

Test Driven Development

Why should you write tests?

- Increases code quality
- Facilitates change
- Simplifies integration
- Documentation

The problem with testing

- It's like documentation, no-one likes to write them
- How do you know what to tests?
- How do you known when there is enough tests?
- Is it really worth it?

The solution: write the tests first!

- Demo: a simple “contact us” application:
 - Contact us page with a form (name, email and question textarea)
 - User fills the form and submit
 - Form data saved in the db
 - Thank you page

TDD: Where to start?

- Write a test that simulate a user interacting with the site
- Start with the easiest
- Test one thing at a time
- Make sure the test fails
- Make the test pass

TDD: Where to start?

- `test_contactus_page_loads`

TDD: Where to stop?

- “Normal case” is tested ?
- Corner cases / error conditions ?
- You're done!
- you will have written tests for any non- trivial code (template tags, managers, ...)
- When fixing a bug, always write a test that fails first.

TDD: Benefits

- The code is easier to write / I can handle more complex programs
- It's easier to collaborate
- You know what to test and when to stop

TDD: Unresolved issues

- Keeping the tests independent of the presentation (eg: form validation)
- Testing the front-end
- Performance testing
- Testing things like “a login form exists on every page”