

# 04~06

## 비주얼 스튜디오 코드(vs code)

- MS사가 개발하는 오픈 소스 에디터 소프트웨어
- JavaScript, TypeScript 및 Node.js를 기본적으로 지원
- 다른 언어(ex.C++,C#,Java,Python,PHP,GO) 및 런타임(ex. .NET 및 Unity)에 대해 풍부한 확장 기능 시스템
- 파일 편집과 버전 관리로 지원
- 유저 버전 : 현재 사용자만 설치 가능 / 사용자 폴더에 설치
- 시스템 버전 : 해당 컴퓨터에 모든 사람 사용 가능 / 시스템 설치 기본 폴더 / 공용 버전으로 사용 추천

## Git 명령어

### 버전 확인

```
$ git --version
```

### 설정 확인

```
$ git config --list
```

### 도움

```
$ git --help
```

## 깃 설정 명령 구조

```
$ git config --설정범위 설정변수 설정값
```

- 설정 범위

```
[- --system | --global | --local]
```

- --global인 경우 설정 파일

```
C:\Users\[사용자-계정]\.gitconfig
```

지역 저장소 생성 전 다음 설정 반드시 수행

## 사용자 설정

- `$ git config --global user.name ~`
- `$ git config --global user.email ~`

## 맥(lf)과 윈도우(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`

## 리눅스 명령어

### < 폴더 관련 명령 >

`$ pwd`

print Working Directory / 현재 폴더 표시

`$ cd`

Change Directory

`$ mkdir dname`

Make Directory

`$ ls`

File or folder list

- `-l` : 파일의 상세 정보
- `-a` : 숨김 파일 표시
- `-al` : 위 두개를 조합

### < 파일 관련 명령 : 생성 >

`$ touch fname`

빈 파일 fname 생성(ex. touch a b c = 파일abc를 한번에 만듦)

< 표준 출력으로 결과 표시 echo >

이후 문자열을 컴퓨터 터미널에 출력(disply)하는 명령어

```
$ echo git bash / $ echo 'print()'
```

단순한 문자열은 가능하나 괄호 등이 있는 경우, 작은 또는 큰 따옴표 사용

< 파일 관련 명령 : 내용 보이기, 복사, 이름 바꾸기 >

```
$ cat fname
```

파일 내용 보이기(ex. cat a b c = abc를 한번에 보이기)

```
$ cp a b
```

파일 a를 b로 복사

```
$ mv f1 f2
```

파일 f1을 f2로 이름 수정

```
$ rm fname
```

파일 fname 삭제

```
$ rm -rf dname
```

하부에 서브폴더와 파일이 있어도 폴더 삭제, 옵션 사용

- **-f** : 강제로 파일이나 디렉토리를 삭제
- **-r** : 디렉토리 내부의 모든 내용을 삭제

< 명령어 ls >

현재 위치의 파일 목록 조회

**ls -l** : 파일의 상세 정보

**ls -a** : 숨김 파일 표시

**ls -t** : 파일들을 생성 시간 순(제일 최신 것부터)으로 표시

**ls -rt** : 파일들을 생성 시간 순(제일 오래된 것부터)으로 표시

**ls -f** : 파일 표시 시 마지막 유형에 나타내는 파일명을 끝에 표시

- **'/'** : 디렉터리

- `'*'`: 실행파일
- `'@'`: 링크 등등

#### < 명령어 cat(concatenate) >

파일의 내용을 화면에 출력

- `cat file1`

file1의 내용을 출력

- `cat file1 file2`

file1과 file2의 내용을 출력 내용을 출력

— 파이프(pipe)기호인 `|`(vertical var)의 의미 —

앞 명령의 표준 출력을 뒤 명령의 표준 입력으로 처리

- `cat file1 file2 | more`

file1과 file2의 내용을 페이지별로 출력

- `cat file1 file2 | head`

file1과 file2의 내용을 처음부터 10번째 줄까지만 출력

- `cat file1 file2 | tail`

file1과 file2의 내용을 끝에서부터 10번째 줄까지만 출력

#### < Redirecrion 명령어 연산 `>`, `>>`, `<` >

`'>','>>'`: 화면의 출력 결과를 파일로 저장

- `'>'` 기호

기존에 있는 파일 내용을 지우고 저장

- `'>>'` 기호

기존 파일 내용 뒤에 덧붙여서 저장

- `'<'` 기호

파일의 데이터를 명령에 입력

— 활용사례 —

- `echo aaa > a.txt`

(새)파일 a.txt에 문자열 aaa 복사(replace)

- `echo bbb >> a.txt`

(새)파일 a.txt에 문자열 bbb 추가

- `cat file1 file2 > file3`

file1,file2의 명령 결과를 합쳐서 file3이라는 파일에 저장

- `cat file4 >> file3`

file3에 file4의 내용 추가

- `cat < file1`

file1의 결과 출력

< 저장소 삭제 후 다시 저장소 설정 >

전체 폴더를 삭제하거나 하부폴더 .git 삭제

\$ `rm -rf.git`

- `-f` : 강제로 파일이나 디렉토리를 삭제
- `-r` : 디렉토리 내부의 모든 내용을 삭제