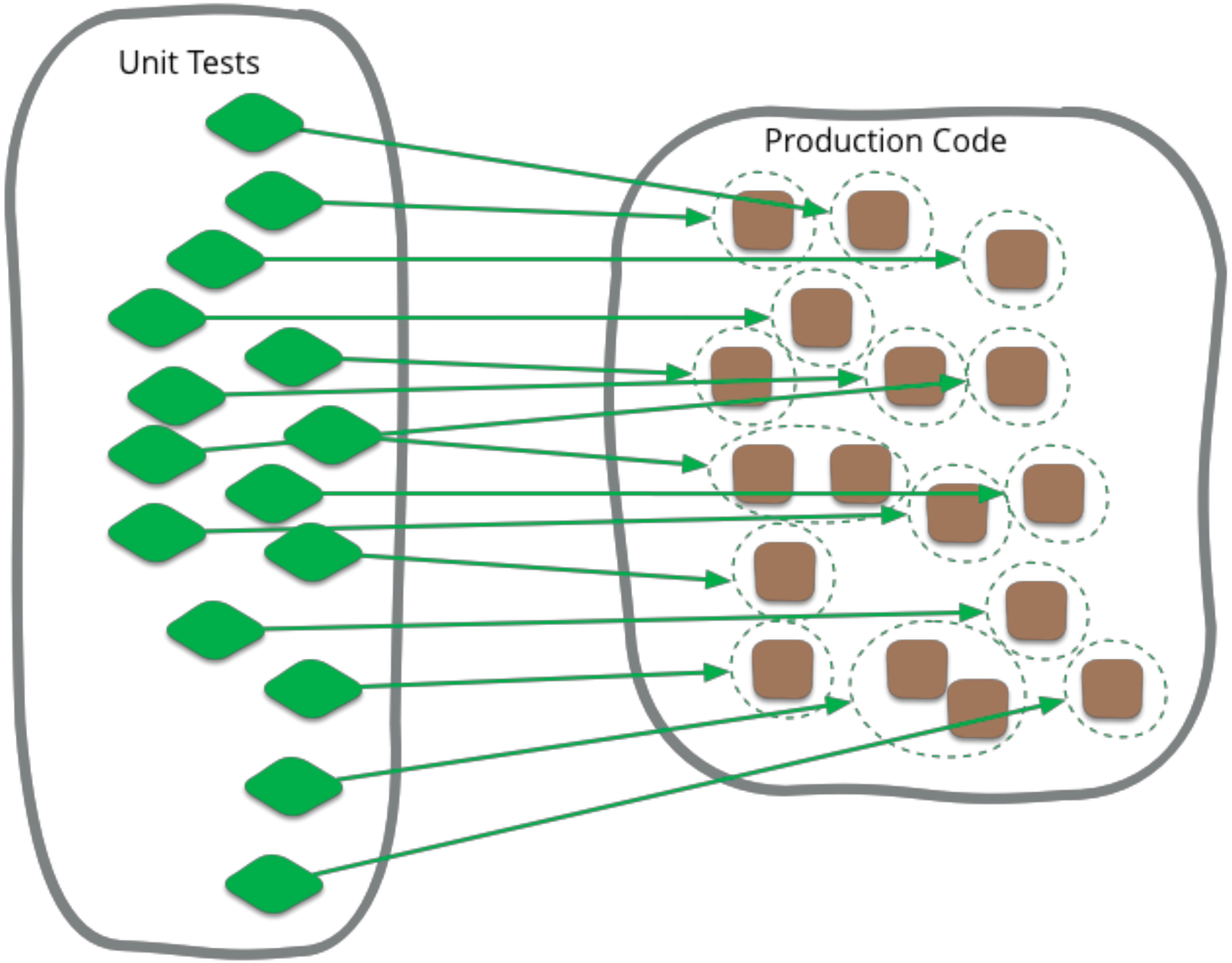
Jest, Enzyme 을 통한

리액트 컴포넌트 유닛 테스팅

Unit Testing?

소프트웨어를 기능별로,
쪼개고 쪼개서 작은 단위로
테스팅을 하는 것.



https://martinfowler.com/bliki/UnitTest.html

“홍차 끓이기” 를
유닛 테스트 한다면?

1. 물 끓이기 ✓
2. 티포트와 찻잔에 뜨거운물 부어 데우기 ✓
3. 찻잎 놓기 ✓
4. 차 우리기 ✓
5. 찻잔에 차 따르기 ✓

테스트 자동화:

테스트 작업을 사람이 직접 하는 것 이 아니라,
테스트 코드를 작성하여 모든 케이스가 성공하는지 기계가
검사하게끔 하는 것

어떠한 경우에 유용할까?

이게 왜 안되지...

(삽질 후)

이전에 수정했던 코드 때문에
쌩뚱맞게 버그가 났었네!

내가 작성 한 코드가,
다른 코드를 망가뜨리지 않았는지
바로 확인 가능!

직접 확인 할 수도 있겠지만,
프로젝트가 커지다보면 실수로
빠트릴 수 도 있기에.. 테스팅을 합니다!

언제나 좋은 것은 아니다.

소규모 프로젝트에선,
생산성을 낮출수도 있다.

나중에 프로젝트가
커질 가능성이 있다면,

시작 단계부터 해두면 나중에
시간을 많은 시간을 아낄 수 있다.

리액트 컴포넌트 테스팅

1. 특정 props 에 따라 크래쉬 없이 잘렌더링 되는지 확인
2. 이전 렌더링 결과와 일치하는지 확인
3. DOM 이벤트를 시뮬레이트하여, 원하는 변화가 제대로 발생하는지 확인
4. 렌더링된 결과물을 이미지로 저장하여 이전 이미지와 픽셀을 하나하나 비교하여 일치하는지 확인


Jest

Enzyme

리덕스 코드 테스트

통합 테스트

이상적:

1. 프리젠테이셔널 컴포넌트가 props 에 따라 제대로 렌더링되는가?
2. 액션 생성함수가 의도한대로 액션을 잘 만드는가?
3. 리듀서에 상태와 액션을 전달하면 의도한대로 제대로 업데이트 하는가?
4. 컨테이너 컴포넌트가 제대로 작동하는가?

현실적

갑자기 너무 많은걸 하려고 하면 개발 시간이 늘어남
시간자원/인력자원이 충분하지 않다면 최소한 할 수 있는 만큼만 하는게 좋음

중요한 로직만 틈틈히!

1. 프리젠테이셔널 컴포넌트 테스팅 ✓
2. 액션생성함수 ✗ (특히 FSA를 따른다면)
3. 리듀서 ✓
4. 컨테이너 ✓ (액션이 제대로 디스패치 되는지만 확인, 실제로 업데이 되는지는 확인하지 않음)