**frida script 기본설명(1) : frida API**

**Java.perform(fn)**

Java.perform(function() {

});

: 현재 스레드가 가상머신에 연결되어 있는지 확인하고 fn을 호출

**Java.use(className)**

// 앱에서 사용하는 클래스와 연동되는 myClass를 정의한다.

var myClass = Java.use(com.mypackage.name.class);

// myClass를 통해 객체 인스턴스 생성 및 정의를 한다.

var myClassInstance = myclass.$new();

// 클래스 내부에 있는 메소드에 접근해 인자 값을 넘겨주고 해당 결과 값을 result에 받는다.

var result = myClassInstance.myMethod(“param”);

// .implementation : 앱에서 정의된 메소드의 구현 내용을 재작성 한다.

myClass.myMethod.implementation = function(param) {

};

: 자바스크립트 래퍼(wrapper)를 className에 동적으로 가져와서 생성자를 호출하기 위해 $new()를 호출하여 객체를 인스턴스화 함

// 래퍼에 대한 설명

간단히 말하면, 기본 자료형(int, long) 같은 데이터를 객체에 넣기 위해 제공하는 함수들이라고 생각하면 된다.

ex) int num = 5 --> Integer num = new Integer(5)

* 이를 이용하여 메소드 구현을 변경할 수 있고, 예외를 적용할 수 있음
* 이는 변수와 메소드에 액세스 할 수 있는 클랙스 객체를 반환(이 클래스의 인스턴스가 아니기에 구현을 변경하려면 새로운 메소드로 덮어써야 함)

**Frida script 기본설명 (2) : frida API**

**Java.choose(className, callbacks)**

Java.perform(function() {

Java.choose(com.mypackage.name.class, )}

});