

# 1장. 자바(Java) 개발 환경 및 기본문법



*JDK & Eclipse*



# 프로그래밍과 자바

## ● 프로그래밍이란?

- 컴퓨터 프로그램을 만드는 일
- 컴퓨터에게 일을 하도록 명령어를 만드는 것

만약 사람이 원하는 바를 대략 말하면 컴퓨터가 알아서 프로그램을 만들어 준다면 좋은 것이다. (현재, 인공지능과 빅데이터 기술로는 아직 먼 미래의 일)

## ● 프로그래밍 언어의 종류

- C언어, C++, C#, Java, Python, Javascript

## ● 자바(Java)

- 제임스 고슬링 등이 오크(Oak)언어에서 가전제품에 탑재할 용도로 만들어 냄
- 인터넷에 적합하고, 운영체제(플랫폼)에 독립적으로 변경함
- 썬 마이크로시스템즈, 1996년 1월 발표(지금, 오라클에 인수됨)



# 자바란?

## ● 자바 언어의 특징

- 운영 체제에 독립적이다. – JVM(자바가상머신)이 가능하게 함
- 객체지향 언어이다. – 유지보수가 쉽고, 확장성이 좋다.
- 풍부한 기능이 제공되는 오픈 소스이다.
- 네트워크와 멀티 쓰레드를 지원하는 다양한 API(라이브러리)
- 안드로이드용 스마트폰 App(앱) 개발 언어로 사용되고 있다.



# 자바(Java)로 개발한 프로그램

## ▪ 웹 사이트(서버)

- 웹 사이트를 운영하려면 반드시 서버(server)가 필요하다.
- 검색 사이트, 쇼핑몰, 금융 사이트 등 자바로 개발한 웹 서버 프로그램으로 운영

## • 안드로이드 앱

- 안드로이드 폰에서 사용하는 앱을 만들 수 있다.

## • 게임

- 게임을 만들때는 C++, C를 주로 사용하지만 마인크래프트처럼 게임을 구현하는데도 사용된다.



# 자바 가상 머신(JVM)

## ◆ JVM(Java Virtual Machine)

- 자바 프로그램 실행 환경을 만들어 주는 소프트웨어
- 자바 코드를 컴파일한 .class(바이트 코드)는 JVM 환경에서 실행됨
- 컴퓨터의 운영체제에 맞는 자바 실행 환경(JRE)가 설치되어 있다면 자바 가상머신이 설치되어 있는 것이다. (JRE > JVM)

## ◆ JDK와 JRE

**JDK**(Java Development Kit) – 자바 개발을 위해 설치하는 라이브러리이다.

**JRE**(Java Runtime Environment) – 자바 프로그램이 실행되는 자바실행환경이다.

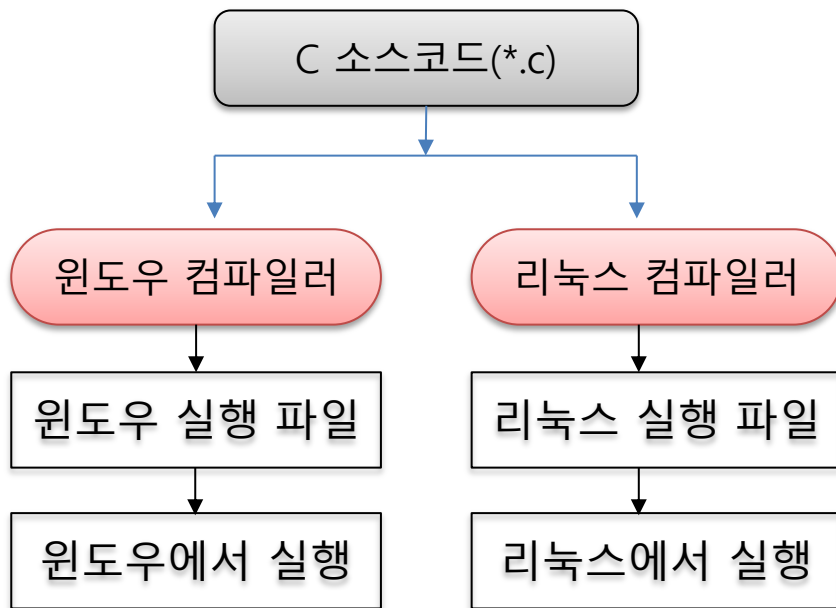
### ▶ 컴파일(Compile)과 컴파일러

컴파일은 프로그램(코드)를 컴퓨터가 알 수 있는 언어(기계어)로 바꿔 주는 일  
컴파일러는 프로그램 언어를 기계어로 번역해 주는 프로그램으로 자바(JDK)를 설치하면 자바 컴파일러도 설치

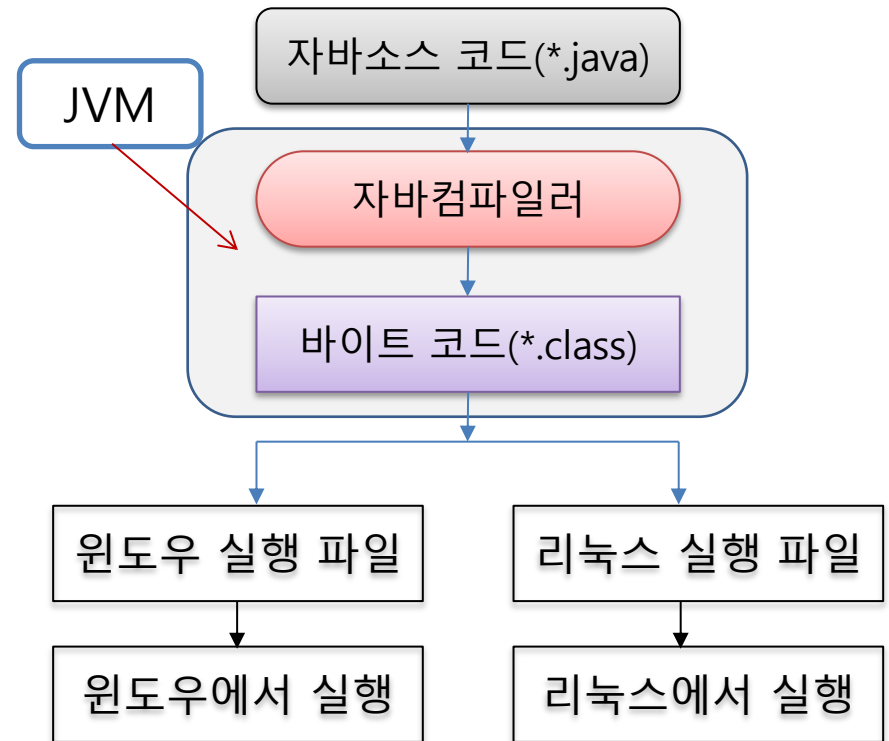


# 자바(Java) 언어

## ◆ JVM의 기능(역할)



구조적 언어-C언어



객체 지향 언어- Java, Python



# 자바 개발 환경 구축

## ◆ 자바 개발도구(JDK) 설치

- jdk 다운로드(검색)-> windows> Java SE11 다운로드 -> x64 인스톨러

**Java SE Development Kit 11.0.15.1**

Java SE subscribers will receive JDK 11 updates until at least **September of 2026**.

These downloads can be used for development, personal use, or to run Oracle licensed products. Use for other purposes, including production or commercial use, requires a Java SE subscriber license.

JDK 11 software is licensed under the [Oracle Technology Network License Agreement for Oracle Java SE](#).

JDK 11.0.15.1 [checksum](#)

**Linux**   **macOS**   **Solaris**   **Windows**

Product/file description	File size	Download
x64 Installer	140.41 MB	<a href="#">jdk-11.0.15.1_windows-x64_bin.exe</a>
x64 Compressed Archive	158.1 MB	<a href="#">jdk-11.0.15.1_windows-x64_bin.zip</a>

```
C:\Users\김기용>java -version
java version "17.0.1" 2021-10-19 LTS
Java(TM) SE Runtime Environment (build 17.0.1+12-LTS-39)
Java HotSpot(TM) 64-Bit Server VM (build 17.0.1+12-LTS-39,
```



# 자바 개발 환경 구축

## ◆ 자바 Documentation 설치

Java api 를 설명하고 있는 문서

JDK 11.0.15.1 [checksum](#)

**Linux**   **macOS**   **Solaris**   **Windows**

---

Product/file description

---

x64 Installer

---

x64 Compressed Archive

---

**Documentation Download**

### Java® Platform, Standard Edition & Java Development Kit Version 11 API Specification

This document is divided into two sections:

#### Java SE

The Java Platform, Standard Edition (Java SE) APIs define the core Java platform.

#### JDK

The Java Development Kit (JDK) APIs are specific to the JDK and will not necessarily be part of the Java SE APIs whose names start with `jdk`.

All Modules	Java SE	JDK	Other Modules
Module	Description		
<code>java.base</code>	Defines the foundational APIs of the Java SE Platform.		
<code>java.compiler</code>	Defines the Language Model, Annotation Processing, and Java Compiler APIs.		
<code>java.datatransfer</code>	Defines the API for transferring data between and within applications.		
<code>java.desktop</code>	Defines the AWT and Swing user interface toolkits, plus APIs for accessibility.		

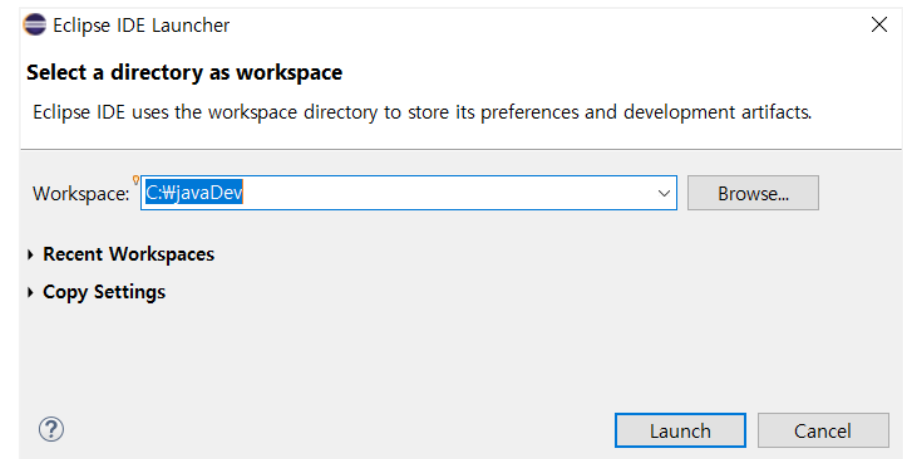
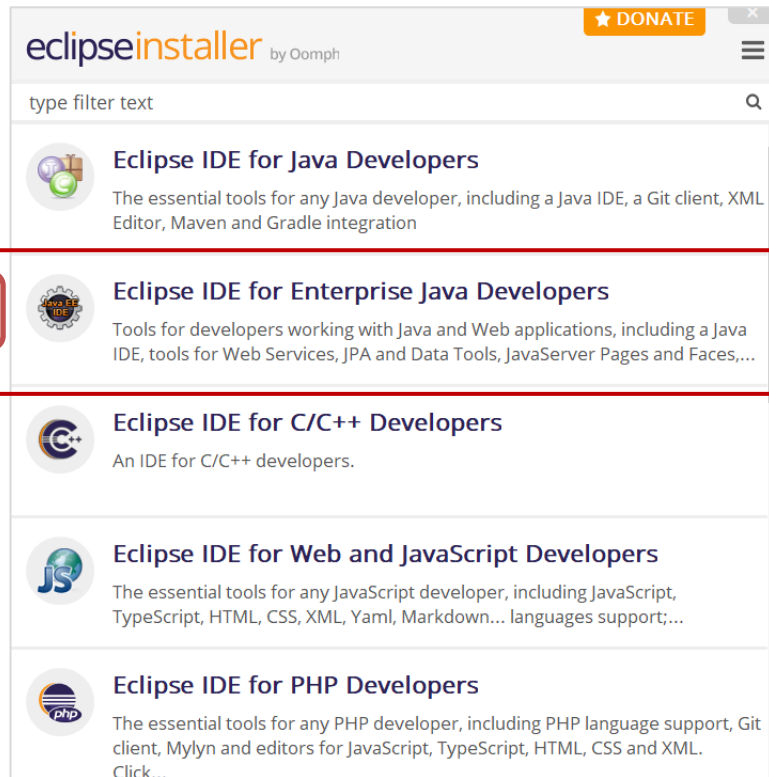




# 이클립스(Eclipse) IDE 설치

## ◆ 이클립스 IDE(통합개발환경) 설치

- 검색 \_ 이클립스(<https://www.eclipse.org/downloads/>)
- 버전 - Eclipse IDE 2022-03



Workspace – C:\Wjavaworks

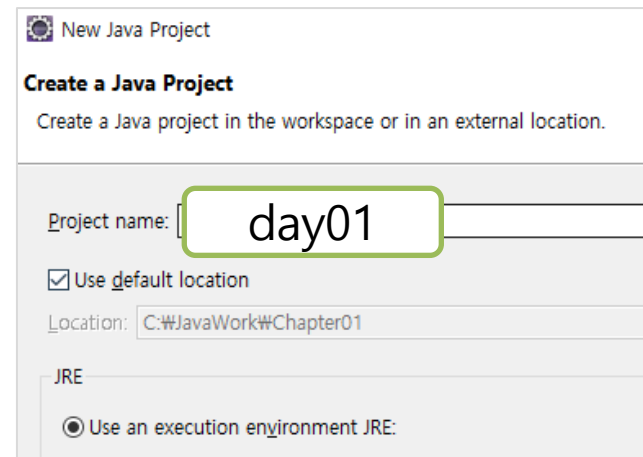
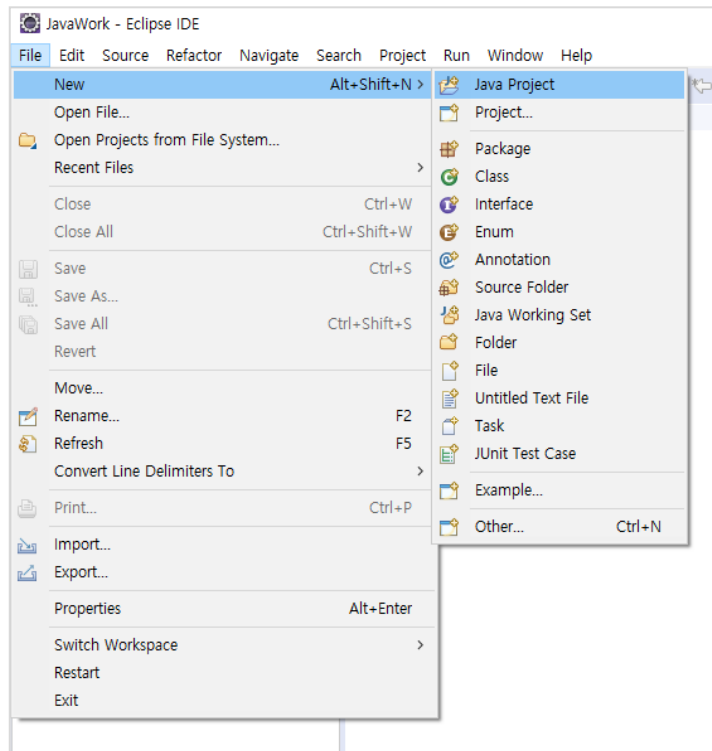


# 프로젝트 만들기

- 첫 자바 프로젝트(Project) 만들기

File->New->Java Project

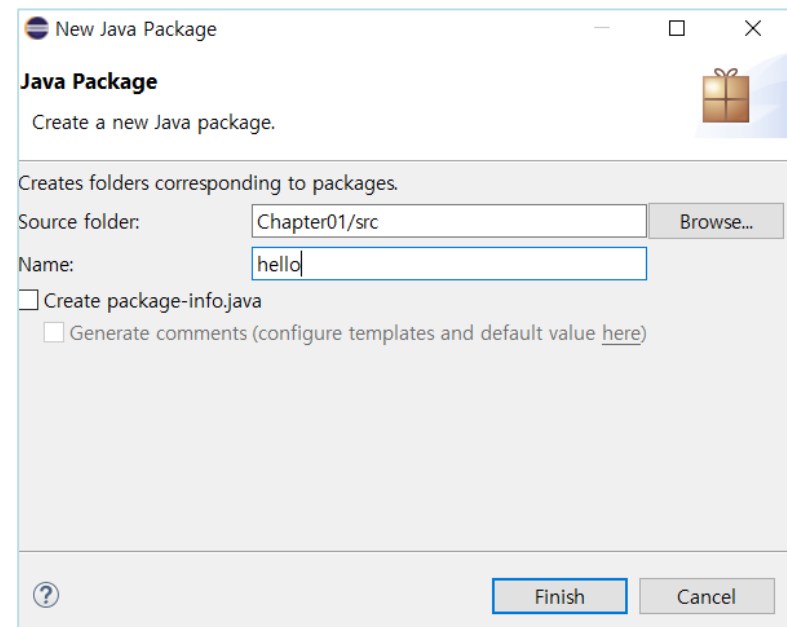
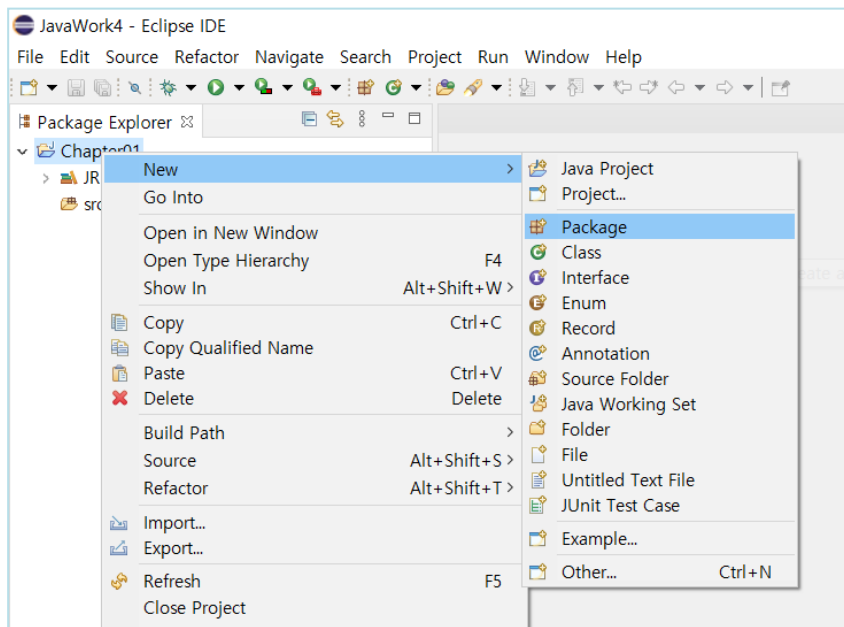
Project Name : day01



# 첫번째 패키지 만들기

## 첫번째 패키지 만들기

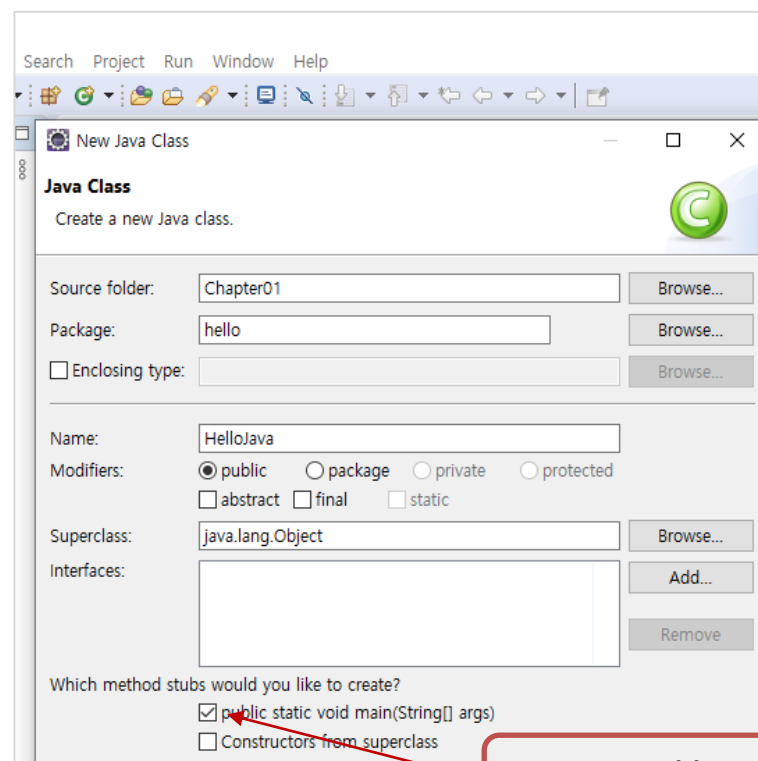
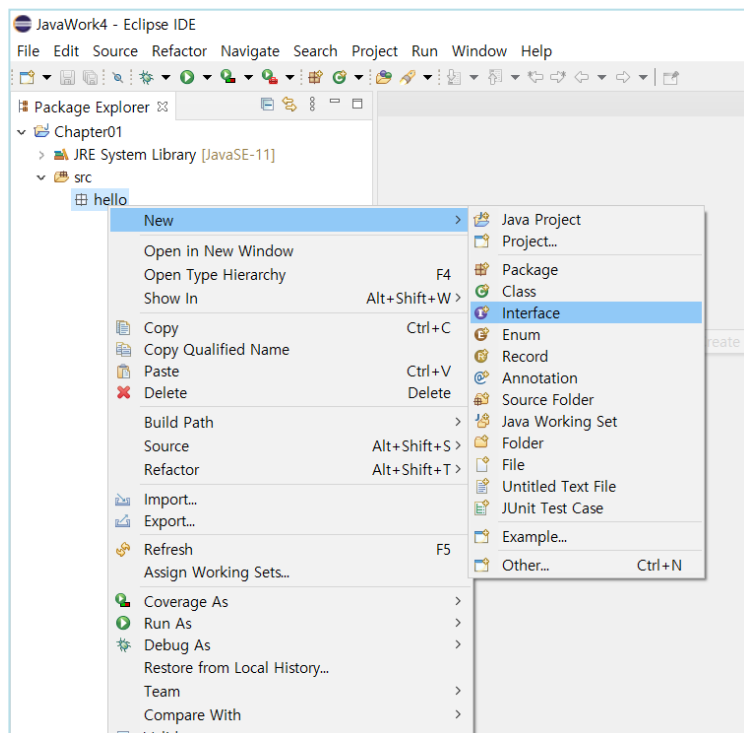
패키지 만들기 : day01(마우스우측) -> New package -> Name : hello



# 첫번째 클래스 만들기

## 첫번째 클래스 만들기

클래스 만들기 : hello(우측)->New class->Name : HelloJava



main()함수 체크



# 첫번째 클래스 만들기

## 1. java 코드 작성

- 파일 이름 : HelloJava.java
- 클래스 : System , 메서드 : main(), print()
- 파일 실행하려면 main() 메서드가 필요함

```
package hello;
```

패키지 이름

```
public class HelloJava{  
    public static void main(String[] args) {  
        System.out.println("Hello~ Java");  
    }  
}
```

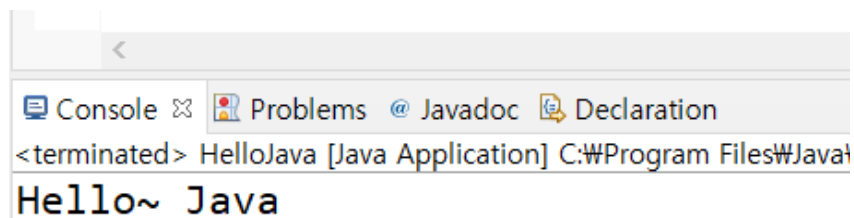
main()메서드



# 첫번째 클래스 만들기

## 2. 컴파일 및 실행

- 컴파일 하기 : 빌드 자동화 옵션 지정 -> 클래스 파일 생성
- 실행 : Run -> Run as -> Java Application [실행 단추(▶) 클릭]
- 실행 결과 : 콘솔(Console)



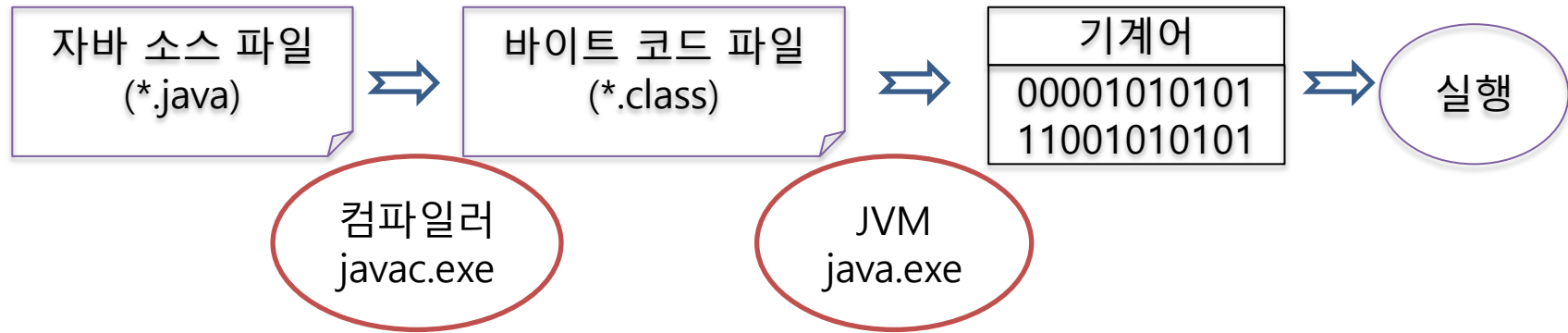
※ 클래스(.class) 파일의 위치는 어디일까?

내 PC > 로컬 디스크 (C:) > JavaWork4 > Chapter01 > bin > hello					hello 검색	
	이름	수정한 날짜	유형	크기		
	HelloJava.class	2020-12-10 오전 8:59	CLASS 파일	1KB		
	PrintData.class	2020-12-10 오전 9:00	CLASS 파일	1KB		



# 컴파일과 빌드

## 컴파일(compile)



## 빌드(build)

컴파일과 링크된 코드들을 실행가능한 파일로 만드는 일련의 과정  
전처리, 컴파일, 패키징, 배포등이 포함된다.

Java 빌드 툴로는 Ant, Maven, Gradle 등이 있다.



# 주석 , 블록, 세미콜론(;)

## 기초 문법

- 주석은 소스코드에 설명을 추가하거나 특정 코드가 컴파일되지 않도록 처리할때 사용
- 한줄 주석 : 문장 앞에 '//' 표시
- 여러 줄 주석 : /\*~ \*/ 기호 사용
- 문장이 종료는 세미콜론(;)을 사용
- { } 블록 안에 코드 작성





# 데이터(data) 출력하기

```
package hello;

public class PrintData {

    public static void main(String[] args) {
        //숫자
        System.out.println(100);
        System.out.println(3.3);
        System.out.println("-----");

        //문자
        System.out.println('A');           //문자
        System.out.println('가');
        System.out.println("apple");       //문자열
        System.out.println("-----");

        //연산
        System.out.println(4 + 5);
        System.out.println(4 + "5");

        //불리언
        System.out.println(true);
        System.out.println(5 < 4);
    }
}
```



# System 클래스

Module `java.base`

Package `java.lang`

## Class `System`

`java.lang.Object`  
`java.lang.System`

```
public final class System
extends Object
```

The `System` class contains several useful class fields and methods. It provides access to system resources, output streams; access to externally defined properties and environment variables.

Since:

1.0

`java.base` > `java.lang` > `System`

<code>void</code>	<code>print(Object obj)</code>
<code>void</code>	<code>print(String s)</code>
<code>PrintStream</code>	<code>printf(String format, Object... args)</code>
<code>PrintStream</code>	<code>printf(Locale l, String format, Object... args)</code>
<code>void</code>	<code>println()</code>
<code>void</code>	<code>println(boolean x)</code>
<code>void</code>	<code>println(char x)</code>



# 연습 문제

실습 문제 : Java 개발 환경 구축

-----  
( ) 안에 들어갈 적당한 말을 맞춰보세요.  
-----

1. 프로그램(코드)을 기계가 이해할 수 있는 언어로 바꾸는 일을 ( )이라고 한다.
2. 자바로 만든 프로그램은 ( )이 설치되어 있으면 운영체제와 상관없이 실행할 수 있다.
3. 자바 개발을 위해 설치하는 자바 라이브러리를 ( ) 라고 한다.

1.컴파일 2.JVM(자바가상머신) 3.JDK

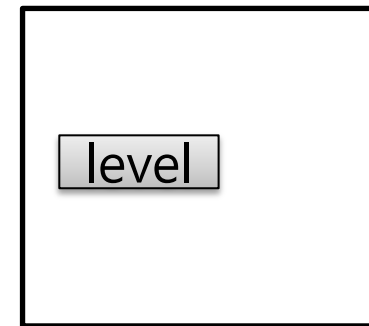


## ■ 변수란?

- 프로그램에서 사용되는 자료를 저장하기 위한 공간
- 할당받은 메모리의 주소 대신 부르는 이름
- 프로그램 실행 중에 값 변경 가능, variable 이라 함

## ■ 변수의 선언 및 초기화

- 변수 선언은 어떤 타입의 데이터를 저장할 것인지 그리고 변수이름은 무엇인지를 결정한다.
- 자료형 변수이름;
- 자료형 변수이름 = 초기값;  
`int level;`  
`double height;`



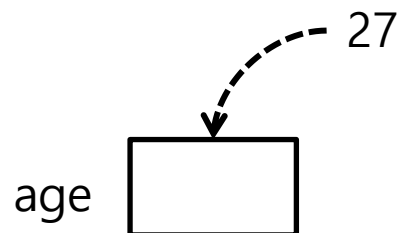
# 변수 사용하기

## ■ 변수의 초기화

```
int age = 27;
```

```
char c = 'k';
```

```
String fruit= "사과"
```



## ■ 변수 이름 선언시 유의점

- 변수의 이름은 알파벳, 숫자, `_`, `$`로 구성된다.
- 대소문자를 구분한다.
- **숫자로 시작할 수 없고**, 키워드(예약어)도 변수의 이름으로 사용할 수 없다.
- 이름 사이에 공백이 있을 수 없다.
- 변수의 이름을 정할 때는 변수의 역할에 어울리는, 의미있는 이름을 지어야 한다.

**예약어(reserved word)**  
프로그래밍 구문에 사용되는 명령어

break, int, const, if, for, class, this 등



# 변수 사용하기

## 실습 예제

- 파일 이름 : Variable.java

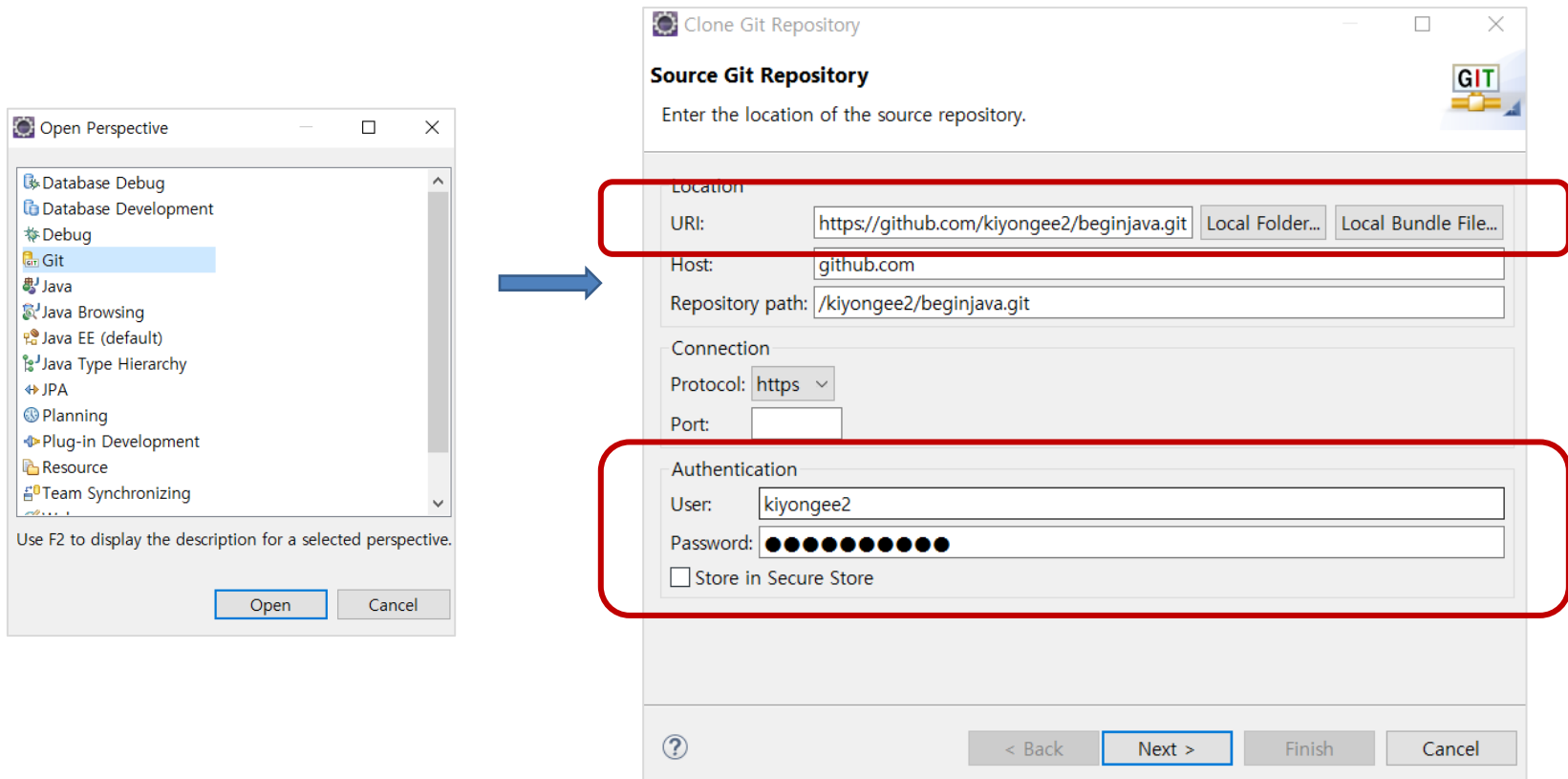
```
public class Variable {  
    public static void main(String[] args) {  
        String name;  
        name = "한지수";  
  
        int grade;  
        grade = 2;  
  
        //int class = 3; //class는 예약어라 오류  
        int schoolClass = 3;  
  
        System.out.println(name + "는 " + grade + "학년 " + schoolClass + "반 입니다.");  
    }  
}
```



# Github 연동

**eclipse에서 깃허브 사용하기**

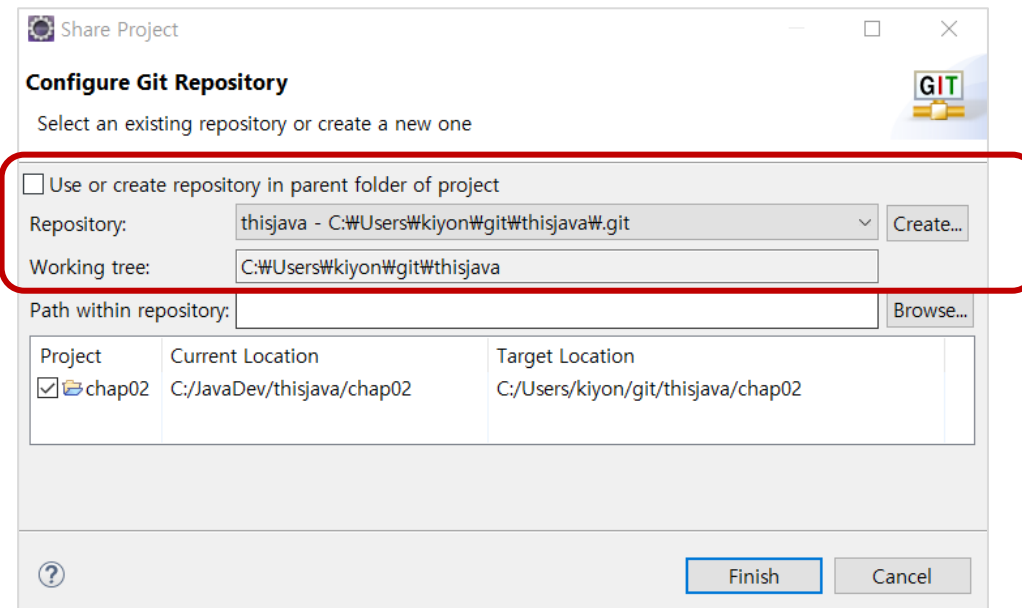
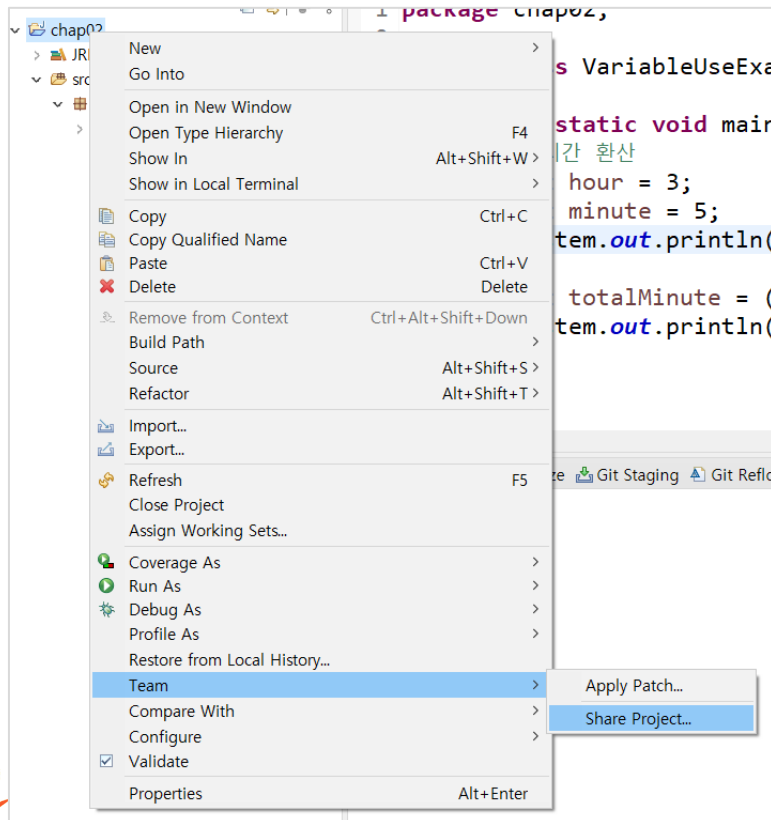
**open perspective > Git > Clone Git Repository > uri(깃저장소)**



# Github 연동

eclipse에서 깃허브 사용하기

project > 우측 > Team > Share Project > Repository(지정)

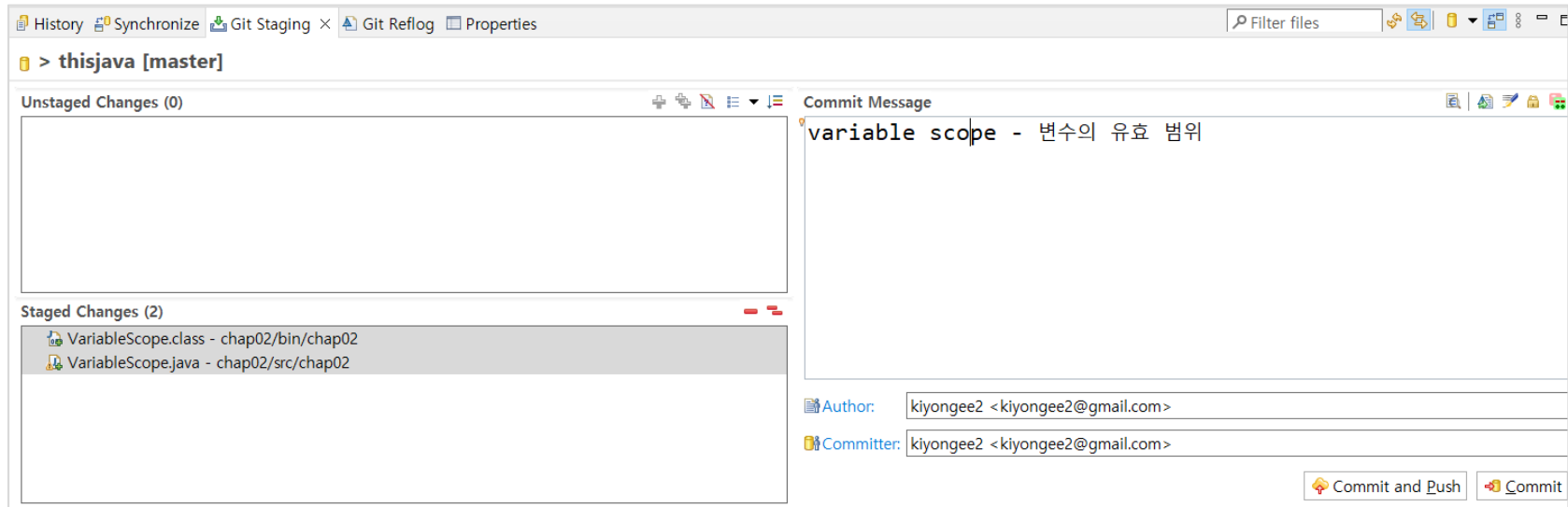




# Github 연동

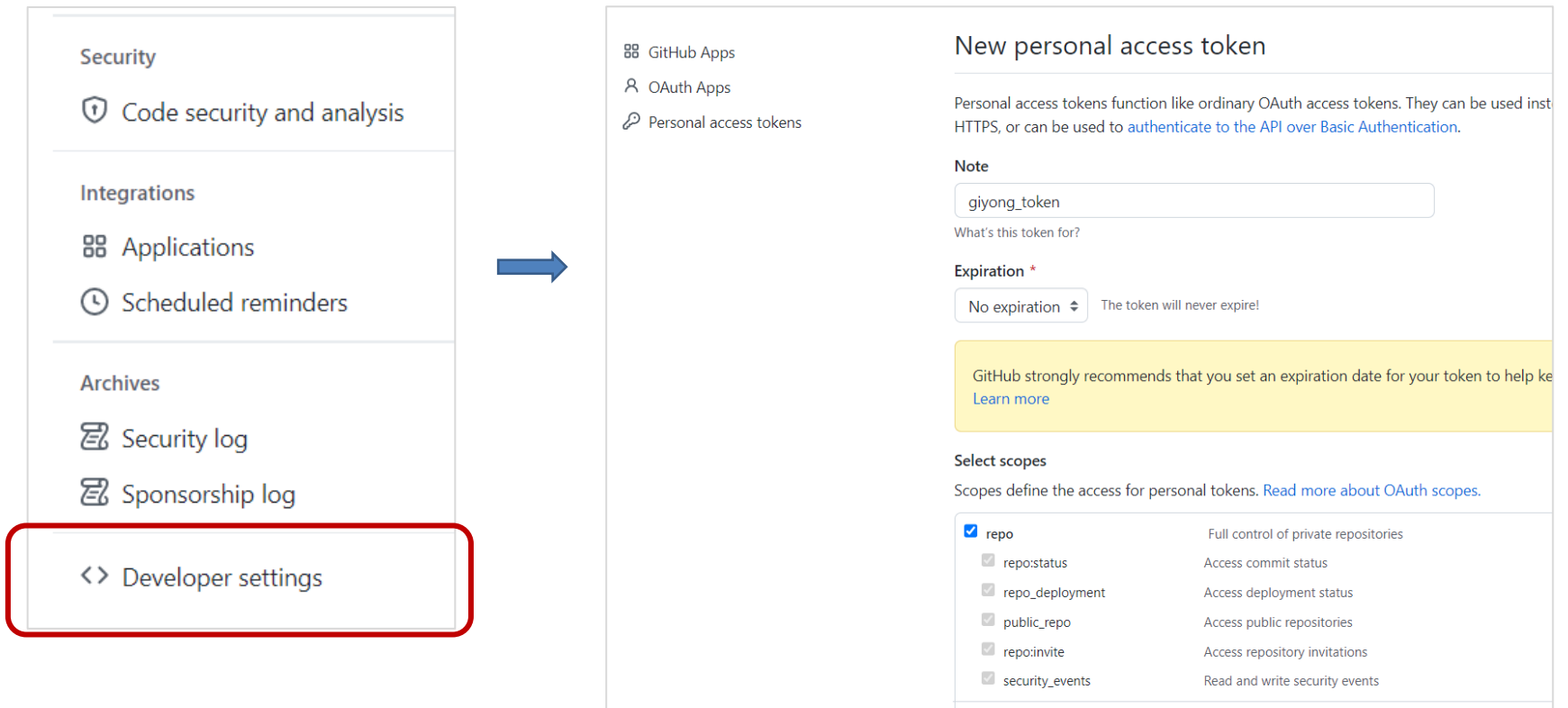
## eclipse에서 깃허브 사용하기

Git Staging > ++ 클릭 > Staged Changes > Commit Message >  
Commit > Push



password 인증이 안될 경우 – 토큰(token) 발행

Github > Settings > Developer settings > Personal access token



The image shows a two-step process for navigating to the 'Personal access token' page in GitHub. On the left, a sidebar menu is shown with categories: Security, Integrations, and Archives. The 'Developer settings' link is highlighted with a red rectangle. A blue arrow points from this link to the right-hand screenshot. The right-hand screenshot shows the 'New personal access token' page. It includes a text input for the token name (filled with 'giyong\_token'), a dropdown for expiration (set to 'No expiration'), and a list of scopes. The 'repo' scope is selected, and several sub-scopes are also checked.

**Security**

- Code security and analysis

**Integrations**

- Applications
- Scheduled reminders

**Archives**

- Security log
- Sponsorship log
- <> Developer settings**

GitHub Apps

- OAuth Apps
- Personal access tokens

### New personal access token

Personal access tokens function like ordinary OAuth access tokens. They can be used inst HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

**Note**

giyong\_token

What's this token for?

**Expiration \***

No expiration The token will never expire!

GitHub strongly recommends that you set an expiration date for your token to help ke [Learn more](#)

**Select scopes**

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events




## Personal access tokens

### Personal access tokens

Generate new token Revoke all

Tokens you have generated that can be used to access the [GitHub API](#).

Make sure to copy your personal access token now. You won't be able to see it again!

✓ ghp_gbDebtAgng31CgnpCdUAxY08QwH270B44XA 	Delete
yong_token — admin:org, delete_repo, repo, write:discussion ⚠ This token has no expiration date.	Last used within the last 2 months Delete

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).



# Github 연동

기존의 github password를 token으로 변경

Eclipse > Git Repository > origin > 깃저장소 > 우측 Change Credentials

