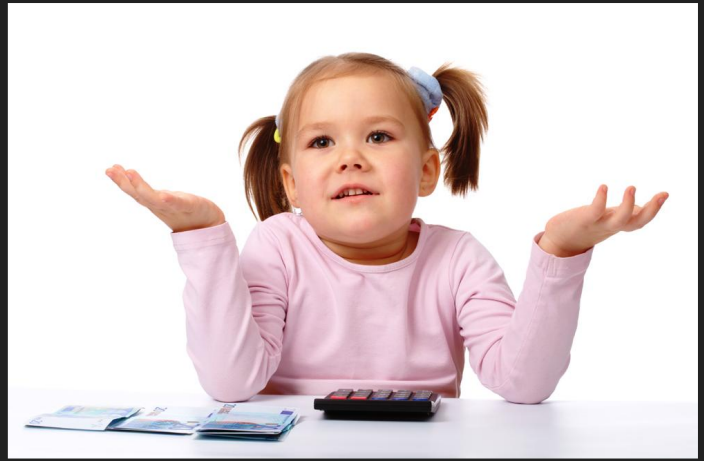
Javascript & React Testing with Jest

Bonus content: a quick refresher on basic testing concepts

Leafly

Why write tests?

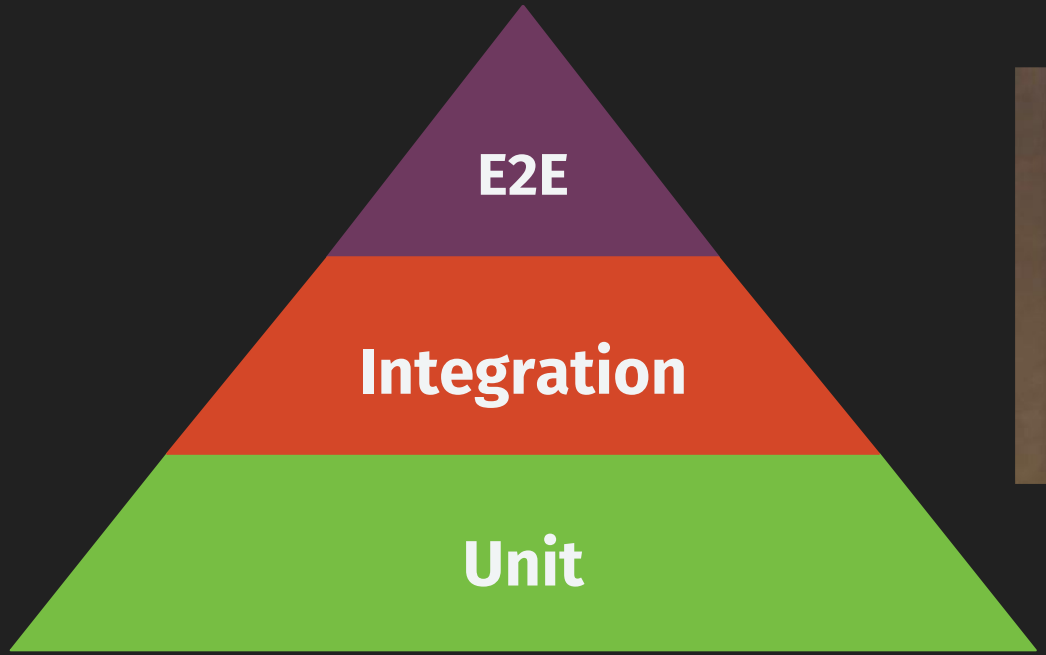
- You're more productive because you don't have to spend ~~the~~ "as much" time manually testing the code yourself.
- You can make changes to the code without worrying if something is broken (as long as the tests pass).
- Tests provide documentation for what your code is actually meant to do.



Bray, R. (2017, November 16). *Why should you write tests*. Retrieved from https://medium.com/stratajet-tech/why-should-you-write-tests-910a3175d33c

Leafly

Unleash the power of The Pyramid



Unit Tests

- Unit tests increase confident in changing/maintaining code.
- Unit tests typically force code to be more modular and reusable
- Unit tests can make development faster by not requiring an entire application to be built/loaded to validate a small change or fix
- Unit tested code is more reliable (if the tests are well written)

Integration Tests

- Improved reliability when adding to or changing existing user workflows
- Quickly add test coverage to large applications that lack testing
- More closely aligned with user workflows than unit tests
- Faster to execute and easier to maintain than end to end tests

End to End (E2E) Tests

- Ensures complete correctness and health of an application
- Increases confidence in OKR drivers
- Decreases repetitive efforts
- Most closely simulates real user interaction with an application

Assertions

Used to verify things are correct

```
expect( sum(1,2) ).toEqual( 3 )

expect( authToken ).not.toBeNull()

expect( someFunction ).toHaveBeenCalledTimes( 3 )
```

Stubs & Mocks & Spies .. OH MY!

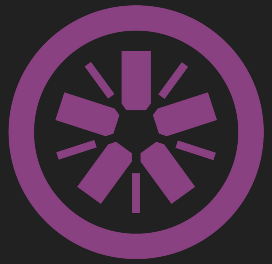
Stubs - fake functions, methods, data, or objects

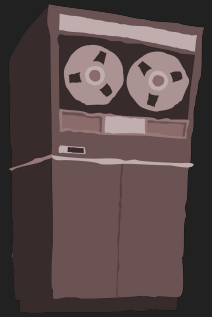
Mocks - simulated resources, often doing something beyond just existing

Spies - used to track if functions or methods are actually called

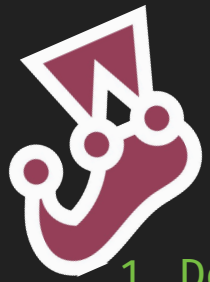
Leafly

Javascript Testing Frameworks





Leafly



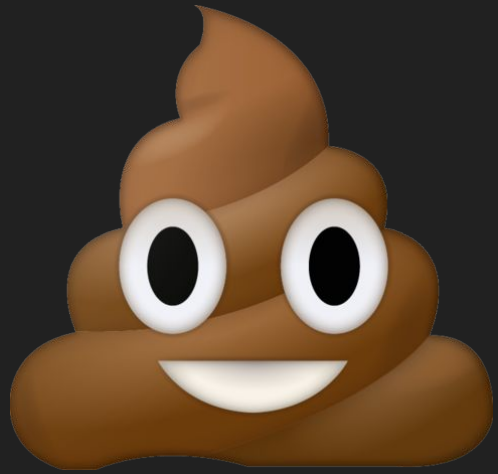
JEST

1. Designed by Facebook specifically for React testing (now OSS)
2. Minimal setup and configuration needed
3. Useful for making sure UI doesn't change unexpectedly
4. Solid solution for quickly adding some type of testing to an application with little to no testing
5. Has all the basics (assertions, mocks, stubs, spies) built in

Leafly

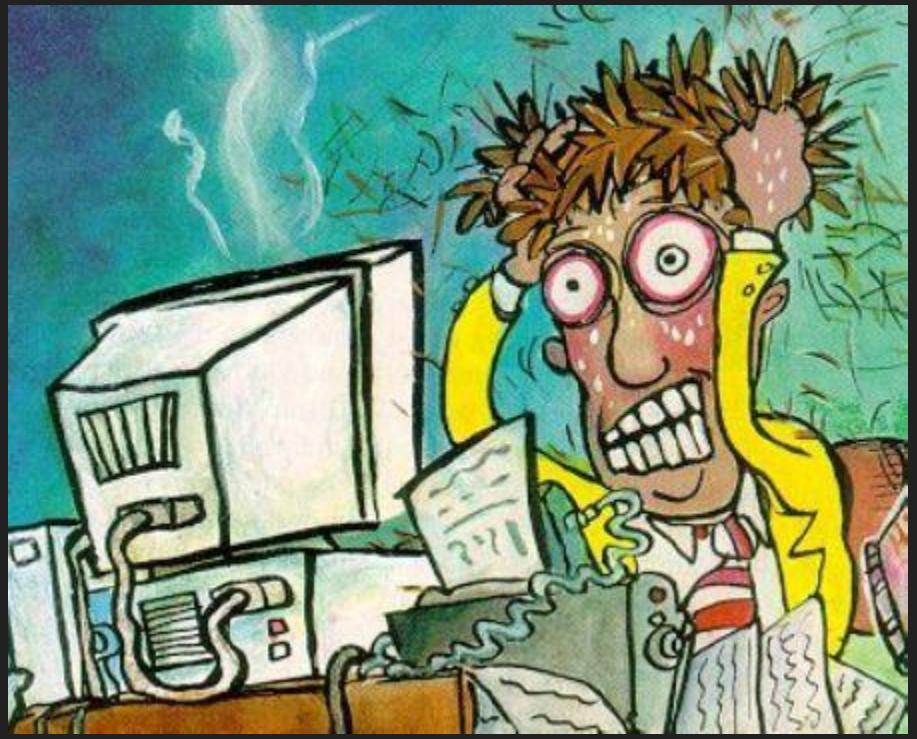
Snapshot are cool ... but may be poop

1. Good for “what”, but not for “why”
2. Very very easy to “fix” tests without really fixing anything
3. Globals ... meh
4. Snapshots are not a great reference for understanding intent



Leafly

Let’s write some code



Leafly

Some References

- Jest -Delightful JavaScript Testing
  https://jestjs.io
- Integrated Tests Are A Scam
  https://blog.thecodewhisperer.com/permalink/integrated-tests-are-a-scam
- Why should you write tests?
  https://medium.com/stratajet-tech/why-should-you-write-tests-910a3175d33c
- Low effort, high value. Integration tests in Redux apps.
  https://hackernoon.com/low-effort-high-value-integration-tests-in-redux-apps-d3a590bd9fd5
- Kent C. Dodds 👏
  https://medium.com/@kentcdodds