

# 1장 C언어 시작하기

---

한림대학교 소프트웨어학부 양은샘.

# 1장 C언어 시작하기

---

- 안녕하세요? 여러분!
- 오늘은 C 프로그램을 작성해 보는 첫번째 시간입니다.
- 이번 장에서는 C 언어의 특징과 가장 간단한 C 프로그램의 구성 형태를 학습하도록 하겠습니다.
- 그리고 Visual Studio Code 개발 환경을 설치해, 여러분께서 작성한 C 프로그램을 컴파일하고 실행해 볼 수 있도록 하겠습니다.
- Visual Studio Code 개발 환경은 설치가 간단하며, 윈도우 및 리눅스 운영체제 모두에서 사용 가능합니다. 또한 C 표준 gcc 컴파일러를 사용하기 때문에 작성된 프로그램의 아무런 변경없이 리눅스에서도 사용가능 합니다.
- 그럼 학습을 시작하도록 하겠습니다.

# 학습 목차

---

1.1 C 언어의 역사 및 특징

1.2 C 언어의 기초

1.3 C 프로그램의 작성과 실행

1.4 Visual Studio Code 개발환경 설치

- ☐ 개념 확인 학습
- ☐ 적용 확인 학습
- ☐ 응용 프로그래밍

# 학습 목표

---

- C 언어의 역사 및 특징을 안다.
- C 프로그램의 기본 구조를 안다.
  - 라이브러리(Library), 선행처리기, main() 함수와 return, 명령문(Statement), 설명문(Comment)
- Visual Studio Code 개발 환경을 설치 한다.
- C 프로그램을 작성하고 실행해 결과를 확인 한다.
- 개념 확인 학습으로 배운 내용을 정리한다.
- 적용 확인 학습으로 개념 습득 여부를 확인한다.
- 응용 프로그래밍으로 문제해결력을 키운다.

# C 언어의 역사

---

- 개발

- 1972년 UNIX에서의 새로운 운영체제를 설계하던 중 미국 Bell 연구소에서 개발.

- 발전과정

- ALGOL -> CPL -> BCPL -> B -> C

- 표준

- 1989년 미국표준협회(ANSI: American National Standard Institute)에서 ANSI C 표준안 발표.
- 1990년에 ANSI가 C언어 표준을 국제 표준화 기구(ISO)에 넘김.
- 현재 시점에서 C의 최신 표준은 ISO의 C18.

# C 언어의 특징

---

- C 언어는 시스템 프로그램을 개발하는 프로그래밍 언어로 적합.
  - UNIX의 90% 이상이 C 언어로 작성 되어 있음.
- 일반적인 문제를 처리 할 수 있는 사용자 프로그래밍 언어의 기능도 갖추.
- 함수들의 집합으로 구성된 언어이기 때문에 모듈 설계 가능.
- 다양한 데이터형, 연산자로 자료구조의 표현에 적합.
- 포인터를 사용하여 주소에 대한 처리 가능.
- 프로그램 코드에서 대/소문자를 구별해 사용.

# C 언어의 기초

---

- [예제 1.1] Hello C world!!

#include <stdio.h>	//선행처리기 정의, Library (header file)
#define PI 3.14	//매크로 선언
int main(void)	//main() 함수의 시작
{	
printf("Hello");	//명령문, 출력 함수1
printf("C world!!\n");	//명령문, 출력 함수2
printf("%f", PI);	
return 0;	//main() 함수를 호출한 운영체제로 오류 없음을 보고하고 복귀
}	//함수의 종료

# 라이브러리(Library)

---

- 라이브러리는
  - 프로그램 작성과정에서 자주 나오는 기능이나 복잡한 프로그램을 C 컴파일러 제조업체나 사용자가 함수(모듈)화 하여 미리 작성해 놓은 일종의 프로그램 집합을 의미.
- 헤더파일에는 각 함수들을 기능에 따라 분류하여 함수의 이름, 전달 인자, 반환 값 형태들이 정의 됨.
- 누가 작성했느냐?에 따라
  - 표준 라이브러리(Standard Library)
    - 컴파일러 제조업체나 표준화 단체에서 작성된 것.
    - 표준 라이브러리에 관련된 헤더 파일은 < >를 사용 : `#include <stdio.h>`
  - 사용자 라이브러리(User Define Library)
    - 사용자가 필요에 의해 만든 프로그램(함수).
    - 사용자 라이브러리에 관련된 헤더 파일은 " "를 사용 : `#include "user_define_library.h"`



# 선행처리기(Preprocessor)

---

- `#include <stdio.h>`와 같이 `#`이 붙은 문을 선행처리기라 함.
  - `#include` 선행처리기는 지정된 파일을 프로그램 내에 포함시키라는 지시어.
  - `stdio.h` 라는 파일을 프로그램 내에 포함시키라는 의미.
- 선행처리기의 역할
  - 매크로 기능, 파일 포함 기능, 선택적 컴파일 기능 수행.
- 매크로 선언  
`#define PI 3.14`
  - 프로그램 내부에서 `PI` 단어는 3.14로 처리.

# main() 함수

---

- main()은 C 프로그램 시작점을 알리는 특수한 함수.
  - C 프로그램은 여러 개의 함수들로 구성.
  - 사용된 괄호(( ))는 함수를 의미.
- 모든 C 프로젝트는 반드시 하나의 main() 함수를 가짐.
- C 프로그램의 동작은 main() 함수의 시작부터 main() 함수의 끝 부분까지 수행 후 종료.
- 대괄호({ })로 쌓여진 부분
  - 함수의 시작과 끝의 유효 범위 표현.
  - 여러 명령문들을 하나의 단위로 그룹화 하여 처리할 때도 사용.

# main() 함수와 return

---

- `int main(void)`
  - `int`는 `main()` 함수가 끝나고 이 함수를 호출한 운영체제로 복귀할 때 정수를 반환한다는 의미.
  - `void`는 메인 함수가 시작할 때 외부로부터 받은 매개변수가 없음을 의미.
  - `void`는 생략이 가능 : `main()`
  - 만약 외부로부터 어떤 입력이 필요하게 되면 `int main(int argc, char *argv[])` 사용 : 7장 함수에서 설명.
- `return 0;`
  - 운영체제로 복귀할 때 오류 없음인 정수 0을 반환하고 `main()` 함수를 종료한다는 의미.

# 명령문(Statement)

---

- 함수 내부에서 원하는 작업을 수행하도록 작성된 문장.
- 주의할 점
  - 각각의 모든 명령문은 세미콜론(;)으로 끝나야 함.
- `printf("Hello");`
  - 큰따옴표 안에 있는 문장을 모니터 화면으로 출력하라는 명령문.

# 설명문(Comment)

---

- 프로그램의 내용을 설명하기 위해 또는 이해를 돕기 위해 사용하는 문장.
- 프로그램의 동작에 전혀 영향을 주지 않는 문장.
- C 프로그램에서 설명문의 작성방법.
  - 한 줄 설명문 : 슬래시 두 개(//)를 연속으로 사용.
  - 한 줄 이상의 설명문 : 설명의 시작(/\*) 과 설명의 끝(\*/) 두 기호 사이에 내용을 넣음.

# C 프로그램 작성 형식

//바른 작성 방법

```
#include <stdio.h>
#define PI 3.14
int main(void)
{
    printf("C programming");
    printf("language");
    printf("%f", PI);
    return 0;
}
```

//실행결과

C programminglanguage3.140000

//비 추천 형식 1

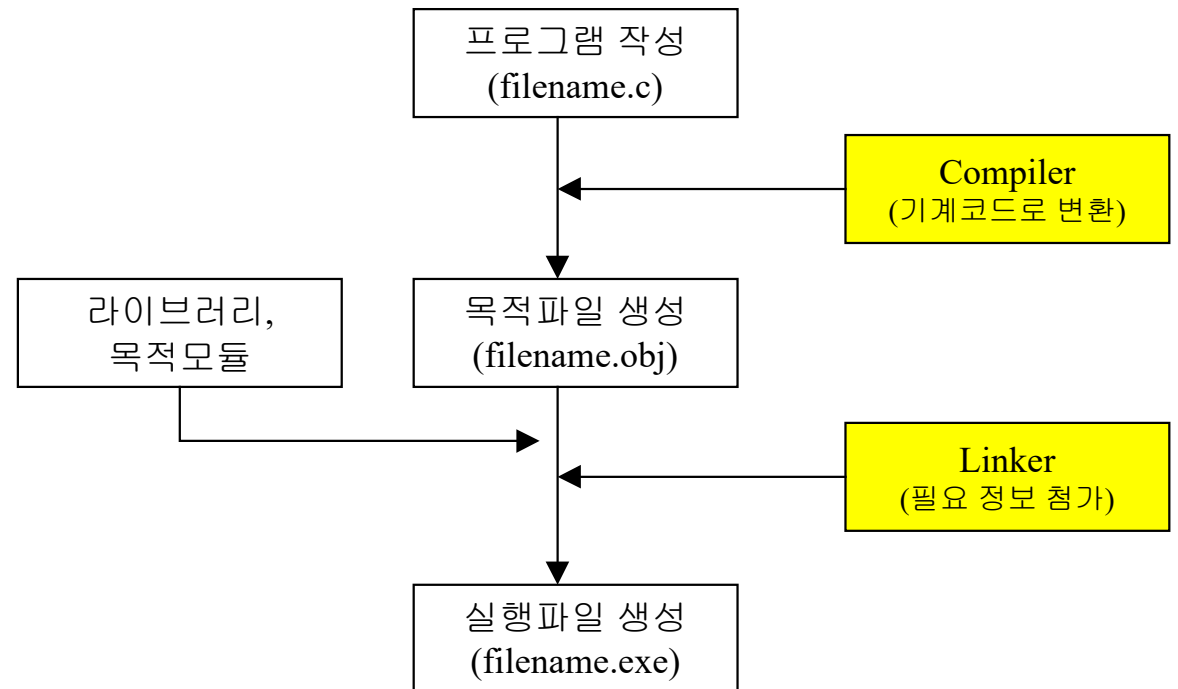
```
#include <stdio.h> #define PI 3.14 int main(void)
{ printf("C programming"); printf("language");
printf("%f", PI); return 0; }
```

//비 추천 형식 2

```
#include <stdio.h>
#define PI 3.14
int main(void)
{
    printf("C programming");
    printf("language");
    printf("%f", PI);
    return 0;
}
```

# C 프로그램의 작성 및 실행 과정

- C 프로그램 작성 방법
  - 사용자는 원하는 작업을 수행할 수 있는 프로그램 코드만 작성하면 됨.
  - 나머지 과정은 컴파일러가 알아서 처리 함.
- 사용자 C 프로그램 코드 작성
  - 소스 파일 : .c 확장자 사용
- 사용자 C 프로그램 코드 실행
  - 컴파일러에 의해서 만들어진 실행파일 실행 : .exe 확장자



# C, C++ 개발환경 두 가지

---

- Microsoft Visual Studio Code (vscode)
  - Microsoft Visual Studio 20xx 보다 가볍고,
  - 윈도우/리눅스/맥(애플PC) 모두에서 사용할 수 있으며,
  - gcc, g++(GNU Compiler Collection) 컴파일러를 사용하는 등 여러 가지 장점이 있음.
- Microsoft Visual Studio 20XX



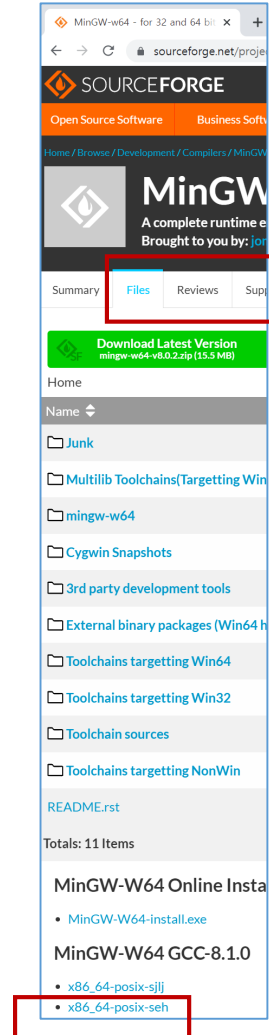
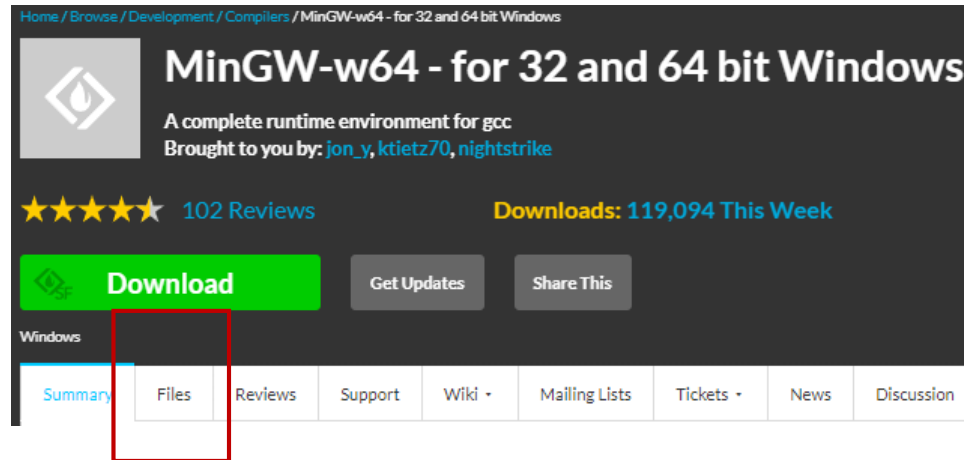
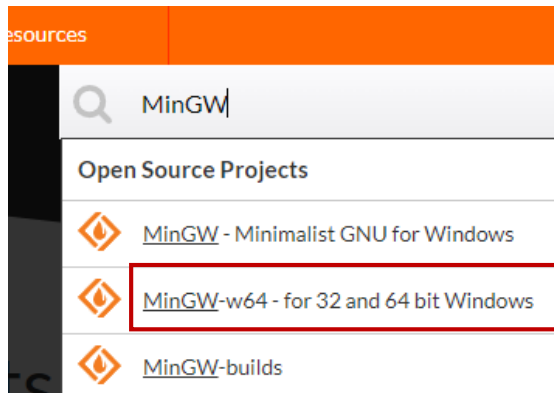
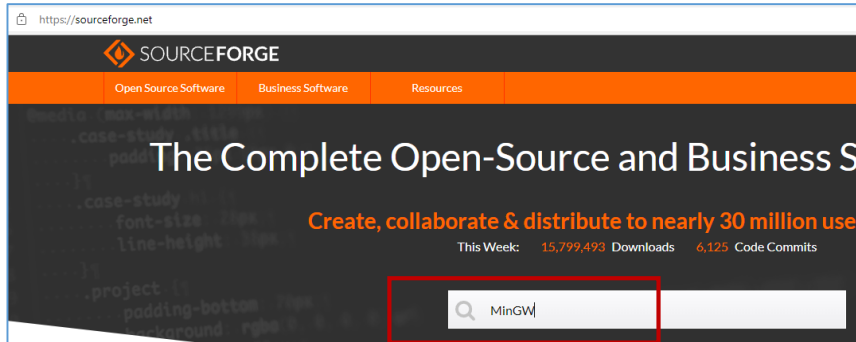
# C, C++ 개발환경 설치 및 실행

---

- 개발환경 설치 순서
  - 애플 PC 사용 시
    - Visual Studio Code 설치
    - vscode 시작 화면의 [Extention] 버튼을 클릭하여 필요한 개발 환경 설치 : C/C++ install, Code Runner install
  - Microsoft 윈도우 PC 사용 시
    - MinGW(Minimalist GNU for Windows) 설치.
      - 윈도우 환경에서 gcc, g++ 컴파일러를 사용할 수 있도록 해주는 유틸리티.
    - Visual Studio Code 설치
    - vscode시작 화면의 [Extention] 버튼을 클릭하여 필요한 개발 환경 설치 : C/C++ install, Code Runner install
- c 또는 cpp 프로그램 작성 및 실행

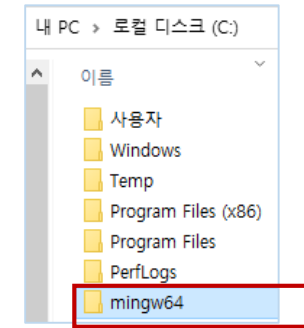
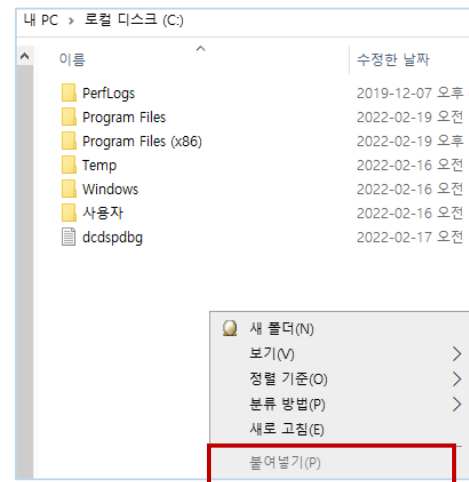
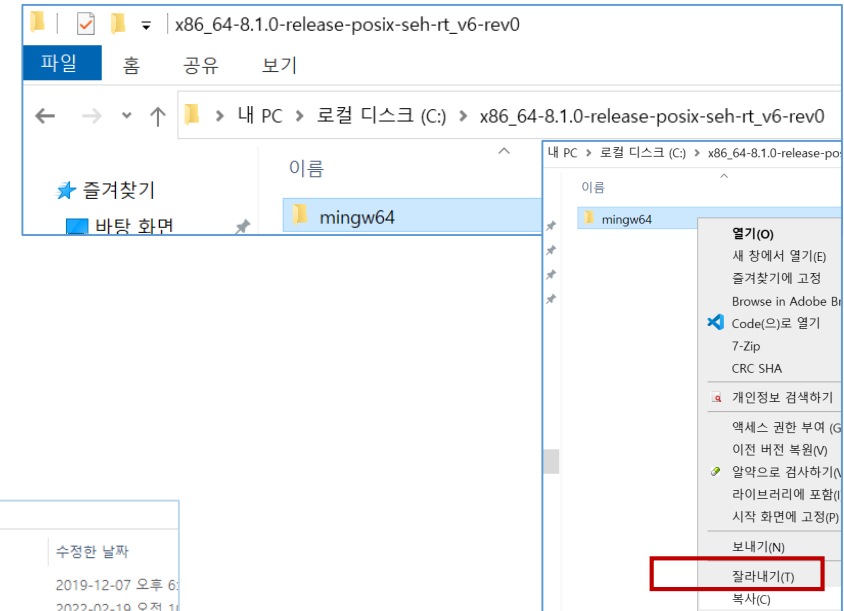
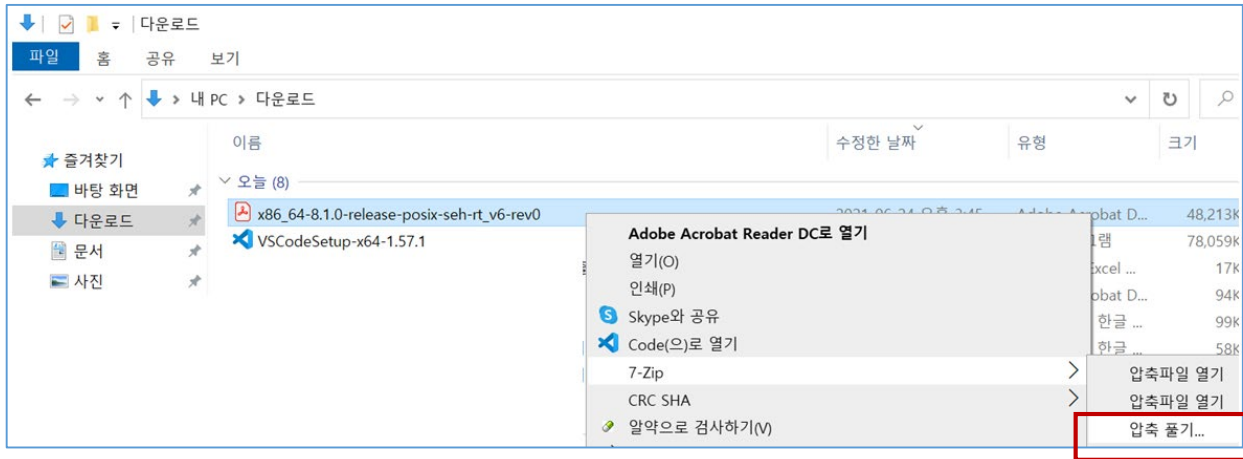
# 윈도우 사용자 : MinGW-w64 설치 (1)

- MinGW-w64 설치(<https://sourceforge.net/>)



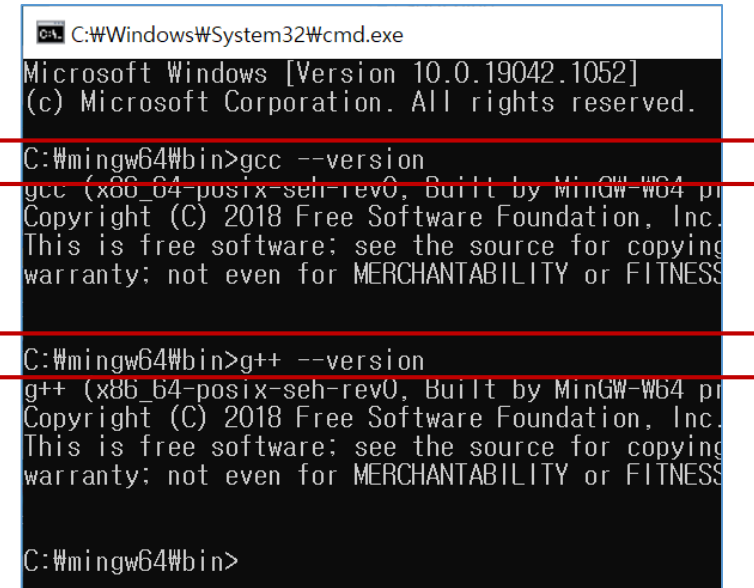
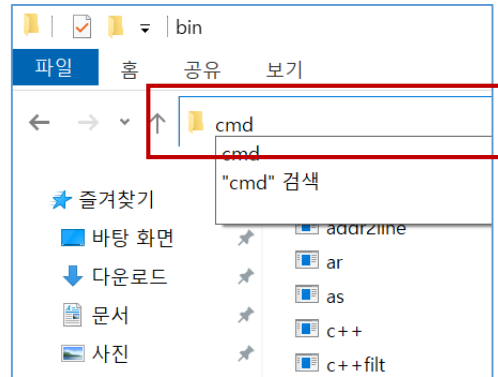
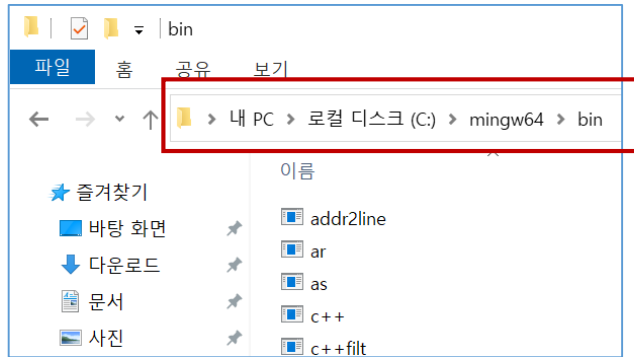
# 윈도우 사용자 : MinGW-w64 설치 (2)

## ■ 압축 풀고 C:\w에 복사



# 윈도우 사용자 : MinGW-w64 설치 (3)

## ■ MinGW-w64 설치 확인



# 윈도우 사용자 : MinGW-w64 설치 (4)

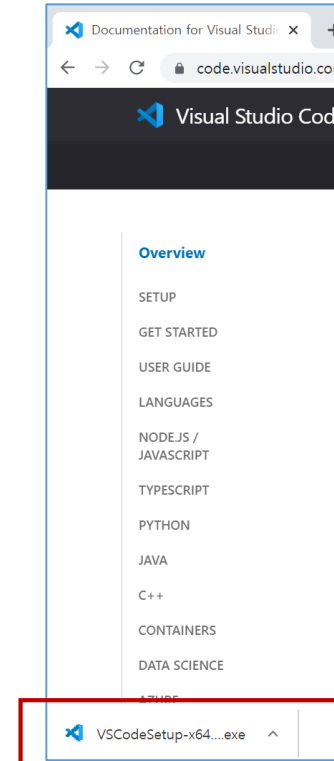
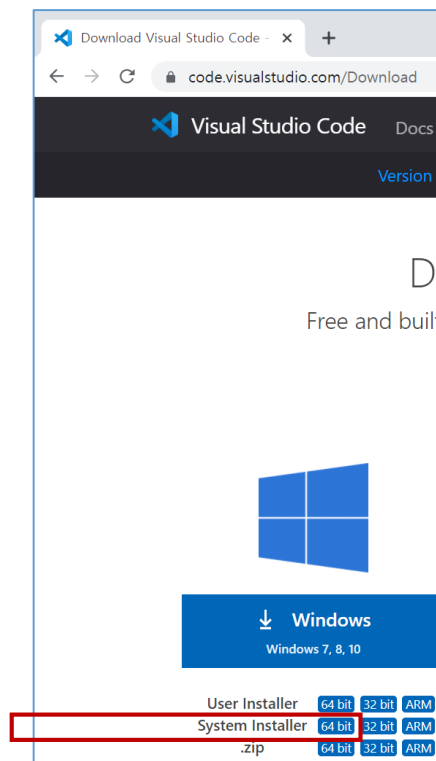
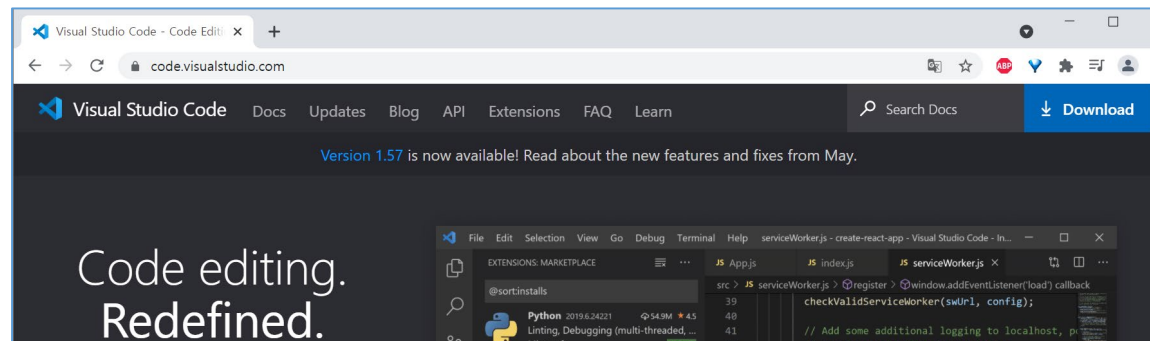
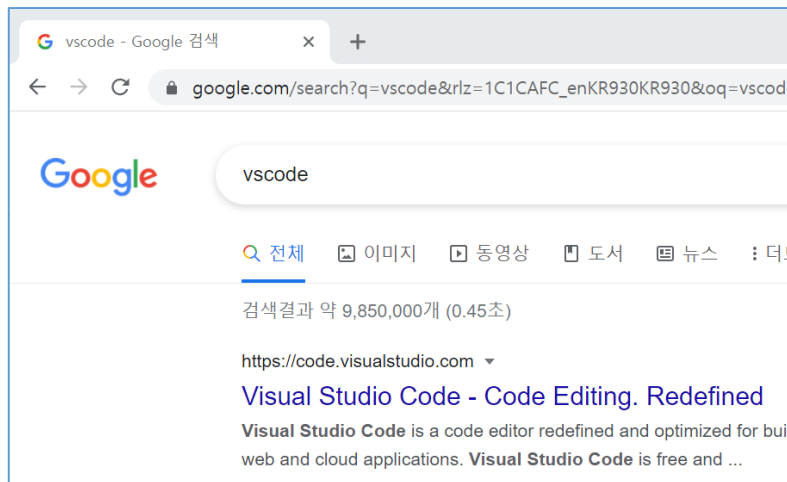
## ■ 환경변수 설정

The following steps illustrate the process of setting environment variables for MinGW-w64:

- Step 1:** Open the Start menu search bar and type "속성(R)".
- Step 2:** In the Settings app, navigate to **시스템** (System) > **정보** (Information). Click on **고급 시스템 설정** (Advanced system settings).
- Step 3:** In the **시스템 속성** (System Properties) dialog box, select the **고급** (Advanced) tab. Click on the **환경 변수(N)...** (Environment variables...) button.
- Step 4:** In the **환경 변수 편집** (Edit environment variables) dialog box, select the **시스템 변수(S)** (System variables) tab. Find the **Path** variable, click **편집(I)...** (Edit...), and add the path **C:\mingw64\bin**.

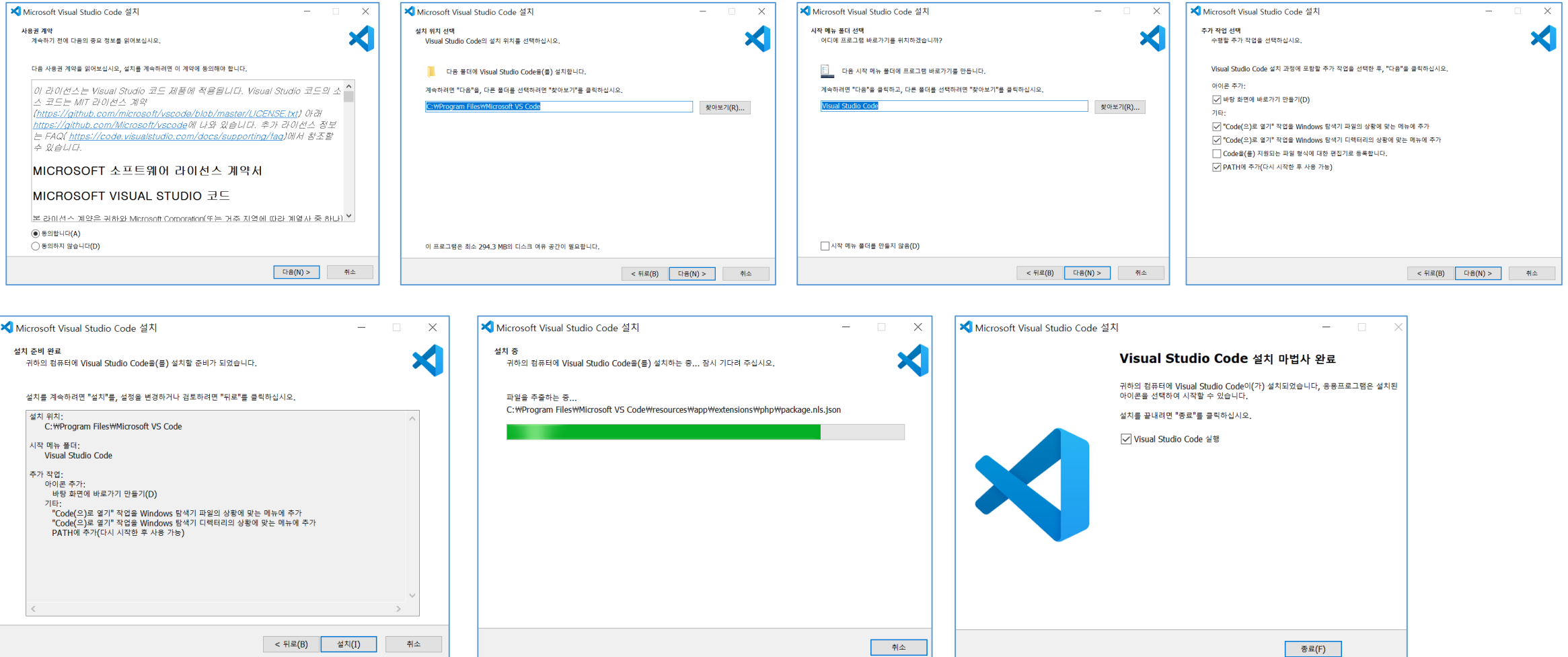
# 윈도우/애플 : vscode 개발환경 설치 및 실행 (1)

## ■ vscode 다운로드



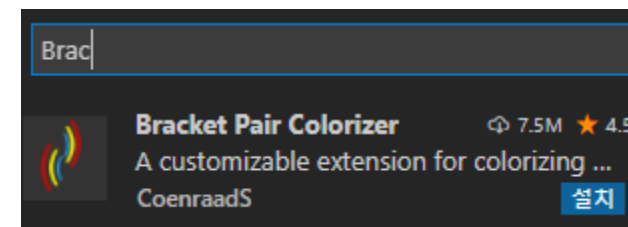
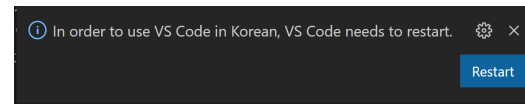
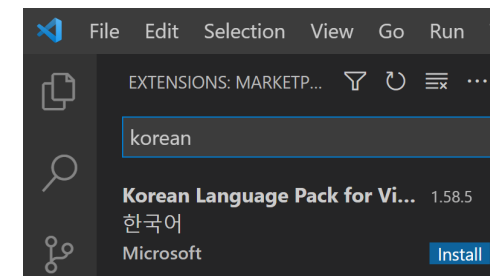
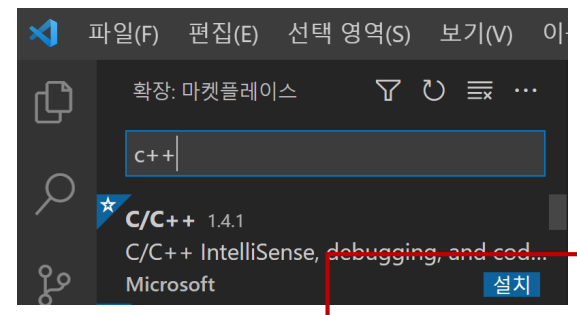
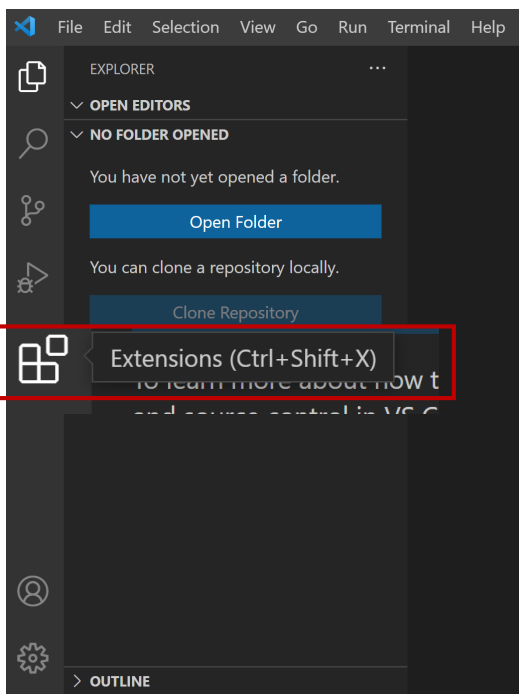
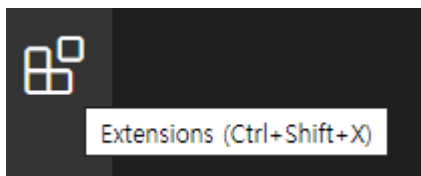
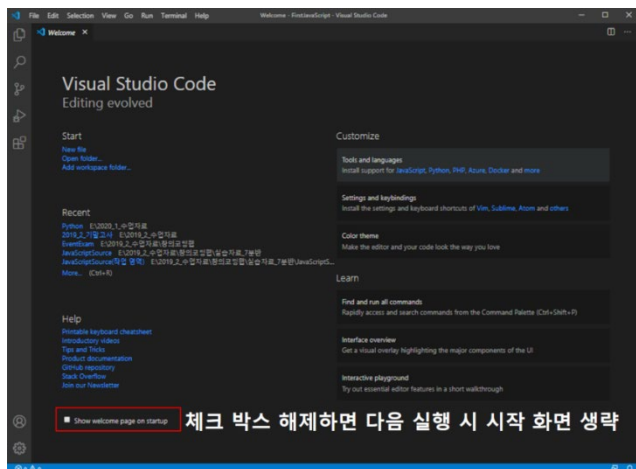
# 윈도우/애플 : vscode 개발환경 설치 및 실행 (2)

## ■ VSCode 설치



# 윈도우/애플 : vscode 개발환경 설치 및 실행 (3)

## ■ Extension 설치

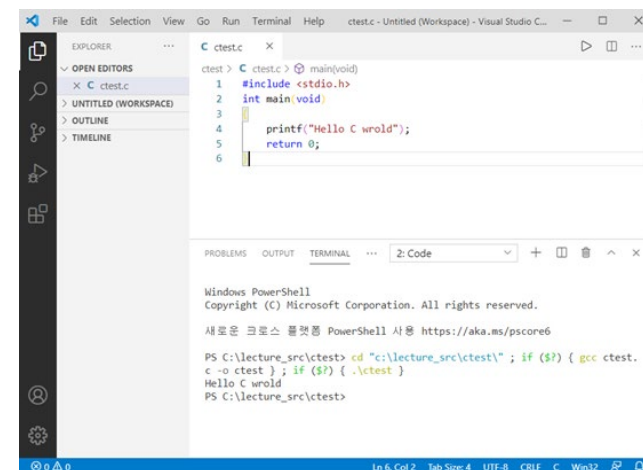
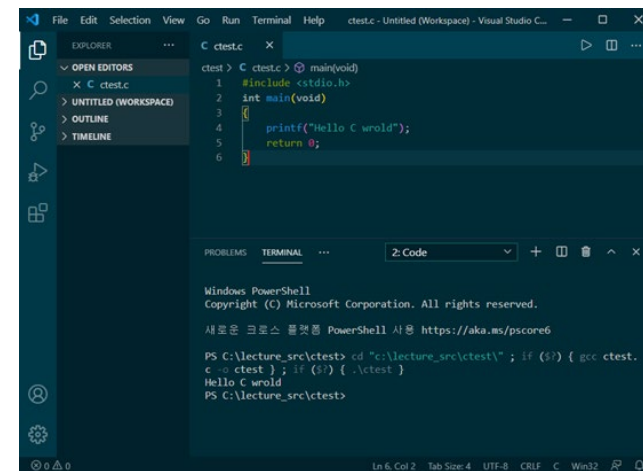
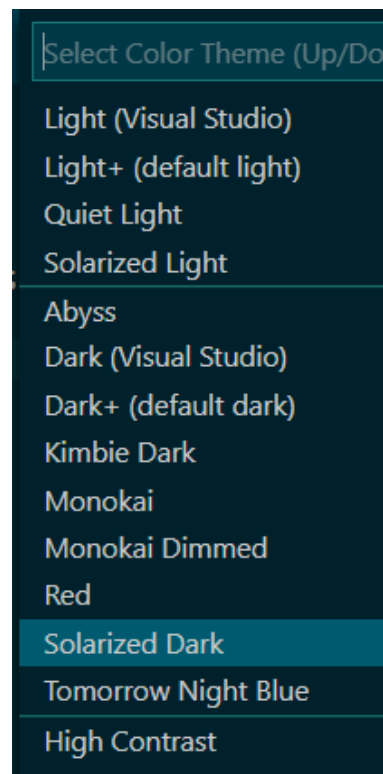
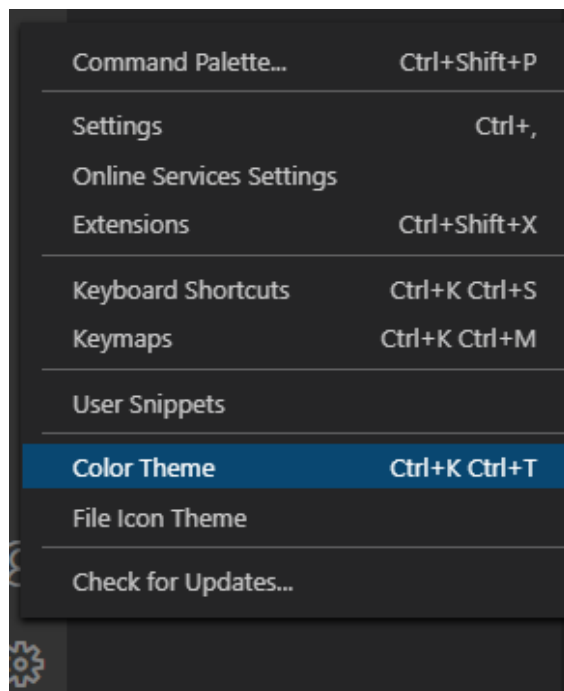
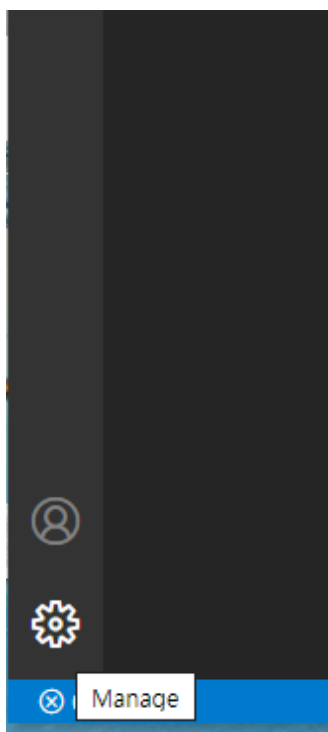


Bracket Pair Colorizer  
영역별로 괄호 색상 설정 – 사용자 프로그램 코드의 가독성을 좋게 함.



# 윈도우/애플 : vscode 개발환경 설치 및 실행 (4)

- 화면 배경 색상 변경
  - 왼쪽 아래에 톱니바퀴를 클릭하고 [Color Theme] 선택.



# 윈도우/애플 : vscode 개발환경 설치 및 실행 (5)

- 작업 폴더 생성 / c 또는 cpp 파일 생성

The image illustrates the process of setting up a development environment in Visual Studio Code (VS Code) for Windows or macOS. It shows the following steps:

- File Explorer:** A screenshot of the Windows File Explorer showing the 'lecture\_src' directory. The directory contains files like '.metadata', 'c\_src', and 'cpp\_src'.
- VS Code File Menu:** A screenshot of the VS Code 'File' menu with 'Folder Open...' highlighted, indicating the selection of the 'lecture\_src' directory as the workspace.
- VS Code Editor:** A screenshot of the VS Code editor showing the source code for 'cpptest.cpp'. The code includes `<iostream>` and defines a `main()` function that prints "Hello" and "첫 번째 맞보기입니다." (First meeting).
- VS Code Terminal:** A screenshot of the VS Code terminal window showing the execution of the program. The commands used are:

```
PS D:\lecture_src\c_src> gcc ctest.c -o ctest
PS D:\lecture_src\c_src> ./ctest
c program test
PS D:\lecture_src\c_src>
```

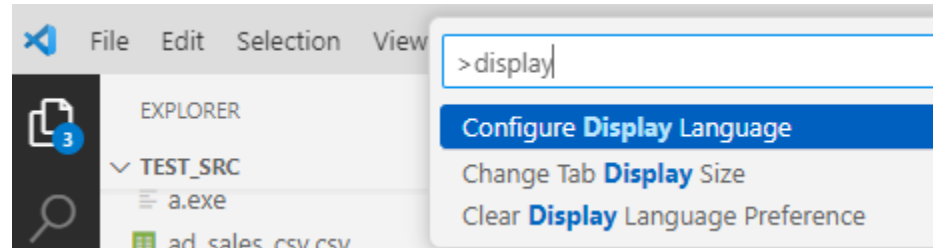
# vscode 글씨 크기 및 한글 설정 변경

- 글씨 크기 변경
  - [Ctrl]키와 [+]키, [Ctrl]키와 [-]키를 동시에 눌러 글씨 크기 조절.

- 한글 설정

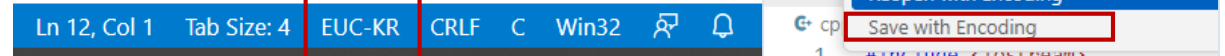
- 방법 1)

- <ctrl+ shift + p>를 동시에 누름
    - display 입력
    - Configure Display Language 선택
    - 한국어 [ko] 선택
    - Restart(vscode 재시작)



- 방법 2)

- vscode 오른쪽 아래 [EUC-KR] 부분이 "UTF-8"로 설정되어 있을 경우 한글이 깨져서 나타날 수 있음.
    - [EUC-KR] 또는 [UTF-8] 부분 클릭 해 <save with Encoding> [EUC-KR]로 설정 변경



- 방법 3)

- 소스 자체에서 한글이 깨지면 : Ctrl + Z

- 방법 4)

- 이 방법까지 진행하지는 않아도 됨
    - 설정 -> 세팅 -> 엔코딩 -> Auto Guess Encoding 체크
    - 제어판 -> 시계/국가 -> 국가또는지역 -> 관리자옵션 -> 시스템로컬변경 -> Beta 세계언어 체크

# 개념 확인 학습

---

1. 선행처리기란 무엇을 의미하는 것인가요?
2. 표준 라이브러리와 사용자 정의 라이브러리란 어떻게 다른가요?
3. 배운 내용 중 표준 라이브러리 함수에는 어떤 종류가 있었나요?
4. `int main(void)`의 의미는 무엇인가요?
5. 설명문(comment)의 역할은 무엇이며 어떻게 만드나요?
6. 컴파일이란 무엇인가요?
7. C 프로그램의 실행파일 생성과정을 설명하세요.

# 적용 확인 학습 & 응용 프로그래밍

---

- 여러분의 PC에 개발 환경을 설치 하시기 바랍니다.

# Q & A

---

- “C 프로그래밍을 시작하는 방법”에 대한 학습이 모두 끝났습니다.
- 모든 내용을 이해 하셨나요?
- 아직 이해가 안되는 내용이 있다면 다시 한번 복습하시기 바랍니다.
- 질문은 한림 SmartLEAD 쪽지 또는 e-mail 또는 전화상담을 이용하시기 바랍니다.



- 다음 시간에는 “C의 자료형과 입출력 방법”에 대해 알아보겠습니다.
- 수고하셨습니다.^^