

# 모바일 테스트 실습

**소개 :** 안드로이드 개발자라면 모두 ADB(Android Debug Bridge)를 사용한다. 안드로이드 SDK 에 포함되어 있는 기능인데. 쉽게 말해 에뮬레이션이나 안드로이드 단말기기와의 연결고리, 도구를 의미한다. 특히나 QA(Quality Assurance)를 진행할 때 ADB를 사용하면 아주 유용하다. 이번 실습에서는 ADB 사용을 위한 환경 셋팅과 간단한 사용법을 배워본다.

## 사용 도구:

이번 실습에 사용할 툴은 안드로이드 스튜디오이다.

안드로이드 스튜디오를 다운로드 받을 때 아래와 같은 환경도 같이 설치된다.

- 안드로이드 가상머신(Android Virtual Device Manager)
- 안드로이드 SDK(Software Developer's Kit)
- ADB(Android Debug Bridge) (안드로이드 SDK 에 포함)

## 1. 도구 Setup

- ✓ 안드로이드 스튜디오 설치 및 환경설정

## 2. ADB 실습

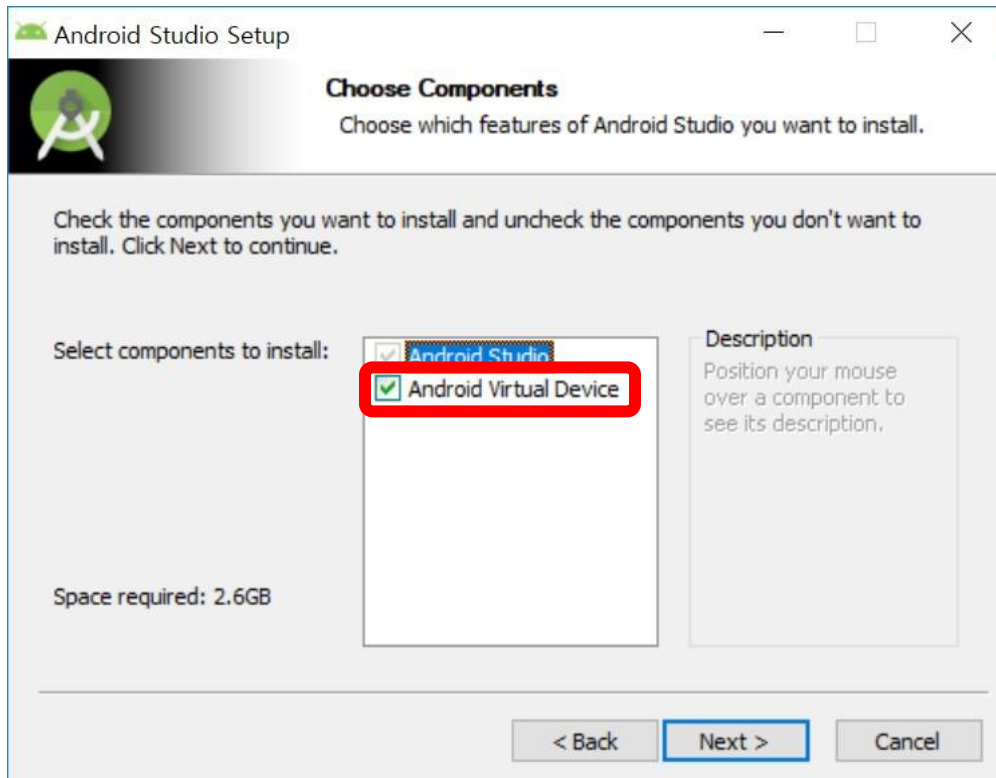
- ✓ ADB 명령어 실습
- ✓ Monkey Test 실행

## 제출 파일:

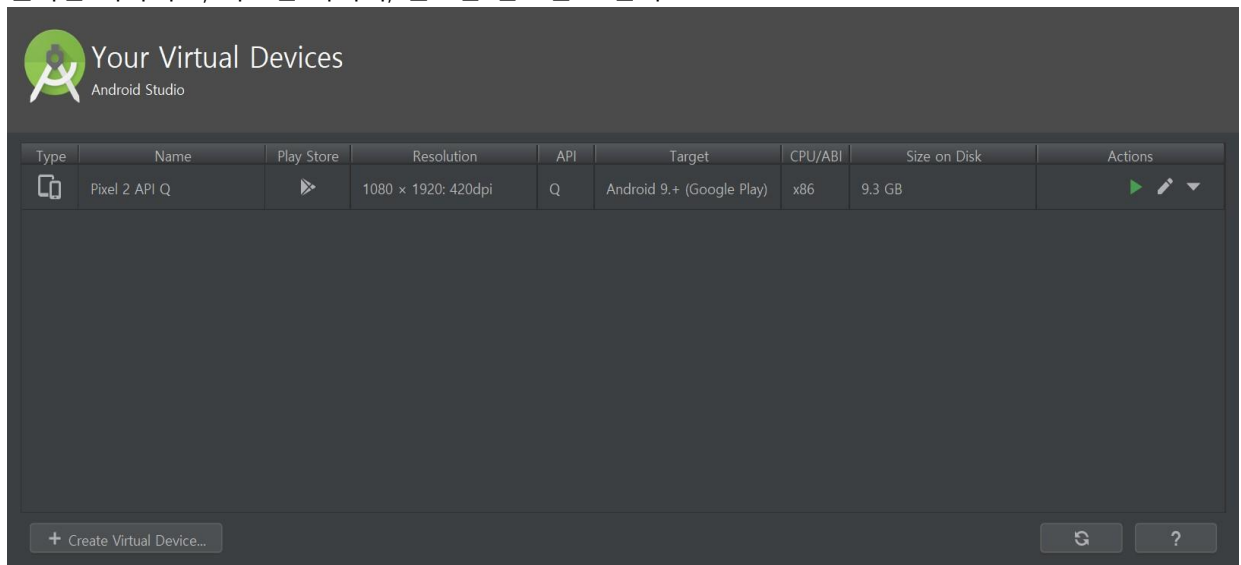
- ✓ CMD에 실습한 ADB 명령어 캡처
- ✓ CMD에 실습한 Monkey 실행 캡처 & 각 명령어의 기능 부록 참조하여 정리
- ✓ Monkey 테스트 중 영상을 녹화 후 압축하여 제출  
Monkey 테스트 실습한 자료를 참조하여 2000~5000 강도를 설정하여 녹화  
(2가지 녹화 후 압축하여 제출)
- ✓ 보고서로 제출

## 1. 도구 Setup

- 1.1. 링크에서 안드로이드 스튜디오 다운로드 (<https://developer.android.com/studio/>)
- 1.2. OS별 안드로이드 스튜디오 설치 방법 (<https://developer.android.com/studio/install>)
- 1.3. 설치 중 Android Virtual Device 선택 후 다운로드



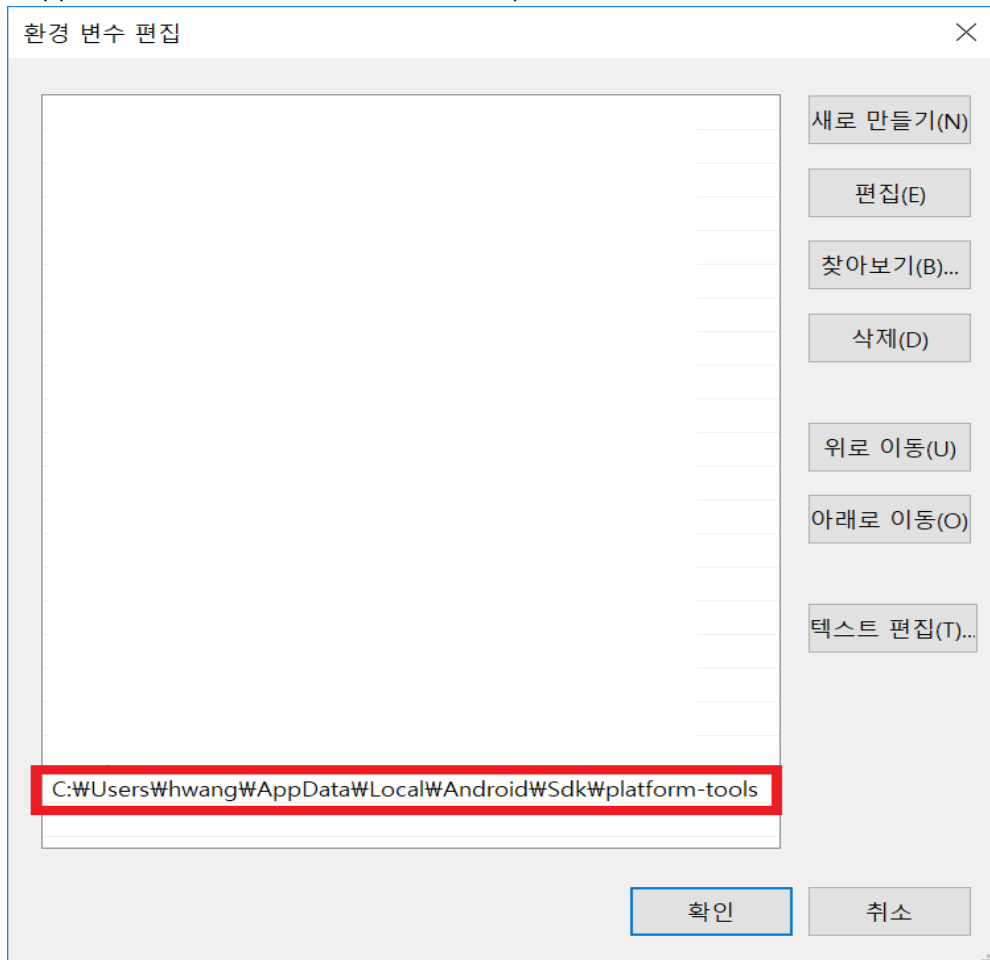
- 1.4. 빈 프로젝트로 프로젝트 생성
- 1.5. Tools -> AVD Manager -> create new virtual Device 선택
- 1.6. 원하는 디바이스, 시스템 이미지, 필요한 컴포넌트 설치



## 1.7. Action -&gt; 실행 버튼으로 실행

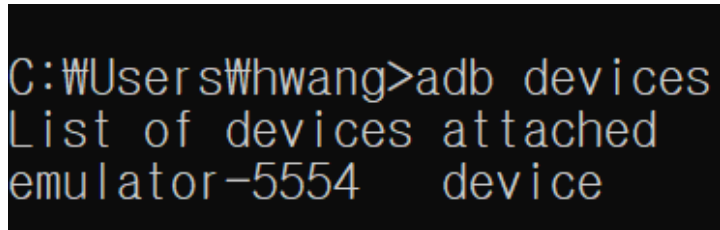


## 1.8. (AppData-&gt; Local &gt; &gt; Android &gt; Sdk &gt; platform-tools Path) 환경변수 PATH 설정



## 2. ADB 실습

2.1. "adb devices" 명령어로 가상화 기기 연결되었는지 확인



```
C:\Users\Whwang>adb devices
List of devices attached
emulator-5554    device
```

2.2. adb reboot 명령어로 단말 초기화

2.3. adb shell getprop ro.build.version.release 명령어로 버전 확인

2.4. adb shell screencap -p /sdcard/screen.png 가상화 기기에 스크린샷을 저장

2.5. adb pull /sdcard/screen.png 가상화 기기에 있는 스크린샷을 PC로 저장

2.6. adb shell rm /sdcard/screen.png 가상화 기기에 스크린샷을 삭제

2.7. adb shell screenrecord /sdcard/example.mp4 로 동영상 녹화/ Ctrl + C 로 녹화종료

2.8. adb shell input text "[입력할 텍스트]" 로 텍스트 입력 (타이핑 가능한 위치)

### 3. MonkeyTest 실습

3.1. adb -s <테스트할 device 명> shell

Ex) adb -s emulator-5554 shell

3.2. monkey -v 200

※ monkey -p <패키지 명> (패키지를 만들어서 실행)

3.3. monkey -p <패키지 명> -v 200

3.4. monkey -p <패키지 명> --pct-touch 100 -v 100

3.5. monkey -p <패키지 명> --pct-touch 70 --pct-majornav 30 -v 100

3.6. monkey -p <패키지 명> --throttle 1000 -v 100

Ex) monkey -p com.example.myapplication -v 100

3.7. Monkey 스트레스 테스트 (3가지 중 2가지만 녹화 및 캡처)

3.8.1 monkey -p <패키지 명> 5000

3.8.2 monkey -p <패키지 명> -p <패키지 명> 2000

3.8.3 monkey --pct-touch 30 -p <패키지 명> 3000

3.8. 패키지 예제

youtube :

com.google.android.youtube

google map :

com.google.android.apps.maps

clock :

com.google.android.deskclock

chrome :

com.android.chrome

## 부록. Monkey 명령어 문서

### General

--help 도움말 표시  
 -v 출력 레벨 조정 (레벨 0 - 디폴트로 시작, 종료 등의 최소한의 출력, 레벨 1 - 어떤 이벤트를 발생시켰는지 상세 표시, 레벨 2 - 레벨 1보다 더 상세한 정보 출력)

### Events

-s monkey는 무작위 이벤트를 발생시키지만, seed를 지정하면 에러를 발견하고 수정한 후 이전과 동일한 이벤트를 발생시킬 수 있다. 같은 이벤트를 발생시키고자 한다면 같은 seed 값을 지정하면 된다.  
 --throttle 이벤트 발생 속도를 지정. 지정하지 않는 경우에는 가능한 빠르게 이벤트를 발생시킨다.  
 --pct-touch 터치 이벤트 발생 비율을 조정한다.(터치 이벤트는 화면의 한 곳을 눌렀다 떼는 신호를 말한다)  
 --pct-motion 모션 이벤트의 발생 비율을 조정한다.  
     (모션 이벤트는 화면 한 지점을 누르고 무작위로 다른 지점으로 이동 후 떼는 것을 말한다)  
 --pct-trackball 트랙 볼 이벤트의 발생 비율을 조정한다.  
     (하나 이상의 무작위 이동 이벤트를 말하며 가끔 한 번의 클릭이 올 수 있다)  
 --pct-nav "basic"내비게이션 이벤트의 발생 비율을 조정한다.  
     (내비게이션 이벤트란 H/W(폰)의 상하좌우 방향키 입력을 말한다.)  
 --pct-majornav "major"내비게이션 이벤트의 발생 비율을 조정한다.  
     (백 키나 메뉴 키, 방향키의 가운데 키와 같은 이벤트를 말한다.)  
 --pct-syskeys 시스템 이벤트의 발생 비율을 조정한다.  
     (Home 버튼, 백 키, 전화 걸기, 전화 끊기, 볼륨 조절과 같은 이벤트를 말한다.)  
 --pct-appswitch 액티비티 실행 비율을 조정한다. 패키지 안의 다른 액티비티로 변환되는 것을 말한다.  
     Monkey는 startActivity()에 의한 Activity 실행도 테스트한다.  
 --pct-anyevent 그 밖의 이벤트의 비율을 조정한다.

### Constraints

-p <allowed-package-name 한다.=""" 패키지를="" 패키지="" 테스트할="" 지정한다.=""" 지정이="" 없는="" 시스템="" 모든="" 명을="" 대상으로="" 경우에는="" -p 와 같이 복수의 패키지 지정도 가능하다. 테스트할 패키지 내에서 다른 패키지의 액티비티를 사용하고 있다면 그 패키지를 지정한다.  
 -c 카테고리를 지정한다. 지정하지 않으면 Intent.CATEGORY\_LAUNCHER 또는 Intent.CATEGORY\_MONKEY를 지정한 것으로 된다. 복수의 카테고리를 지정할 때에는 -c 옵션을 카테고리마다 지정한다. </allowed-package-name >

### Debugging

--dbg-no-events 액티비티의 실행은 하지만 이벤트를 발생시키지는 않는다.  
 -v 커맨드와 함께 사용하고 최상의 결과를 얻기 위해 -v 의 조합과 함께 하나 이상의 패키지 제약과 non-zero throttle 로 30 초 이상 monkey 실행을 유지한다.  
 --hprof 프로파일 리포트를 작성한다. 리포트 파일은 data/misc 디렉터리에 작성되고 파일 사이즈는 5M

이상이 되므로 주의한다. Traceview 어플리케이션 도큐먼트에 상세하게 설명되어 있다.

--ignore-crashes 지정을 하지 않는 경우에 Monkey는 어플리케이션의 크래시나 익셉션이 발생하면 정지된다. 이 옵션을 지정하면 이벤트 개수만큼 계속 이벤트를 보내게 된다.

--ignore-timeouts 지정을 하지 않은 경우에 Monkey는 타임아웃 에러가 발생하면 정지한다. 타임아웃 에러는 Application Not Responding 다이얼로그와 같은 것이 나올 때는 말한다. 옵션을 지정하면 에러 후에도 이벤트를 계속 발생시킨다.

--ignore-security-exceptions 지정하지 않은 경우엔 Monkey는 시큐리티 익셉션(퍼미션 에러)가 발생하면 정지한다. 이 옵션을 지정하면 에러 후에도 이벤트를 계속 발생시킨다.

--kill-process-after-error 통상 에러에 의해 Monkey 가 정지될 때에는 테스트 된 어플리케이션은 fail 이 나지만 프로세스는 돌고 있다. 이 옵션을 지정하면 시스템에서 프로세스로 종료 시그널(kill)을 보낸다.

--monitor-native-crashes Android의 시스템 네이티브 코드에서 일어나는 크래시를 리포트한다. --kill-process-after-error 옵션을 함께 사용하면 시스템은 정지한다.

--wait-dbg 디버거가 attached 할 때까지 Monkey 실행을 정지한다.