

c언어

강사명: 최수길 강사님(과장)

휴대폰: 010-4573-5540

github: github.com/freshmea

카톡ID: freshmea

---

1일차

성취 - 무엇을 이룰 것인가

집중 - 무엇에 집중할 것인가

1. github를 잘 가꿔라 - 학습 정리
2. 포트폴리오(프로젝트) 준비 - github로 올리기
  - 계획, 과정(자료수집, 작성, 디버깅), 결과
  - 이런 육성 과정의 프로젝트는 주의 깊게 보지 않기 때문에, 개인 프로젝트를 진행하여야 함
    - > 거창하지 않더라도 작은 거라도 해보자
  - 정보를 찾고 만들어 보았다라는 것만으로도 인식이 좋아짐
    - > 수동적이 아닌 능동적인 인상을 줄 수 있음
3. 알고리즘 연습
  - 이력서에 소개하는 케이스도 있음(논리력 입증/과거엔 비중이 높았으나 현재 비중이 낮긴함)
  - 이를 테스트 할 수 있는 사이트가 있음
- 1) 백준
- 2) 프로그래머스
- 3) 삼성 소프트웨어 ~

AI의 발전으로 코딩보다는 프로젝트 전반을 이해하는 능력이 더 중요해져 버림  
(코딩은 기본으로 깔고 가는 것, 무시가 아님)

---

운영체제 OS - windows, linux, mac

1. Linux
  - 1) Ubuntu
  - 2) Android
  - 3) Redhat
  - 4) CentOS
  - 5) Raspbian

## # 가상환경 설정 방법

1. VMware Workstation Pro 설치(카톡으로 배포)
    - 가상환경 안에서 os(운영체제)로 리눅스를 설치하여 실습
  2. ubuntu 22.04 LTS Desktop image download
    - 범용 리눅스인 우분투를 사용
  3. VMware Player에서 새로운 가상 환경 생성
    - 1) VMware Player을 실행하여 ubuntu 설치한 것을 선택
    - 2) 프로젝트 생성하여 프로젝트명 작성 및 아이디 비번 만들기(aa)
      - 비번은 자주 사용하니, 중요 프로젝트가 아니라면 간단히
      - 세부설정: 하드 50~100GB / 가상 램은 4~8GB / Processors core 6  
(코어는 높으면 경고 문구를 띄워주니 안 뜰 때까지 낮추기)
    - 3) 디폴트값 따라가며 키보드는 나중에 변경하니 아무거나, 시간설정은 서울로 하면 됨
    - 4) 설정에서 한글로 바꿀 때 파일명에 대해서도 한글로 바꾸는 것은 적용하지 않기
    - 5) cmd를 여는 방법: ctrl + alt + t
  4. 설정에서 지역 및 언어 설정에서 한국어를 고른다. -> 재부팅 시 ui가 한글로 바뀜
    - > 만약 폴더명도 한글로 바꾸냐고 물어볼 시, “”“NO”“” 파일명이 한글로 바뀔 시 추후 패스 선언 때 오류 발생 가능.
  5. 입력기에서 한국어를 누르면 (hangle)이 있음
    - 톱니바퀴 달린 거
    - 그거 기본 설정에서 한/영 키가 안 먹히므로 대신할 키로 추가해야 한/영 키보드를 이용할 수 있음
    - 변경 후, 우상단에 입력기를 눌러 한국어(hangle)로 설정 되었는지 확인해주기
- ex) infinit.c
- ```
int main(){  
    return 0;  
}
```

mkdir 폴더명 -> 폴더명 디렉터리 생성 (make directory)  
cd 폴더명 -> 을 통한 폴더명으로 이동 (change directory)  
tab -> 을 통한 자동완성 기능이 있음  
touch 파일명 -> touch를 통한 빈 파일을 생성  
ls -> 해당 디렉터리의 내용물 리스트 불러오기 (list)  
nano 파일명 -> 해당 파일 편집 (nano에서는 tab으로 들여쓰기 안 됨, 차라리 4번 띄우기)  
ctrl s -> 저장  
ctrl x -> 나가기  
cat 파일명 -> 내용물 보기  
cc 파일명 -> 해당 파일 적용/컴파일  
./ -> 접근 명령  
./a.out -> 컴파일된 해당 파일 실행  
ps -> 프로세스가 실행 중인지 알아볼 때 (process)  
ctrl c -> 프로세스 중단  
sudo apt install 패키지명 -> sudo(관리자 권한 or 강제로) 패키지를 설치

sudo apt-get install git

git init -> github에 업로드용 링크 생성  
git remote add origin 링크-> git 링크와의 연결  
ls - al -> 숨긴 폴더 포함한 리스트 출력(.폴더명은 숨김 폴더이기 때문)  
git fetch origin -> 연결된 git에서 새로운 브랜치 생성  
git checkout -b main origin/main-> 메인이라는 것을 브랜치로 지정 및 따라가게 설정

git pull -> git 클라우드 변경점 당겨오기  
git add 폴더명 -> 폴더명(README.md)을 git으로 올리기위해 추가  
git commit -> 변경점 commit  
git 관련 유저 이메일과 이름 등록 과정이 있는데 뜨는거 보고 하면 됨  
git push -> 해당 기기에서의 변경점을 클라우드로 올리기(터미널에서 안 될거임)  
-> vscode와 같이 git 연동을 지원하는 스크립트 툴을 이용할 것.

commit&sync를 함으로써

나의 수정된 변경 점을 클라우드에 올리고, 클라우드에 저장되어 있는 다른 이의 변경점을 가져옴  
-> cloud와 local의 데이터의 서로 다른 내용이 없도록 동일하게 해줌

**\*\*commit, commit&sync 작업시 message를 꼭 첨부할 것!!**

무슨 내용에 대해 수정이 있었는지 메모하면 추후 확인 및 다른 사람이 확인하기 용이.

안 쓰면, 쓰라고 혼나니 asdf라도 적는 습관 들이기

클라우드

github -> ID -> Repository

local의 .git 폴더 -> Repository: push

Repository -> .git 폴더 : fetch

저장 시점 저장 : commit

local의 시점에서 cloud를 보는 것 : remote

branch(브랜치) -> origin/main -> main(local)

---

변수 설정 (전역변수 지역변수)

메모리에 타입을 설정해서 할당

메모리의 주소가 있다

```
int var_a;
```

```
var_a = 32 ;
```

ex) 00001이란 주소에 저장이 됐으면 &var\_a를 통해서 주소에 저장된 32란 값을 불러옴

함수 scope(범위)

```
main(){
```

```
    int a;
```

```
    .
```

```
    .
```

```
    .
```

```
    ~
```

```
}
```

즉, main{}안에서 지정된 것이기에 안에서만 사용가능

scanf란 함수를 사용한다면 주소를 통해 불러옴

## ## 1일차 학습 내용

### 1. 가상환경 Vmware(하드, 메모리 할당)

- window 환경에서 돌린다면 - wsl2 -> 최적화해서 실행하기에 빠름

### 2. Ubuntu 22.04

### 3. 텍스트 프로그램

#### 1) nano, vi (기본 제공 프로그램 ide)

#### 2) vscode(확장성이 용이)

- AI인 copilot을 연동
- ssh 연결한 원격 작업
- git과의 연동 (github는 클라우드 서비스)

### 4. c 언어

숙제0. github에 오늘 수업 다 올려라

숙제1. vscode 단축키 숙지 + 확장프로그램 확인

숙제2. README.md에서 md라는데 마크다운 문법의 요약이다

숙제3. make 문법에 대해 공부해라

ex) make 파일의 구조 - 여러 파일을 묶음명으로 지정한 후에 동일한 규칙에 대해 수행 시킴  
묶음 이름: 파일명,

규칙: 묶음이름