







- 1. 깃 설정변수와 값
- 2. 깃 설정 명령 구조
- 3. 주요 깃 설정
- 4. 깃 저장소 생성



- 1. 깃의 설정 명령을 이해하고 깃 설정을 할 수 있다.
- 2. 깃 저장소 생성 전에 주요 후 깃 설정 6개를 실행할 수 있다.
- 3. 깃 실행 후 깃 저장소를 생성할 수 있다.









1 깃설정

### 🐡 깃 설정

- ☑ 깃을 설치한 이후 깃 배시에서 처음 할 일
  - 사용자의 이름과 전자메일 설정
    - 설정값에서 큰따옴표와 작은따옴표 모두 사용 가능하며 공백이 없으면 따옴표 생략이 가능
- ▼ 주요 설정변수와 값

## 설정변수 설정값

### 주요 설정변수와 값

core.editor ['codewait'   'notepad']		기본 편집기 설정	
core.autocrlf [true   false   input]		줄바꿈 자동변환	
core.safecrIf [true   false]		줄바꿈 안전확인	
user.name ai7dnn		사용자 이름	
user.email ai7dnn@gmail.com		사용자 전자메일	
init.defaultBranch main		기본 브랜치 이름	





#### 깃설정

### 🍅 설정 명령과 설정 범위

- ☑ 설정 명령 구조
  - \$ git config --설정범위 설정변수 설정값
- ☑ 설정 범위
  - [--system | --global | --local]
    - SystemGlobalLocal
      - ➡ 모든 사용자 ➡ 현재 사용자의 모든 저장소 ➡ 현재 사용자의 현재 저장소

### ☑ 설정 명령 사용

PC@DESKTOP-482NOAB MINGW64 /c/[smart Git]

\$ git config --global user.name ai7dnn

PC@DESKTOP-482NOAB MINGW64 /c/[smart Git]

\$ git config --global user.email ai7dnn@gmail.com

PC@DESKTOP-482NOAB MINGW64 /c/[smart Git]

\$ git config --global init.defaultBranch main



#### 1 깃설정



### ☑ 전역 설정은 해당 사용자에게만 적용되는 설정으로 명령

- --global 옵션
- 설정 내용은 .gitconfig 파일에 저장되며
  - 윈도에서는 C:₩Users 하부의 사용자 계정 폴더
- 사용자 계정인 PC인 경우의 설정 파일
  - C:\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Users\Use

주요 명령		
\$ git configglobal core.editor notepad	기본 편집기를 노트패드로 지정	
\$ git configget core.editor	기본 편집기 조회·참조	
\$ git configglobaledit	전역 설정 파일 편집	
\$ git configglobal -e	선탁 결정 씨걸 현법	











● 옵션 --edit이나 -e를 사용

주요 명령		
\$ git configglobal core.editor notepad	기본 편집기를 노트패드로 지정	
\$ git configget core.editor	기본 편집기 조회·참조	
\$ git configglobaledit	전역 설정 파일 편집	
\$ git configglobal -e	선탁 결정 씨를 한법	





#### 1 깃설정



### 

- 윈도에서 autocrlf를 true로 설정
- 맥과 리눅스에서 input으로 설정

### 營 줄바꿈 설정이 필요한 이유

- 플랫폼에 따라 줄 끝의 처리 방법이 다르기 때문
  - windows : CR(Carriage Return)와 LF(Line Feed)로 한 줄의 끝을 인식
  - mac/linux : LF(Line Feed)로 한 줄의 끝을 인식

#### 주요 명령

. 33		
\$ git configglobal core.autocrlf true	줄바꿈 처리를 자동으로 지정	
\$ git configget core.autocrlf	줄바꿈 처리 방법을 조회·참조	
\$ git configglobal core.safecrlf false	줄바꿈 안전 설정을 false로 지정	
\$ git configget core.safecrlf	줄바꿈 안전 설정을 조회·참조	





### 1 깃설정

### 줄바꿈 안전 설정

### 

- 설정이 잘못되면
  - 명령 add나 commit에서 다음과 같은 줄바꿈 경고 메시지가 발생
- 맥과 리눅스에서 input으로 설정
  - 명령 add나 commit에서 심각한 오류가 발생할 수 있으니 주의가 필요

PC@DESKTOP-482NOAB MINGW64 /c/[smart Git]/gcrlf (main)

\$ git add file.txt

warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next time Git touches it

#### 주요 명령

\$ git config --global core.autocrlf true 줄바꿈 처리를 자동으로 지정
\$ git config --get core.autocrlf 줄바꿈 처리 방법을 조회·참조
\$ git config --global core.safecrlf false 줄바꿈 안전 설정을 false로 지정

\$ git config --get core.safecrlf 줄바꿈 안전 설정을 조회·참조





### 깃설정

### ₩ 저장소 생성 전에 다음 반드시 설정 확인 후 시작

### ₩ 다음 설정 명령

- 자신이 사용하는 컴퓨터라면 1회 설정 이후에 설정 파일에 저장
  - 수정하지 않는다면 이후 다시 설정할 필요 없음

#### 주요 6가지 설정

1 2 3 1 1 2 8	
\$ git configglobal user.name ai7dnn	사용자 이름
\$ git configglobal user.email ai7dnn@gmail.com	사용자 전자메일
\$ git configglobal core.autocrlf true	줄바꿈 자동변환
\$ git configglobal core.safecrlf false	줄바꿈 안전 설정
\$ git configglobal core.editor 'codewait'	기본 편집기 설정
\$ git configglobal init.defaultBranch main	기본 브랜치 이름









#### **2** 저장소생성

- 🐞 git init
  - ☑ 디렉토리를 git repository로 만들어야 git으로 버전 관리가 가능
  - ❤ 2가지 방식
    - \$ git init
    - \$ git init .
      - 현재 디렉토리를 git repository로 만들기 위해서 사용
    - \$ git init basic
      - 현재 폴더 하부에 폴더 basic을 생성하고 git repository로 만들기 위해서 사용

PC@DESKTOP-482NOAB MINGW64 /c/[smart Git]

\$ git init basic

Initialized empty Git repository in C:/[smart Git]/basic/.git/



#### **2** 저장소생성

### ₩ 저장소 생성 확인

- ৺ 생성된 깃 저장소 폴더 basic으로 이동
  - 하부에 폴더 .git를 확인

```
PC@DESKTOP-482NOAB MINGW64 /c/[smart Git]
$ cd basic
PC@DESKTOP-482NOAB MINGW64 /c/[smart Git]/basic (main)
$ ls -al
total 8
drwxr-xr-x 1 PC 197121 0 Dec 12 16:43 ./
drwxr-xr-x 1 PC 197121 0 Dec 12 16:43 ../
drwxr-xr-x 1 PC 197121 0 Dec 12 16:43 ../
```





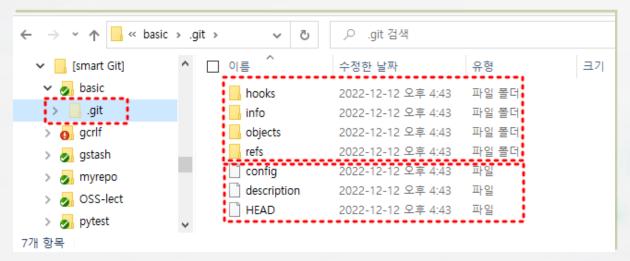
#### 저장소생성

### 🐡 깃 저장소 하부 .git 폴더

- ✓ 깃 저장소에는 반드시 .git 폴더가 포함
  - 여러 폴더와 파일이 저장
    - 커밋된 모든 파일의 모든 버전 기록이 저장

### ✓ 기 저장소에서 폴더 .git을 삭제

○ 깃 저장소로서의 기능을 상실







### 저장소생성



### ☑ 기준 폴더를 하나 정하고 git bash 실행

• C:₩[git-master]

	명령	설명
	\$ git configglobal user.name hskang	사용자 이름 설정
	\$ git configglobal user.email hskang@gmail.com	전자메일 설정
0	\$ git configglobal core.autocrlf true	줄바꿈 자동 설정
	\$ git configglobal core.safecrlf false	줄바꿈 안전 설정
	\$ git configglobal core.editor 'codewait'	기본 편집기 설정
	\$ git configglobal init.defaultBranch main	기본 브랜치 이름 설정
2	\$ git init basic	저장소 생성
	\$ cd basic	폴더 이동
	\$ Is -al	파일 확인





## Summary

#### >> 깃 설정 명령 구조

- ◆ \$ git config --설정범위 설정변수 설정값
- ◆ 설정 범위
  - [--system | --global | --local]
- ◆ --global 인 경우 설정 파일
  - C:₩Users₩[사용자-계정]₩.gitconfig





## Summary

#### >> 지역 저장소 생성 전 다음 설정 반드시 수행

- ◆ 사용자 설정
  - \$ git config --global user.name "hs kang"
  - \$ git config --global user.email hsakng@gmail.com
- ◆ 맥(If)과 윈도(crlf) 간의 자동 변환
  - \$ git config --global core.autocrlf true
- ◆ 뉴라인 경고 발생 없애기(옵션)
  - \$ git config --global core.safecrlf false
- ◆ 기본 편집기 설정
  - \$ git config --global core.editor 'code --wait'
- ◆ 깃 저장소 기본 브랜치 이름 설정
  - \$ git config --global init.defaultBranch main





# Summary

### >>> git repository 생성 방법

- \$ git init
- \$ git init .
  - 현재 디렉토리를 git repository로 만들기 위해서 사용
- \$ git init basic
  - 현재 폴더 하부에 폴더 basic을 생성하고 git repository로 만들기 위해서 사용