

# Coverage Test with Jest

소프트웨어 개발의 원리 및 실제

2019. 04. 18 (목)

## (Code) Coverage

---

- 테스트를 수행했을 때, 대상 프로그램 코드가 얼마나 잘 실행되었는지 확인
  - ✓ **Statement:** 프로그램의 실행문이 한 번 이상 실행되었는지를 확인
  - ✓ **Condition:** 조건문의 각 내부 조건이 참 / 거짓을 모두 나타낼 수 있는지 확인
  - ✓ **Decision:** 분기의 전체 결과가 참 / 거짓 모두 나타나는지 확인

- **React/Redux** 를 포함한 자바스크립트 테스트 도구
  - ✓ **Zero-configuration:** 사용자마다 개별적인 설정 없이도 사용 가능
  - ✓ **Coverage report: code coverage** 테스트 결과 제공 (**--coverage** 옵션)
  - ✓ **Mocking library: mock function/module** 생성을 위한 라이브러리 제공

# Jest Install & 실행

---

- **npm install jest**
  - ✓ <https://facebook.github.io/jest/docs/tutorial-react.html#content>
- **Module** 이 없다고 나올 시 아래 방법으로 설치
  - ✓ **npm install --save-dev** 모듈이름
- **Jest 실행**
  - ✓ **npm test**
- **Jest code coverage 실행**
  - ✓ **npm test -- --coverage**

# Jest Coverage Report

---

- **Jest** 에서 제공하는 **coverage** 테스트 기능
  - ✓ **Stmts: statement coverage**
  - ✓ **Branch: decision coverage**
  - ✓ **Funcs:** 테스트 수행 시 실행된 함수의 비율
  - ✓ **Lines: statement coverage**와 유사하나 단순히 각 라인의 실행을 확인
  - ✓ **Uncovered Lines:** 테스트에서 실행되지 않은 라인 넘버

# Tutorial Sites

---

- 공식 사이트 : <https://facebook.github.io/jest>
- 간단한 예제 : <https://github.com/facebook/jest>
- <http://airbnb.io/enzyme/index.html>
- Test Utils : <https://reactjs.org/docs/test-utils.html>
- Test Utils 예제 : <https://reactcheatsheet.com/>
- Arc에서 자동생성되는 `test.js`파일들은 좋은 예제가 될 수 있습니다.

- If ( A || B ) 라고 하면, Condition 은 A, B 각각이 T / F 가 나왔었는지를 확인 Decision 은 ( A || B ) 결과 자체가 T / F 가 나왔었는지를 확인
- 실제 Jest coverage 테스트 결과 탭에 대한 설명  
<stmts vs lines> 0: foo(); bar();  
0번 라인에 위와 같이 두 함수가 있을 때,
  - stmts는 foo, bar 각각이 실행되었는지를 측정
  - lines는 그냥 0번 라인 모두가 실행되었는지를 측정
- Mock : 소개원실 수업시간에 교수님께서 설명하셨을 것. 구체적인 구현 없이도 테스트 수행 시 타입에 맞는 임의의 입력 값에 대해 타입에 맞는 임의의 결과 값을 리턴해줌