

Creating an Automated Testing Framework With Selenium

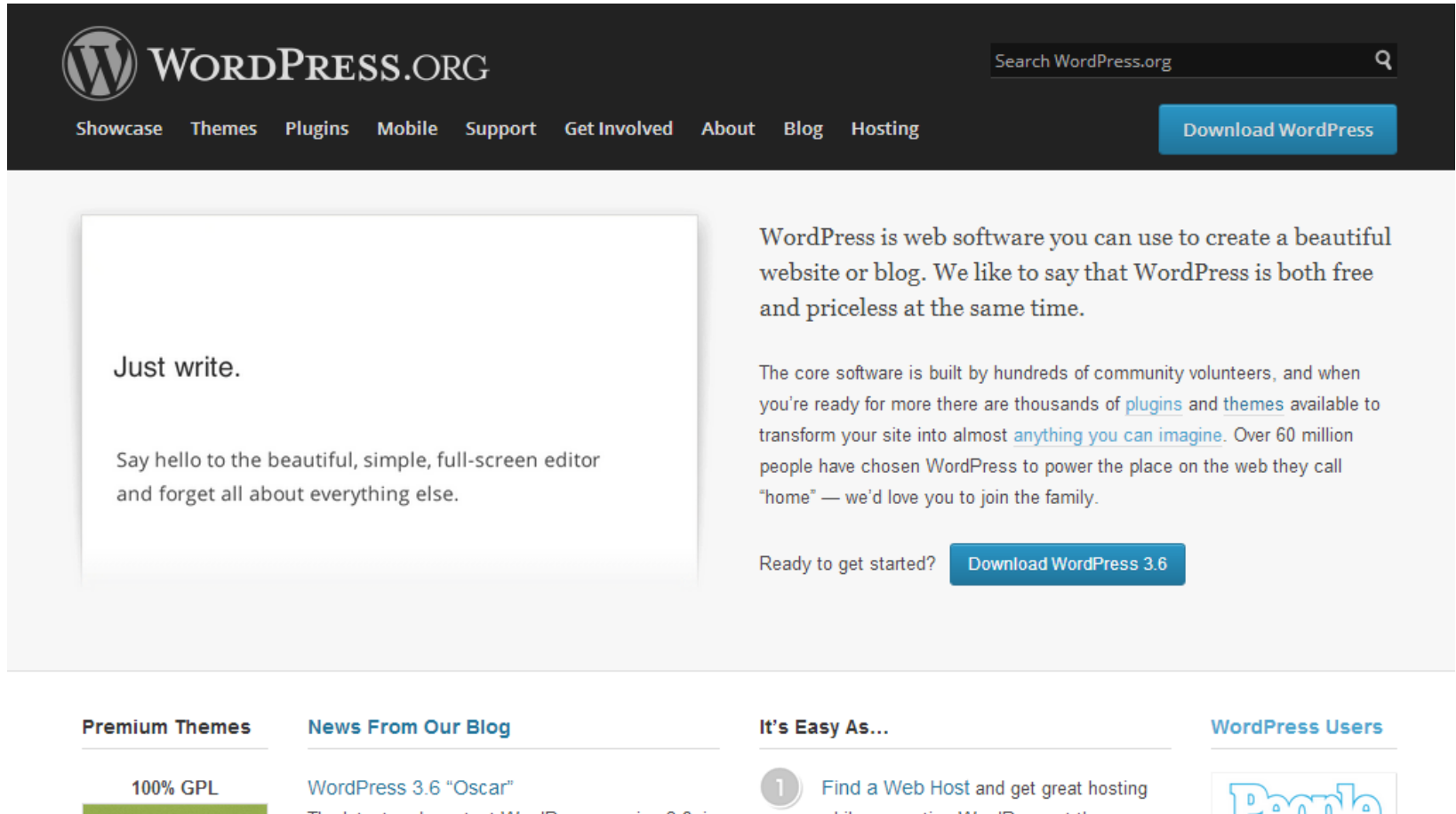
Designing The Architecture

John Sonmez
<http://simpleprogrammer.com>
@jsonmez



pluralsight 
hardcore developer training

The System Under Test (SUT)



The screenshot shows the WordPress.org homepage. At the top is a dark navigation bar with the WordPress logo and 'WORDPRESS.ORG' on the left, a search bar on the right, and a list of links (Showcase, Themes, Plugins, Mobile, Support, Get Involved, About, Blog, Hosting) in the center. A blue 'Download WordPress' button is on the right. Below the navigation bar, the main content area features a large white box on the left with the text 'Just write.' and 'Say hello to the beautiful, simple, full-screen editor and forget all about everything else.' To the right of this box, there is a paragraph about WordPress being free and priceless, followed by another paragraph about the community and the availability of plugins and themes. Below these paragraphs is a 'Ready to get started?' link and a 'Download WordPress 3.6' button. At the bottom, there are four sections: 'Premium Themes' with a '100% GPL' badge, 'News From Our Blog' with a link to 'WordPress 3.6 "Oscar"', 'It's Easy As...' with a numbered list starting with '1 Find a Web Host and get great hosting', and 'WordPress Users' with a 'Danh' logo.

WORDPRESS.ORG

Search WordPress.org

Showcase Themes Plugins Mobile Support Get Involved About Blog Hosting

Download WordPress

Just write.

Say hello to the beautiful, simple, full-screen editor and forget all about everything else.

WordPress is web software you can use to create a beautiful website or blog. We like to say that WordPress is both free and priceless at the same time.

The core software is built by hundreds of community volunteers, and when you're ready for more there are thousands of [plugins](#) and [themes](#) available to transform your site into almost [anything you can imagine](#). Over 60 million people have chosen WordPress to power the place on the web they call "home" — we'd love you to join the family.

Ready to get started? [Download WordPress 3.6](#)

Premium Themes

100% GPL

News From Our Blog

WordPress 3.6 "Oscar"

It's Easy As...

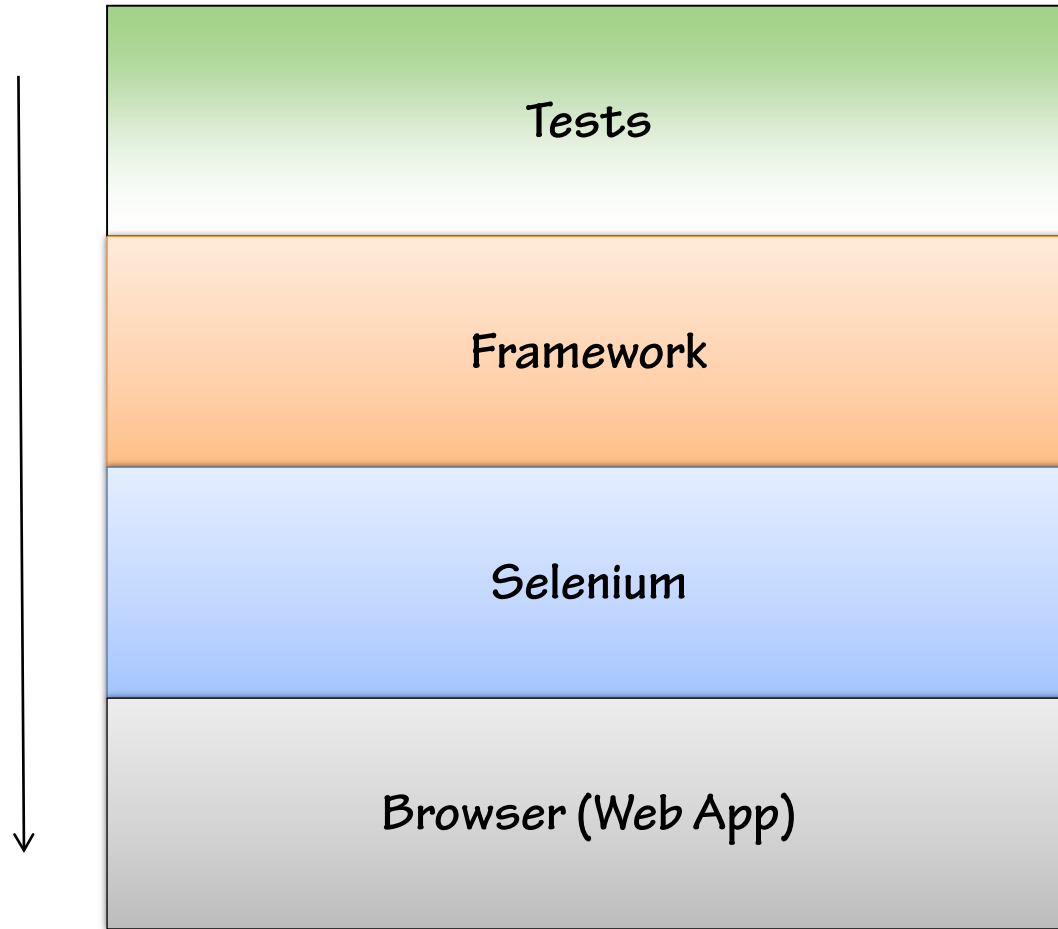
1 Find a Web Host and get great hosting

WordPress Users

Danh

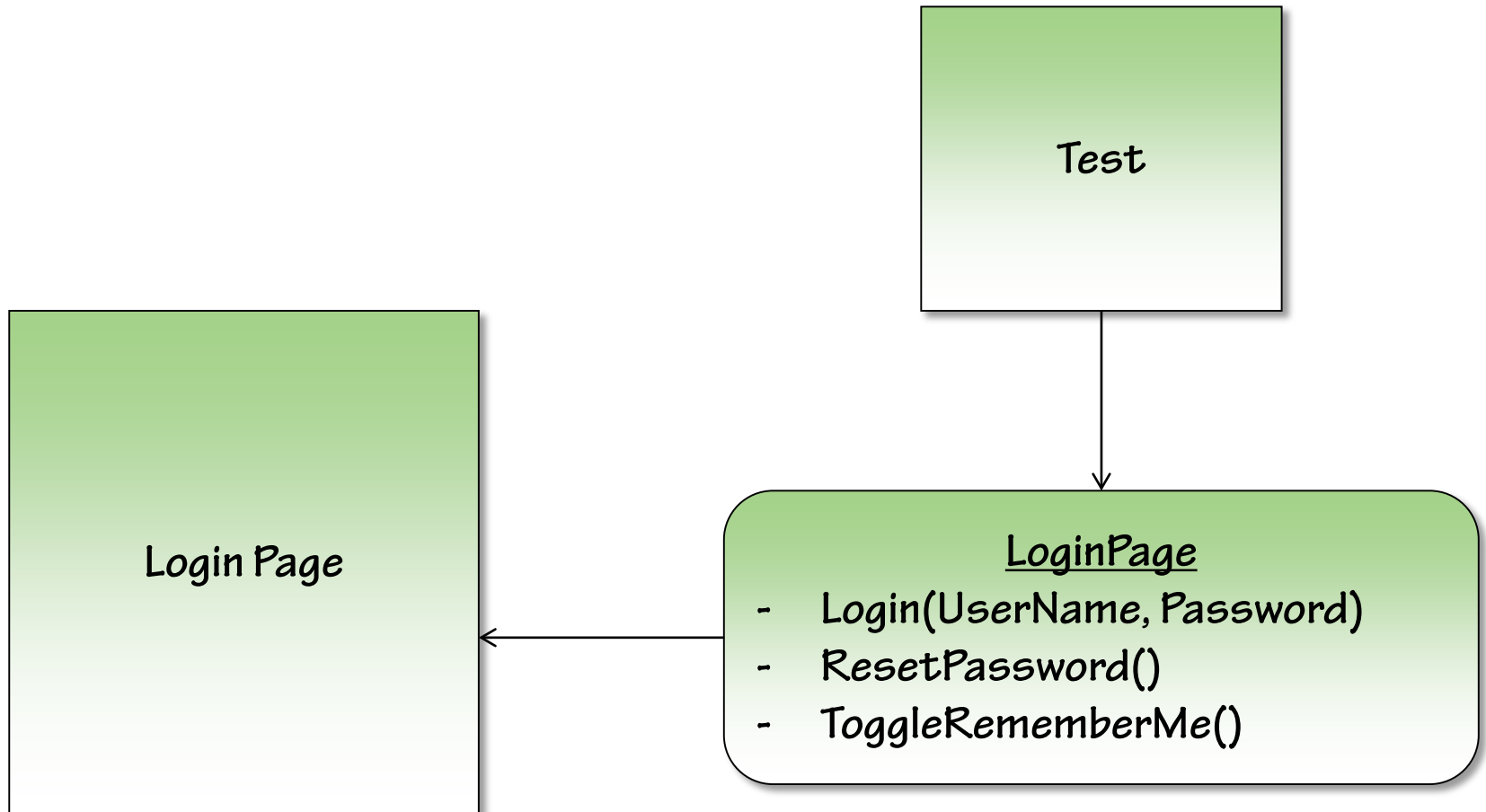
We'll be writing a framework to test the Wordpress blogging engine's backend

Architecture



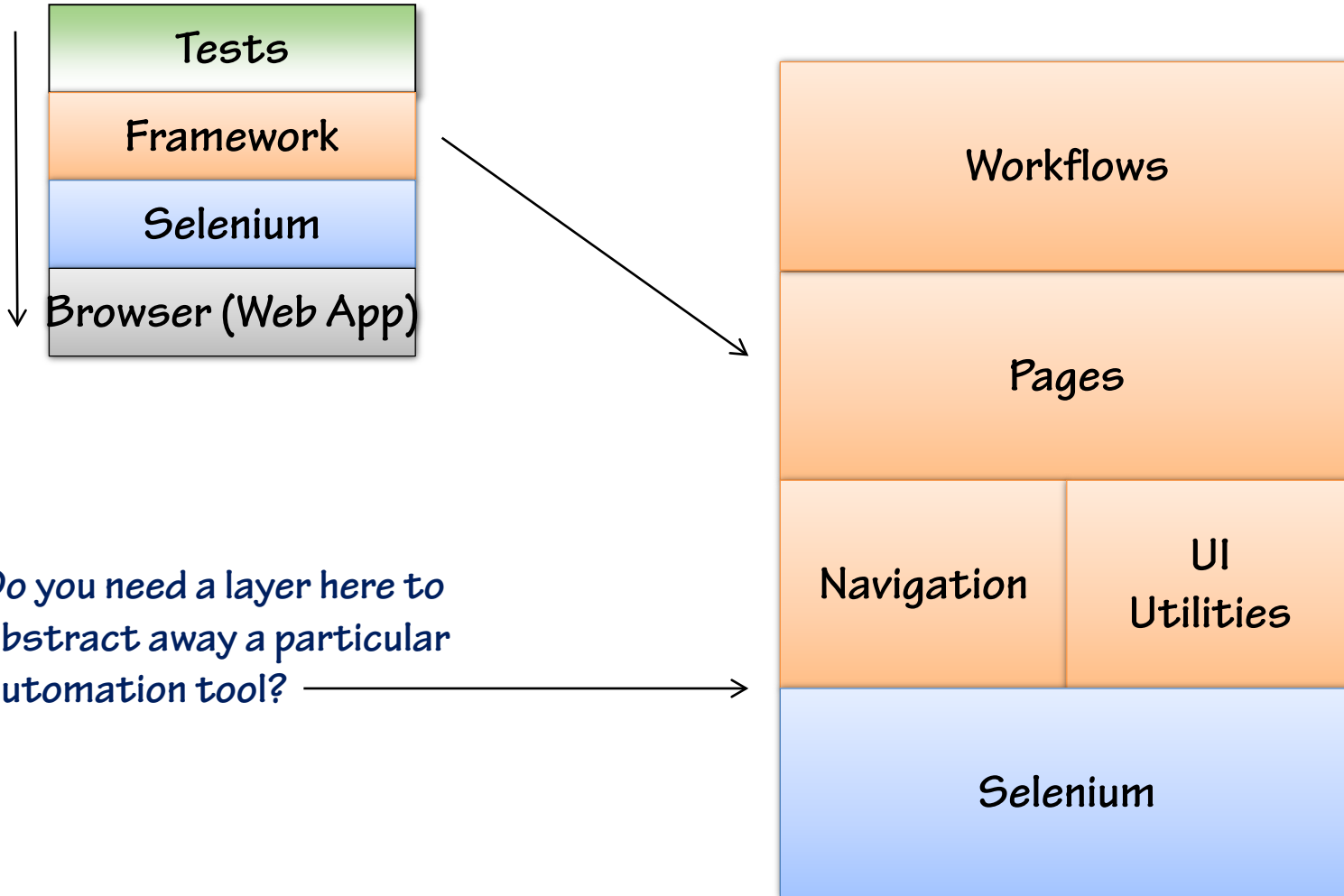
Each layer only interacts with the layer immediately below it

The Page Pattern



Pages are a good way to model the functionality of an application

Thinking About Layers



The framework itself should be divided up by layers of functionality

Thinking About Ease Of Use



VS



Which Is Easier?

```
1 var driver = new Driver();  
2 var loginPage = new LoginPage(driver);  
3 loginPage.Goto();  
4 loginPage.EnterUserName("Joe");  
5 loginPage.EnterPassword("$securitE");  
6 loginPage.LoginButton.Click();  
7
```

OR

```
8 LoginPage.Goto();  
9 LoginPage.Login("Joe", "$securitE");  
10
```

The choice makes a HUGE difference

Some Basic Rules (NEVER)

1. NEVER require tests to declare variables
2. NEVER require test to use the “new” keyword or create new objects
3. NEVER require the tests to manage state on their own
4. NEVER expose the browser or DOM to the tests or let them manipulate it directly



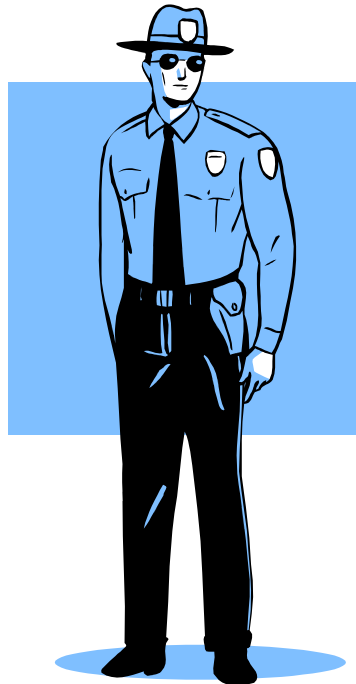
Some Basic Rules (ALWAYS)

1. **ALWAYS** reduce the number of parameters for API calls when possible
2. **ALWAYS** use default values instead of requiring parameters when possible



Some Basic Rules (PREFER)

1. **PREFER** to make the API easier to use over making the internals of the API less complex
2. **PREFER** using enumerations and constants to requiring the test to pass in primitive types



I Thought Static Methods Were Bad...



We are valuing ease of use over maintenance for the framework itself

Domain Specific Languages (DSLs)



Our automation framework should expose a DSL for our tests to use

Internal DSLs

```
8 LoginPage.Goto();  
9 LoginPage.Login("Joe", "$securitE");  
0
```

Still using the programming language our framework is in, but writing the code in a way specific to our domain

External DSLs

```
1 Use LoginPage
2 DoAction Login With
3   UserName : "Joe"
4   Password : "$securitE"
5 EndAction
```

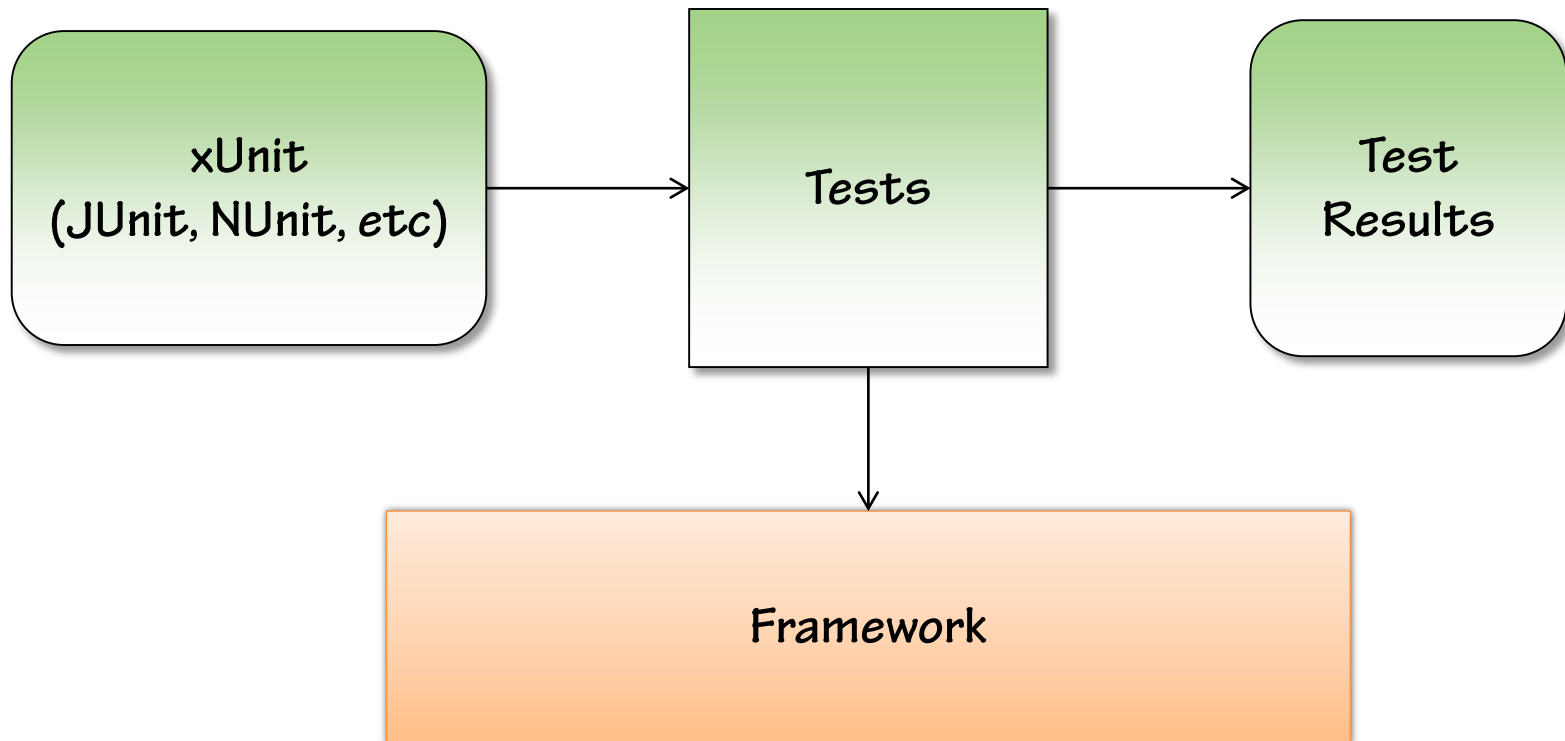
You can create a whole new language to write your tests in, but that is quite a bit of work

Thinking About Running The Tests



Part of designing the framework is designing how the test will run

Running Tests



Consider up front how to run the tests and get output from the tests

Up Next



Don't let this happen to your application