



# TESTING IN FLUTTER AND DART (for the lazy)

---

A quick-fire introduction to testing logic and UI.

Oct 29, 2021

# You don't test enough.

*If you do, that's great!*

## Why test?

One word: confidence.

- Current code works as expected.
- Past code doesn't cause bugs.
- Code will change over time (and still work).

# BuT wHaT dO??!?

*Don't panic.*

## What should I test?

Whatever you want.

- At VGV, we test **what gives the most value**.  
Logic, widgets, context differences, navigation, and more.
- You *can* test anything.



# But how do I test? 🤔

*Glad you asked.*

- Multiple types of tests.
- Generally, try to test for **output**.
- Let's get to it.

# Testing in practice

*Let's see it in action.*

## Test

A block of code that **performs some operation** and makes **assertions** about the result.

# Testing in practice

*Let's see it in action.*

## Groups

Allows grouping a set of related tests.

Supports string concatenation of test names.

# Testing in practice

*Let's see it in action.*

**expect**

Make assertions in testing.

Attempts to *match* A against B.



# Testing in practice

*Let's see it in action.*

## Matchers

Defines how to check or assert values.

*A **equals** B, “hi there” **contains** “hi”,  
this stream **emits** [1, 2, 3]*

# Testing in practice

*Let's see it in action.*

## The **WidgetTester**

A mechanism to test widgets. (Duh.)

Creates a widget tree that can be interacted with.

*pump* the app, *tap* on this button,  
get me the **instance** of this widget,  
find the **state** of my Scaffold

# Testing in practice

*Let's see it in action.*

## Finders

A way to find widgets in the widget tree.

Can be used with matchers.

*find the **MaterialApp** , find the **text** “Home”,  
find the alarm **icon**, find the text **within** this tile*

# Testing in practice

*Let's see it in action.*

## Mocks

Fake external dependencies and dictate their behaviour.

Extremely powerful and flexible.

Use the **mocktail** package. 

# Testing in practice

*Let's see it in action.*

## Code Coverage

A metric of what lines of code have been **run** during tests.

VGV aims for **100%** code coverage.

# Thank you!



Jeroen Meijer (Jay)  
Senior Engineer  
@jfkdev



Experts



hello@verygood.ventures  
@VGVentures