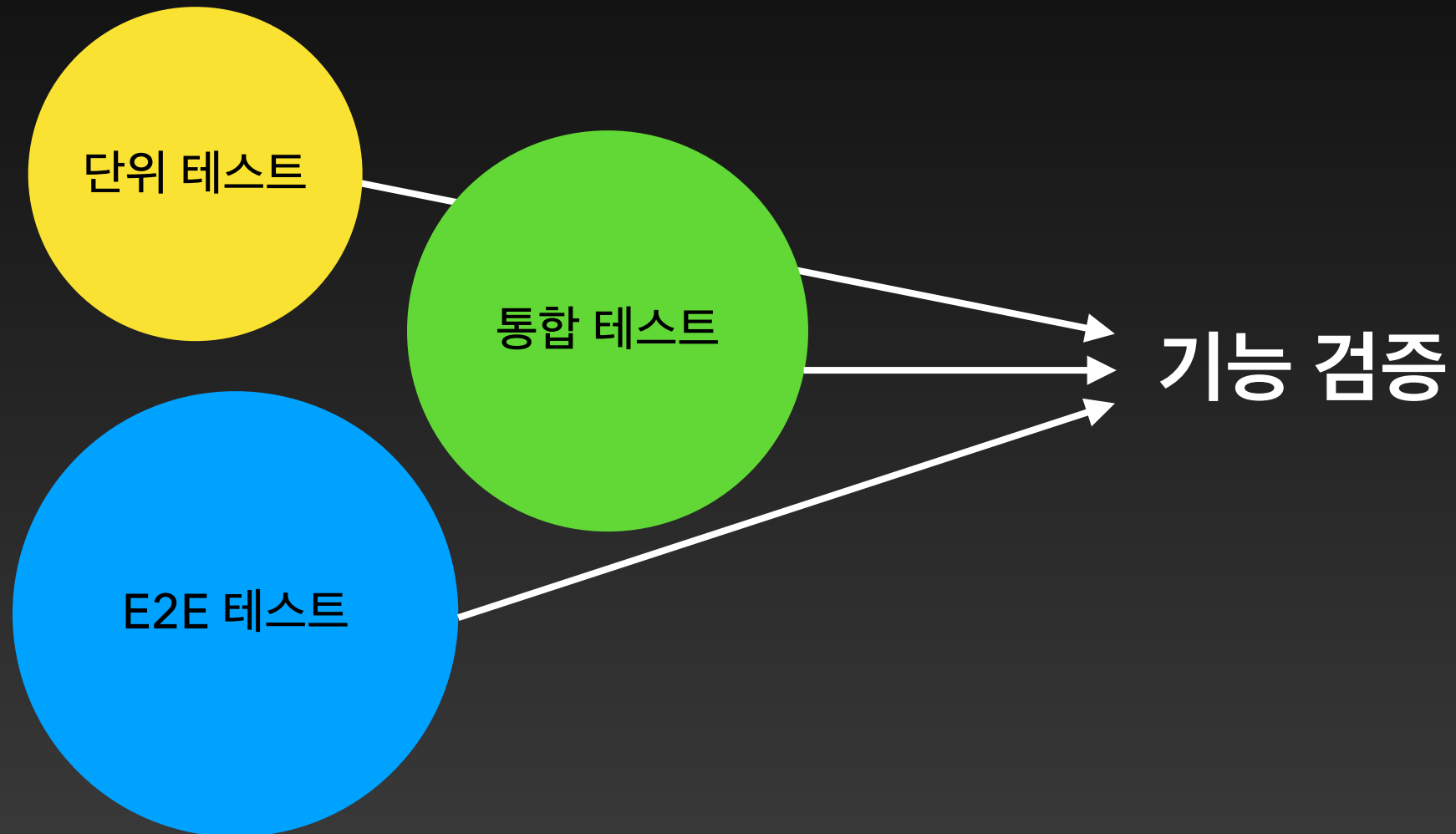


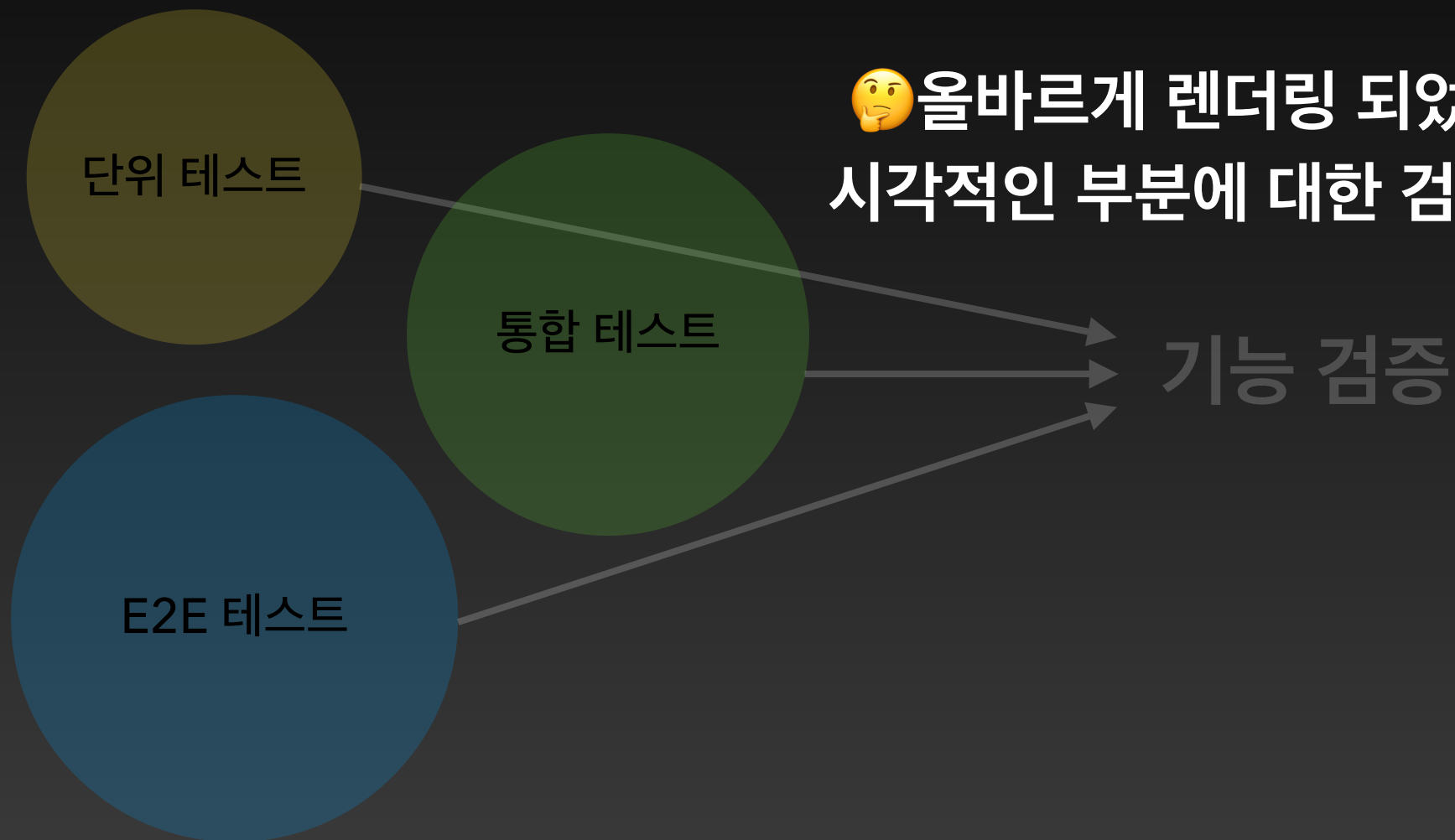
UI 테스트와 스냅샷 테스트

실무에 바로 적용하는 프론트엔드 테스트

프론트엔드 앱 검증



프론트엔드 앱 검증



🤔 올바르게 렌더링 되었는지
시각적인 부분에 대한 검증은?

스냅샷

컴포넌트의 DOM 구조를 직렬화하여 기록하거나
이미지 형태로 기록하는 것

```
const snapshots = `

#



A screenshot of a Storybook interface. The top section, labeled 'Baseline Build 1 on storybook-init', shows a single blue button. The bottom section, labeled 'New Build 4 on storybook-init', shows the same blue button but with a green highlight on its right side, indicating a visual change between the two builds. The interface includes standard icons for copy, refresh, and diff.


```

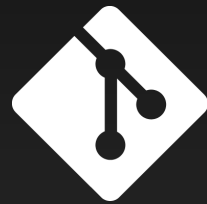
jsDOM 스냅샷 테스트

컴포넌트의 **DOM** 구조를 직렬화하여 기록하고,
이전에 기록한 스냅샷과 비교하여 검증

스냅샷 테스트 과정

1. 대상 컴포넌트를 렌더링한다.
2. 컴포넌트의 DOM을 직렬화해 스냅샷으로 기록한다.
3. 기존의 스냅샷과 새로운 스냅샷을 비교한다.
(비교할 스냅샷이 없다면 현재 스냅샷을 기록한다.)
4. 기존 스냅샷과 현재 스냅샷이 동일한 경우 테스트는 통과한다.
5. 기존 스냅샷과 현재 스냅샷이 다르다면 변경 사항을 확인한다.
 - 5-1. 의도한 변경일 경우 스냅샷을 업데이트 한다.
 - 5-2. 의도한 변경이 아닐 경우 잘못된 부분을 찾아 수정한다.

사용 브랜치



shopping-mall-snapshot-test

```
$ git checkout shopping-mall-snapshot-test
```

정리

스냅샷 테스트(jsDOM)는

- UI 컴포넌트의 렌더링 결과나 함수의 결과를 직렬화하여 기록하고 스냅샷과 비교하는 테스트
- 이를 통해 기존 UI와 어떤 부분이 변경되었는지 탐지하여 의도하지 않은 변경 사항을 빠르게 찾아 수정할 수 있다.

vitest에서 스냅샷을 테스트 할 때는

- toMatchSnapshots(): 스냅샷 기록을 별도 파일로 관리
- toMatchInlineSnapshots(): 스냅샷 기록을 테스트 파일 내에서 관리
- 정답 브랜치: shopping-mall-snapshot-test-with-answer