하나의 어플리케이션 개발을 나눈다.

web 개발 파트
android 개발 파트
ios 개발

project = 관리 단위

project 를 back-end / front -end 로 나누어 처리
ㄴ sub project

본 web application개발은 root project 라고 함

Root project　　Sub project　Task
ㄴweb application 개발　- 데이터처리　- 회원관리, 게시판 등등..
　　- ui

Project　　Task
ㄴAndroid App　-회원관리, 게시판

project 생성 -> 폴더 생성

*팀 프로젝트 = 프로젝트 + 공유 저장소

bitcamp-study----------공유---------> 서버 저장소 bitcamp-study →프로젝트 = 저장소
/src　　↳ 저장소/프로젝트
/bin

bitcamp-study/ ----------공유-------> 서버 저장소 bitcamp-study →프로젝트 ≠ 저장소
java-basic　↳저장소
src/　　→프로젝트
bin/

bitcamp-study/ --------공유---------> 서버 저장소 bitcamp-study →프로젝트 ≠ 저장소
study-project/ ↳저장소
app/　　↳ Root project
src/　　→ Sub project
bin/

bitcamp-study/ = git repo.
java-basic/ = 자바 기본문법을 학습하는 용도의 프로젝트 Project
study-project/ = 자바 기본문법을 프로젝트에 적용하는 실습 프로젝트 Root project

app/ = main sub project
src/
bin/ } Sub project

*study-project 생성하기
gradle.org에서 gradle 설치

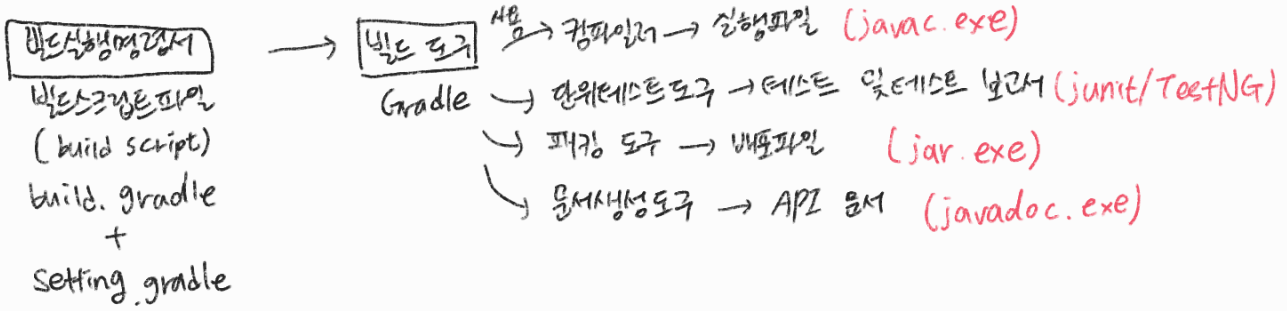
bitcamp-study/study-project$
graeld init
application 선택 프로젝트 유형
java 선택 " 언어
Groovy 선택 빌드스크립트
Junit 선택 단위테스트
study-project 를 프로젝트 이름으로 사용
com.eomcs.pms 를 패키지명으로 지정

*Gradle 빌드 도구가 생성한 프로젝트의 구조
study-project/ -> Root project 폴더
ㄴ.gradle/ ->Gradle 빌드 도구가 작업할 때 사용하는 임시폴더 (깃 저장소에 백업하지 않는다)
ㄴapp/ -> 메인 서브 프로젝트 폴더
  ㄴsrc/
    ㄴ main/
    ㄴ test/
  ㄴbuild.gradle
ㄴ gradle/ →도구를 다운로드하는 설치프로그램이 들어있다 } PC에 Gradle을 설치하지 않은 개발자를 위해 존재
ㄴ gradlew ->gradle 실행 명령 파일
ㄴ gradlew.bat ->gradle 실행 명령 파일
ㄴgitattributes ->git 명령을 실행할때 참고하는 정보
ㄴgitignore -> git 저장소에 백업하지 말아야할 대상을 지정
ㄴsettings.gradle ->gradle 명령을 실행할 때 참고하는 정보

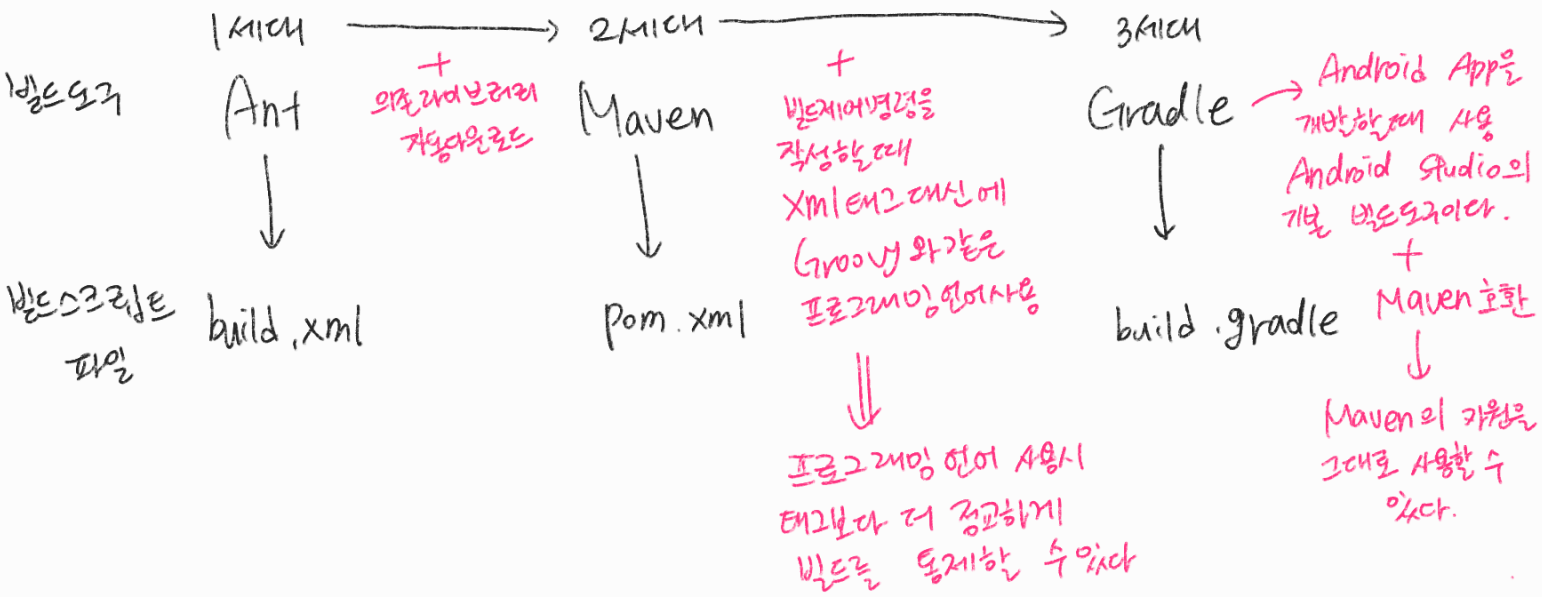
*빌드 도구
애플리케이션을 만들기 위해 수행하는 작업을 도와준다.
ex. 컴파일, 테스트, 배포파일 생성, 문서 생성 등

애플리케이션 생성과정을 통제하는 프로그램

*빌드 도구로 애플리케이션 생성을 통제
Gradle 빌드도구는 빌드 스크립트 파일에 작성되어있는대로 다른 개발도구를 사용해서 작업수행, 산출물 생성



*Build 도구의 종류



Gradle 은 plug 장착 (junit) 해서 실행하면 작업 시 편리함
빌드도구n(build폴더) 가 어떤 원리로 돌아가는지

- 교육 과정 진행
    - 빌드 도구
    - 빌드 도구 개요
    - 다양한 빌드 도구: Ant, Maven, Gradle
    - Gradle 빌드 도구를 이용하여 프로젝트 폴더 준비
        - `gradle init`
    - Gradle 빌드 도구로 생성한 디렉토리의 구조 및 파일 설명
- Gradle 빌드 도구 사용법
    - 빌드 스크립트 파일의 용도
        - settings.gradle : 여러 프로젝트에 공통으로 적용할 설정 정보.
        - app/build.gradle : 서브 프로젝트에만 적용하는 빌드 설정 정보.
    - Gradle 기본 작업
        - init : 현재 폴더를 프로젝트 폴더로 구성
        - wrapper : Gradle 설치 및 실행 파일 생성
        - gradle에 플러그인을 장착하면 더 많은 작업을 수행할 수 있다.
    - 'java' gradle 플러그인
        - compileJava
            - src/main/java 폴더에 있는 소스 파일을 모두 컴파일
            - build/classes/java/main 폴더에 .class 파일을 둔다.
        - compileTestJava
            - src/test/java 폴더에 있는 소스 파일을 모두 컴파일
            - build/classes/java/test 폴더에 .class 파일을 둔다.
        - processResources
            - src/main/resources 폴더에 있는 파일을 build/resources/main 폴더에 복사한다.
        - processTestResources
            - src/test/resources 폴더에 있는 파일을 build/resources/test 폴더에 복사한다.
        - clean

        - build 폴더를 삭제한다.
    - classes
        - compileJava와 processResources를 모두 수행
    - testClasses
        - classes + compileTestJava + processTestResources 수행
    - check
        - test + 단위 테스트 수행
    - javadoc
        - 소스 파일에서 javadoc 주석을 추출하여 HTML된 API 문서를 생성한다.
    - build
        - check + assemble(배포 파일 생성 작업) 수행
- 'application' gradle 플러그인
    - run
        - 'java' 플러그인의 classes 작업을 먼저 실행한다.
        - 그런 후 application 설정에 지정한 클래스를 실행한다.
    - build
        - 이 플러그인을 장착한 상태에서 build 작업을 수행하면 고객에게 배포할 수 있는 파일을 build/distributions 폴더에 생성한다.
        - 자바 프로그램을 실행시킬 수 있는 스크립트 파일도 자동 생성된다.

- 정리

1) init 작업을 통해 프로젝트 폴더를 준비한다.
2) build.script에 빌드 작업이 들어 있는 플러그인을 설정한다.
3) 각 플러그인의 작업을 실행할 때 필요한 정보를 등록한다.
4) 프로젝트에서 사용할 외부 라이브러리 파일을 등록한다.
5) 필요한 작업을 실행하여 애플리케이션을 빌드한다.