
Elixir of the Mongoose, 5, 7
Sulfuras, Hand of Ragnaros, 0, 80
Sulfuras, Hand of Ragnaros, -1, 80
Backstage passes to a TAFKAL80ETC concert, 15, 20
Backstage passes to a TAFKAL80ETC concert, 10, 49
Backstage passes to a TAFKAL80ETC concert, 5, 49
Conjured Mana Cake, 3, 6
```

```
Backstage passes to a TAFKAL80ETC concert, 14, 21
Backstage passes to a TAFKAL80ETC concert, 9, 50
Backstage passes to a TAFKAL80ETC concert, 4, 50
Conjured Mana Cake, 2, 5

-------- day 2 --------
name, sellIn, quality
+5 Dexterity Vest, 8, 18
Aged Brie, 0, 2
Elixir of the Mongoose, 3, 5
Sulfuras, Hand of Ragnaros, 0, 80
Sulfuras, Hand of Ragnaros, -1, 80
Backstage passes to a TAFKAL80ETC concert, 13, 22
Backstage passes to a TAFKAL80ETC concert, 8, 50
Backstage passes to a TAFKAL80ETC concert, 3, 50
Conjured Mana Cake, 1, 4
```

Tests

```
public class BigScaryObject
{
    public void DoStuff()
    {
        // billions of lines here...
    }

    public bool HasDoneStuff { get; private set; }
}
```

```
static void Main(string[] args)
{
    MyHomemadeTestFramework.RunAllTheTests();

    // real application follows ...
```

```
public static class MyHomemadeTestFramework
{
    public static void Assert(bool condition, string message)
    {
        if (!condition)
        {
            Console.WriteLine(message);
            // or even terminate the program
            // Environment.FailFast(message);
        }
    }
```

```
public static void Test_ScaryObject_Can_DO_Stuff()
{
    var obj = new BigScaryObject();
    obj.DoStuff();

    Assert(obj.HasDoneStuff, "Scary Object must be able to do stuff");
}
```

```
public static void RunAllTheTests()
{
    Test_ScaryObject_Can_DO_Stuff();
    // ...more tests
}
```

Refactoring

Deep Nesting
Bad Names
Improper Scope
Long Methods

Rename
Inline
Extract Method
Introduce Local Variable
Introduce Parameter
Introduce Field

Resources

- Emily Bache - http://coding-is-like-cooking.info
- Fowler - Refactoring 2nd Ed
- Feathers - Working Effectively with Legacy Code
- James Shore - Testing without Mocks https://www.jamesshore.com