

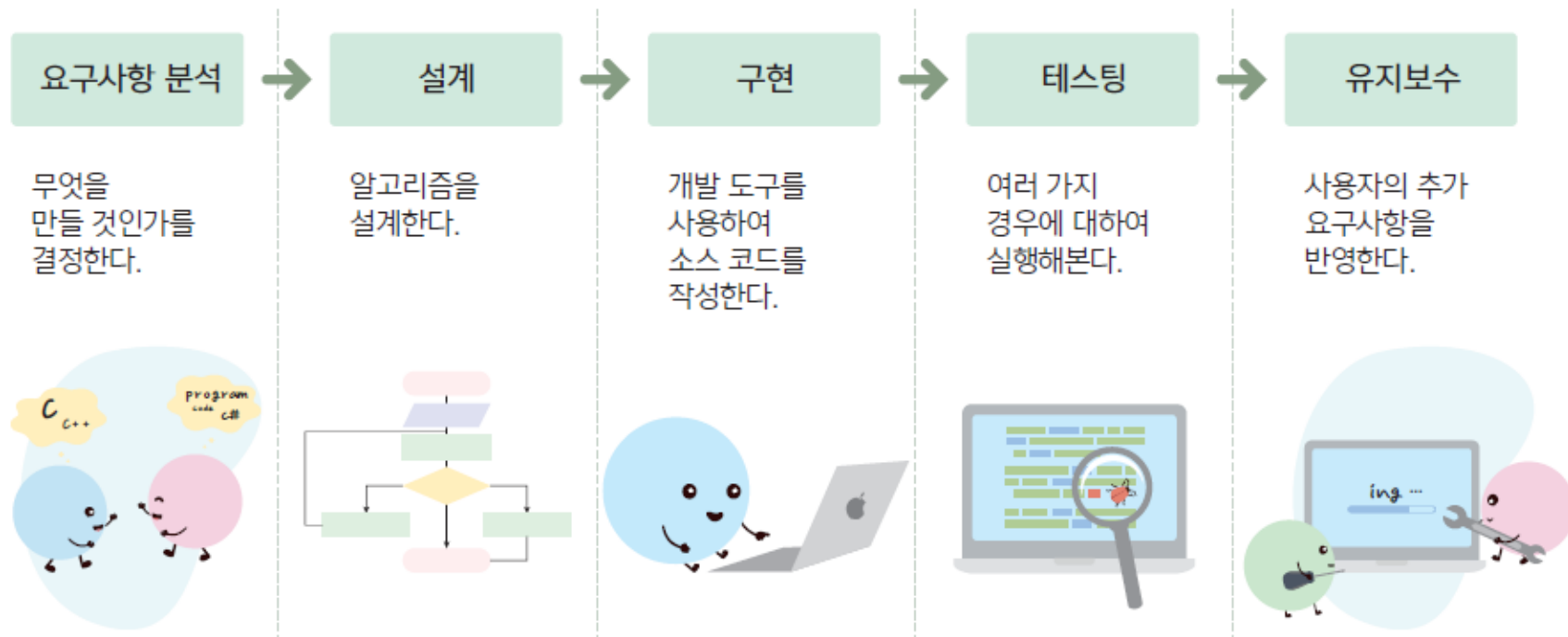
# 기초컴퓨터프로그래밍

## 제 2장 C언어 프로그래밍 작성방법

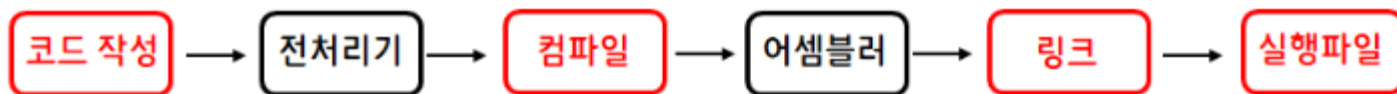
양우석 yws87874912@gmail.com



# 프로그램 개발 과정



실행파일은 코드 작성, 전처리기, 컴파일, 어셈블러, 링크 과정을 거쳐서 만들어지게 됩니다.



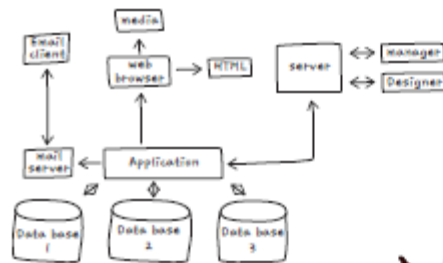
프로그램 빌드 과정

3주차에 자세히!



# 설계

- 문제를 해결하는 알고리즘을 개발하는 단계
- 순서도와 의사 코드를 도구로 사용
- 알고리즘은 프로그래밍 언어와는 무관
- 알고리즘은 원하는 결과를 얻기 위하여 밟아야 하는 단계에 집중적으로 초점을 맞추는 것



흠, 이제야 좀 이해가 되는군.





# 소스 작성

- 알고리즘의 각 단계를 프로그래밍 언어를 이용하여 기술
- 알고리즘을 프로그래밍 언어의 문법에 맞추어 기술한 것을 *소스 프로그램 (source program)*이라고 한다.
- 소스 프로그램은 주로 텍스트 에디터나 통합 개발 환경을 이용하여 작성한다.
- 소스 파일 이름: (예) test.c



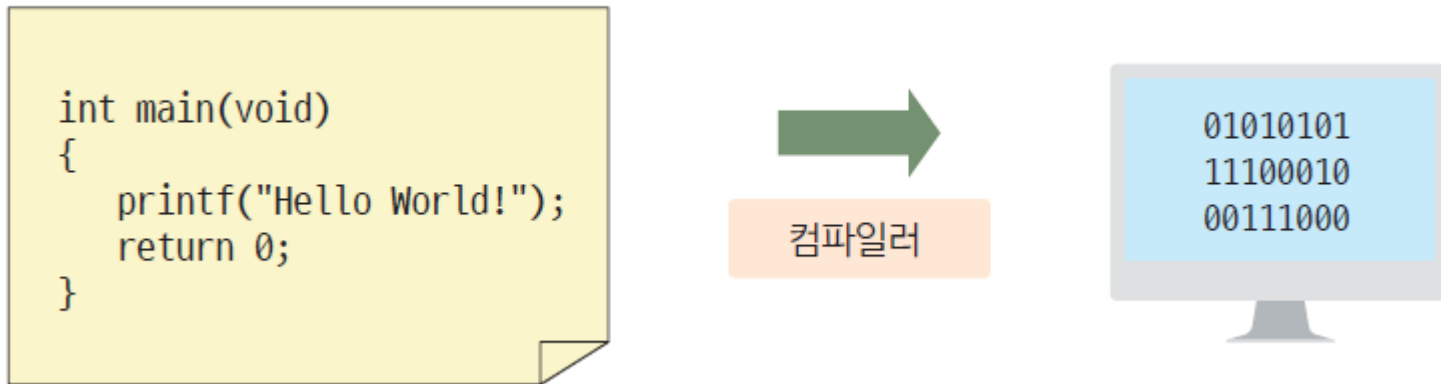
에디터

```
int main(void)
{
    printf("Hello World!");
    return 0;
}
```



# 컴파일

- 소스 프로그램을 오브젝트 파일로 변환하는 작업
- 오브젝트 파일 이름: (예) test.obj
- 오브젝트 파일: 컴파일러를 통해 생성된 바이너리 코드(이진코드)를 의미

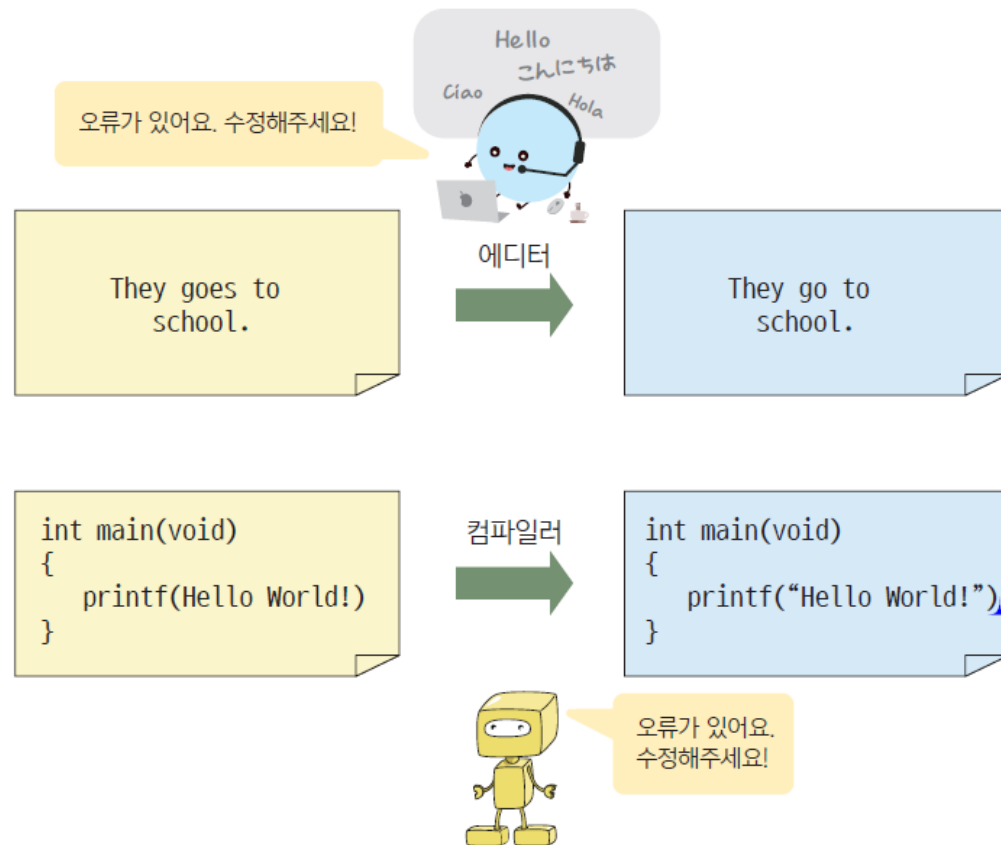


- 실제로는 소스코드 -> 어셈블리코드 -> 바이너리코드 로 변경
- 컴파일러가 어셈블러의 역할까지 해서 컴파일시 바로 바이너리코드 생성!



# 컴파일 오류

- 컴파일 오류(compile error): 문법 오류



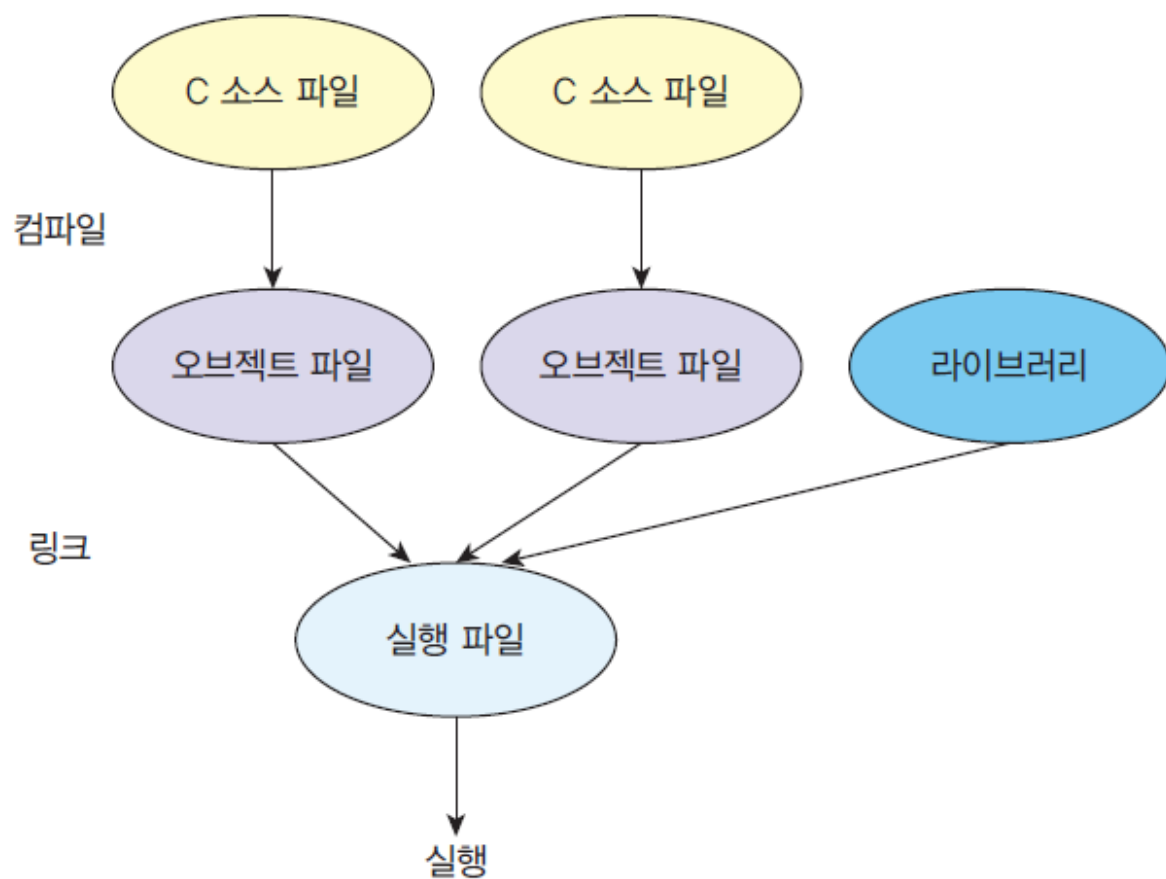


# 링크

- 컴파일된 목적 프로그램을 라이브러리와 연결하여 실행 프로그램을 작성하는 것
- 실행 파일 이름: (예) **test.exe**
- 실행파일이라는건 **cpu**에게 일을 시키기위해 바이너리 형태로 만들어진 파일
- *라이브러리(library)*: 프로그래머들이 많이 사용되는 기능을 미리 작성해 놓은 것
  - (예) 입출력 기능, 파일 처리, 수학 함수 계산
- 링크를 수행하는 프로그램을 *링커(linker)*라고 한다.



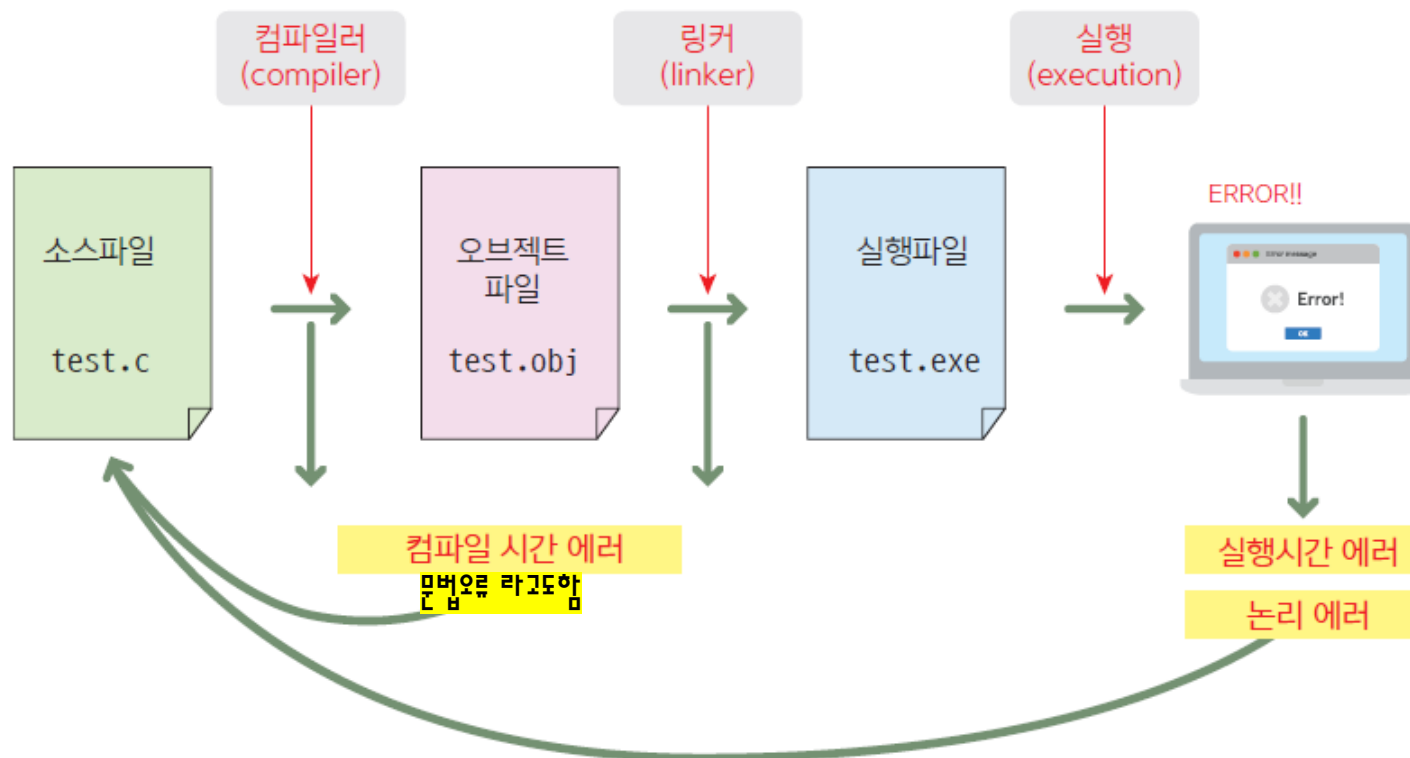
# 링크







# 실행 및 디버깅





# 실행 및 디버깅

- 실행 시간 오류(run time error):
  - 0으로 나누는 것
  - 잘못된 메모리 주소에 접근하는 것
- 논리 오류(logical error): 문법은 틀리지 않았으나 논리적으로 정확하지 않는 것
  - (예)

- ① 그릇1과 그릇2를 준비한다.
- ② 그릇1에 밀가루, 우유, 계란을 넣고 잘 섞는다.
- ③ 그릇2를 오븐에 넣고 30분 동안 350도로 굽는다.

실수로 빈그릇을 오븐에 넣는다면  
논리적인 오류입니다.





# 디버깅

- 소스에 존재하는 오류를 잡는 것





# 디버깅의 유래

- 1945년 마크 II 컴퓨터가 릴레이 장치에 낳아든 나방 때문에 고장을 일으켰고 이것을 “컴퓨터 버그(bug: 벌레)”
- 라고 불렀다. 여성 컴퓨터 과학자인 그레이스 호퍼가 나방을 채집해 기록에 남기고 이를 “디버깅(debugging)”작업이라고 보고하였다



9/2  
9/9

0800 Machine started  
1000 stopped - on fan ✓ { 1200 7.030 897.025  
1500 1300 HP-MC 2.13047695 7.037 896.795 count  
2.13047695 (5.000) 7.061 892.059 (1.4)  
2.13047695  
2.13047695  
2.13047695  
Relays 6-2 in 0.24 fault signal speed test  
14.00 test.

1100 Relays changed  
Started Cassio Tape (Sine check)  
1525 Started Utility Auto Test

1545 Relay 70 Panel F  
(motion relay)

1700 First actual case of bug being found.  
1700 Machine started.  
1700 closed room.



# 소프트웨어의 유지 보수

- 소프트웨어의 유지 보수가 필요한 이유
  1. 디버깅 후에도 버그가 남아 있을 수 있기 때문
  2. 소프트웨어가 개발된 다음에 사용자의 요구가 추가될 수 있기 때문
- 유지 보수 비용이 전체 비용의 **50%** 이상을 차지





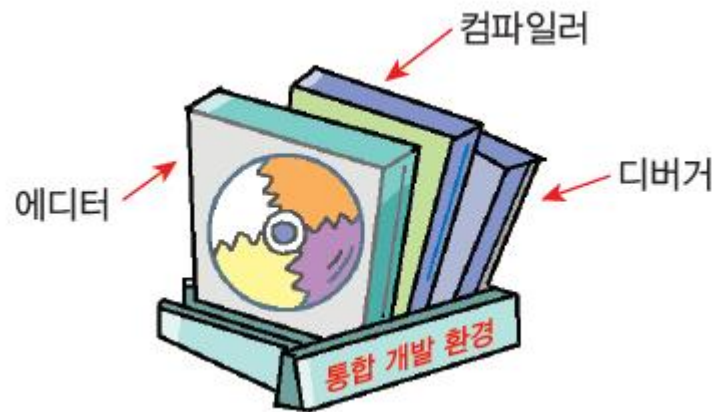
# 점점점

- ① 프로그램 개발 과정을 순서대로 정리하여 보자.
- ② 소스 파일의 이름으로 **test.txt**는 올바른가?
- ③ 소스 파일, 오브젝트 파일, 실행 파일의 차이점을 설명하여 보자.
- ④ 소스 파일이 **test.c**라면 컴파일 과정을 거친 후에 생성되는 오브젝트 파일과 실행 파일의 이름은 무엇인가?
- ⑤ 컴파일과 링크 과정을 거쳐서 실행 파일을 만든 다음에 소스 파일과 오브젝트 파일을 보관해야 하는가? 아니면 삭제하여도 되는가? 그 이유를 말하여 보자.
- ⑥ 디버깅(debugging)이란 무엇인가?
- ⑦ 왜 소프트웨어도 유지 보수가 필요한가?



# 통합 개발 환경

- 통합 개발 환경(IDE: integrated development environment) = 에디터 + 컴파일러 + 디버거





-





# 비주얼 스튜디오 버전

- 커뮤니티(Visual Studio Community) 버전은 “기업 외 응용 프로그램 빌드 개발자를 위한 완벽한 기능의 확장 가능한 무료 도구”이다.(본 수업 사용)
- 프로페셔널 버전(Visual Studio Professional)은 “개별 개발자 또는 소규모 팀을 위한 전문적인 개발자 도구 및 서비스”라고 되어 있다.
- 엔터프라이즈 버전(Visual Studio Enterprise)은 “고급 테스트 및 DevOps를 포함해서 어떠한 크기나 복잡한 프로젝트까지 개발 팀을 위한 고급 기능이 포함된 엔터프라이즈급 솔루션”라고 표시되어 있다.



# 비주얼 스튜디오 설치

애플사의 노트북을 가진 학생이라면 Visual Studio 2022 for Mac을 선택한다.

윈도우즈 사용자라면 이것을 클릭한다.

Visual Studio Tools 다운로드 - V x +

visualstudio.microsoft.com/ko/downloads/

Microsoft | Visual Studio 제품 다운로드 지원 교육자 액세스 무료 Visual Studio

Microsoft 검색 Sign in

## 다운로드

윈도우즈 사용자라면 이것을 클릭한다.

### Visual Studio 2022 | 📌

Windows에서 .NET 및 C++ 개발자를 위한 최고의 포괄적인 IDE입니다. 소스코드, 웹, 개발의 모든 단계를 향상시키고 개선할 수 있는 다양한 도구와 기능이 완벽하게 통합되어 있습니다.

Community  
강력한 IDE, 학생, 오픈 소스, 개인가 및 개인을 위한 무료 도구

Professional  
소규모 팀에 적합한 Professional IDE

Enterprise  
프론트엔드의 질을 위한 확장 가능한 통합 솔루션

미리 보기  
아직 기본 릴리스에는 없는 최신 기능을 미리 경험하세요.

자세한 정보 >  
릴리스 노트 >

릴리스 정보 > 버전 비교 > 오픈라인 설치 방법 >

무료 다운로드

평가판

무료 평가판

### Visual Studio 2022 for Mac | 📌

macOS에 기본 제공되는 .NET 개발자를 위한 포괄적인 IDE입니다. 웹, 클라우드, 모바일 및 게임 개발을 위한 최고 수준의 지원을 제공합니다.

미리 보기  
아직 주 릴리스에는 없는 최신 기능이 빠르게 액세스하세요.

자세한 정보 >  
릴리스 정보 >

자세한 정보: 라이선스 활성화 중

무료 다운로드

릴리스 정보 >

### Visual Studio Code | 📌

Windows, macOS 및 Linux에서 실행되는 통합 실행형 소스 코드 편집기입니다. Java 및 웹 개발자를 위한 최고의 선택이며, 거의 모든 프로그래밍 언어를 지원할 수 있는 수많은 확장 기능을 제공합니다.

Visual Studio Code를 사용하면 라이선스 및 개인정보처리방침에 동의하는 것입니다.

무료 다운로드

릴리스 정보 >



# 비주얼 스튜디오 설치

다운로드받은 파일을 더블  
클릭하여 설치한다.

Visual Studio를 다운로드해 주셔서 감사합니다.

다운로드가 곧 시작됩니다. 다운로드가 시작되지 않으면 [여기를 클릭하여 다시 시도하세요](#).

Visual Studio를 처음 사용하시나요? 새로운 개발 도구를 학습하는 것은 어려울 수 있습니다. 이 단계별 학습 여정을 통해 원하는 언어로 간단한 앱을 성공적으로 만들 수 있습니다. 시작하겠습니다.

### Visual Studio 설치

먼저 Visual Studio 다운로드를 열고 설치합니다. You can save installation time and disk space by 필요한 구성 요소만 선택하여 설치 시간과 디스크 공간을 절약할 수 있습니다. 필요에 따라 언제든지 나중에 더 많은 구성 요소를 점진적으로 추가할 수 있습니다.

### Visual Studio IDE 시작

모든 유형의 앱을 개발하거나 언어를 배우기 위해 IDE(Visual Studio 통합 개발 환경)에서 작업하게 됩니다. Visual Studio IDE는 코드 편집 외에도 그래픽 디자이너, 컴파일러, 코드 완성 도구, 소스 제어, 확장 기능 및 기타 많은 기능을 한 곳에 모읍니다. 이 짧은 비디오를 통해 IDE에 익숙해지고 IDE를 기본 작업에 사용하는 방법에 대해 알아보세요.

### 테마 등을 사용하여 나만의 테마 만들기

어두운 테마를 선호하세요? 장 레이아웃을 원하는 대로 변경하시겠습니까? 애플하는 바로 가기 키가 있나요? Visual Studio에서 사용할 수 있는 테마, 글꼴, 잘 레이아웃 및 사용자 지정 키보드 바로 가기 같은 몇몇 멋진 개인 설정 옵션을

VisualStudioSetup.exe

모두 표시



# 비주얼 스튜디오 설치

설치 — Visual Studio Community 2022 — 17.3.5

워크로드   개별 구성 요소   언어 팩   설치 위치

웹 및 클라우드 (4)

- ☐ ASP.NET 및 웹 개발  
Docker 지원이 포함된 ASP.NET Core, ASP.NET, HTML/JavaScript 및 컨테이너를 사용하여 웹 애플리케이션을 빌드...
- ☐ Python 개발  
Python에 대한 편집, 디버깅, 대화형 개발 및 소스 제어입니다.
- ☐ Azure 개발  
.NET 및 .NET Framework를 사용하여 클라우드 앱을 개발하고 리소스를 만들기 위한 Azure SDK, 도구 및 프로젝트...
- ☐ Node.js 개발  
비동기 이벤트 구동 JavaScript 런타임인 Node.js를 사용하여 확장 가능한 네트워크 애플리케이션을 빌드합니다.

데스크톱 및 모바일 (5)

- ☐ .NET Multi-Platform App UI 개발  
.NET MAUI와 함께 C#을 사용하여 단일 코드베이스에서 Android, iOS, Windows 및 Mac용 앱을 빌드합니다.
- ☒ C++를 사용한 데스크톱 개발  
MSVC, Clang, CMake 또는 MSBuild 등 선택한 도구를 사용하여 Windows용 최신 C++ 앱을 빌드합니다.
- ☐ .NET 데스크톱 개발  
.NET 및 .NET Framework와 함께 C#, Visual Basic 및 F#를 사용하여 WPF, Windows Forms 및 혼용 애플리케이션을...
- ☐ 유니버설 Windows 플랫폼 개발  
C#, VB 또는 C++(선택 사항)를 사용하여 유니버설 Windows 플랫폼용 애플리케이션을 만듭니다.

위치  
C:\Program Files\Microsoft Visual Studio\2022\Community   변경...

계속하면 선택한 Visual Studio 버전의 라이선스에 동의하게 됩니다. Microsoft는 Visual Studio와 함께 다른 소프트웨어를 다운로드할 수 있는 기능도 제공합니다. 이 소프트웨어는 [타사 고지 사항](#) 또는 해당 라이선스에 명시된 것처럼 별도로 라이선스가 부여됩니다. 계속하면 이러한 라이선스에도 동의하게 됩니다.

설치 세부 정보

▼ C++를 사용한 데스크톱 개발

- 포함됨
  - ✓ C++ 핵심 데스크톱 기능
- 선택 사항
  - ✓ MSVC v143 - VS 2022 C++ x64/x86 빌드...
  - ✓ 최신 v143 빌드 도구용 C++ ATL(x86 및 x64)
  - ✓ Windows 10 SDK(10.0.19041.0)
  - ✓ Just-in-Time 디버거
  - ✓ C++ 프로파일링 도구
  - ✓ Windows용 C++ CMake 도구
  - ✓ Test Adapter for Boost.Test
  - ✓ Test Adapter for Google Test
  - ✓ Live Share
  - ✓ IntelliCode
  - ✓ C++ AddressSanitizer
  - ☐ 최신 v143 빌드 도구용 C++ MFC(x86 및 x64)
  - ☐ Windows 앱 SDK C++ 샘플릿
  - ☐ Windows 11 SDK(10.0.22621.0)
  - ☐ v143 빌드 도구에 대한 C++/CLI 지원(최신)
  - ☐ v143 빌드 도구용 C++ 모듈(x64/x86)

필요한 전체 공간 6.6GB

다운로드하는 동안 설치   **설치**










# 비주얼 스튜디오 시작

## Visual Studio 2022

### 최근 파일 열기(R)

#### 오래됨

 <b>hello.sln</b>	2021-12-26 오후 8:55
D:\hello	
 <b>power.sln</b>	2021-12-26 오후 6:15
E:\document\book\c language\C언어(콘서트)(2021)\sources\chap14\power	
 <b>mycopy1.sln</b>	2021-12-26 오후 3:08
E:\document\book\c language\C언어(콘서트)(2021)\sources\chap14\mycopy1	
 <b>Project55.sln</b>	2021-12-23 오후 3:35
D:\tmp5\Project55	
 <b>Project14.sln</b>	2021-12-23 오전 10:50
C:\Users\kim\source\repos\Project14	
 <b>Project15.sln</b>	2021-12-22 오후 5:24
C:\Users\kim\source\repos\Project15	
 <b>김민성_19328_498095_20214093_김민성.sln</b>	2021-12-22 오후 5:14
D:\tmp9	

### 시작



#### 리포지토리 복제(C)

GitHub 또는 Azure DevOps 같은 온라인 리포지토리에서 코드 가져오기



#### 프로젝트 또는 솔루션 열기(P)

로컬 Visual Studio 프로젝트 또는 .sln 파일 열기



#### 로컬 폴더 열기(F)

폴더 내에서 탐색 및 코드 편집



#### 새 프로젝트 만들기(N)

시작하려면 코드 스캐폴딩과 함께 프로젝트 템플릿을 선택하세요.

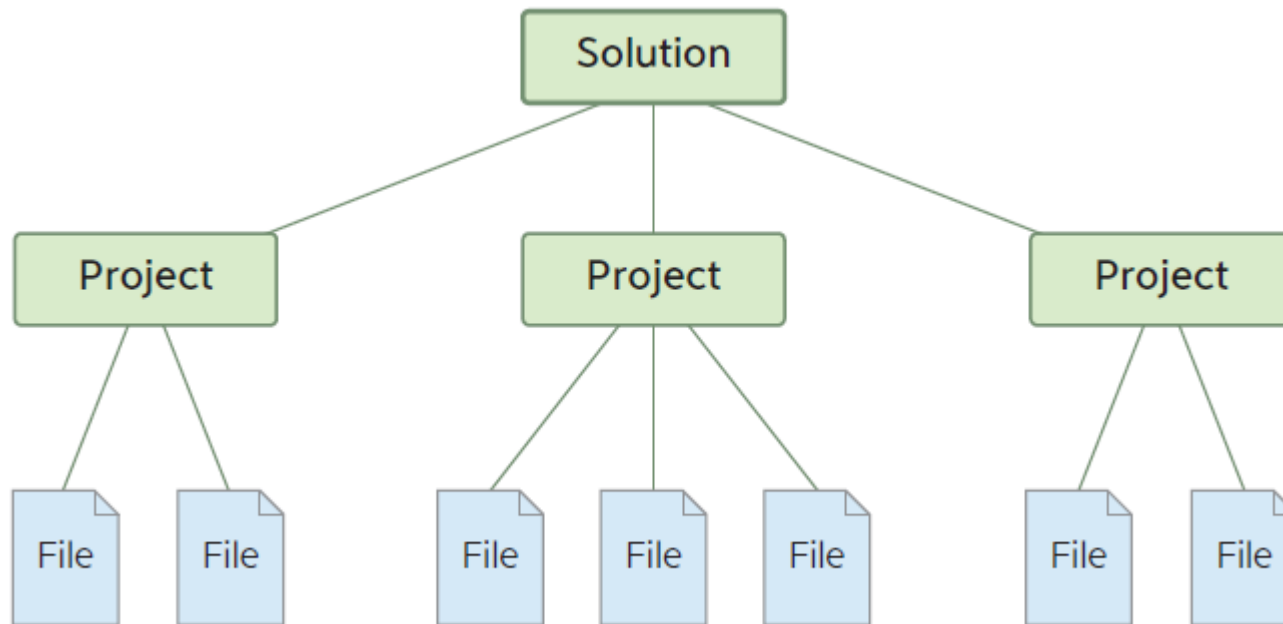
코드를 사용하지 않고 계속(W) →

클릭



# 솔루션과 프로젝트

- **솔루션(solution)**; 문제 해결에 필요한 프로젝트가 들어 있는 컨테이너
- **프로젝트(project)**: 하나의 실행 파일을 만드는데 필요한 여러 가지 항목들이 들어 있는 컨테이너







# 프로젝트 생성하기

## 새 프로젝트 만들기

### 최근 프로젝트 템플릿(R)

최근에 액세스한 템플릿 목록이 여기에 표시됩니다.

템플릿 검색(Alt+S)(S)

모든 언어(L)

모든 플랫폼(P)

모든 프로젝트 형식(T)



빈 프로젝트

Windows용 C++를 사용하여 처음부터 시작합니다. 시작 파일을 제공하지 않습니다.

C++

Windows

콘솔



콘솔 앱

Windows 터미널에서 코드를 실행합니다. 기본적으로 "Hello World"를 출력합니다.

C++

Windows

콘솔



CMake 프로젝트

.sln 또는 .vcxproj 파일에 종속되지 않은 최신 플랫폼 간 C++ 앱을 빌드하세요.

C++

Windows

Linux

콘솔



Windows 데스크톱 마법사

마법사를 사용하여 고유한 Windows 앱을 만드세요.

C++

Windows

데스크톱

콘솔

라이브러리



Windows 데스크톱 애플리케이션

Windows에서 실행되는 그래픽 사용자 인터페이스를 사용하는 애플리케이션용 프로젝트입니다.

C++

Windows

데스크톱



빈 솔루션

프로젝트 없는 빈 솔루션 만들기

(1) 클릭

(2) 클릭

다음(N)





# 프로젝트 생성하기

새 프로젝트 구성

빈 프로젝트   콘솔   C++   Windows

프로젝트 이름(N)

hello

위치(L)

C:\Users\kim\source\repos

솔루션 이름(M) ⓘ

hello

☒ 솔루션 및 프로젝트를 같은 디렉터리에 배치(D)

뒤로(B)   만들기(C)

(1) hello로 입력

(2) 원하는 디렉토리 선택

(3)



# 소스 파일 생성하기

Visual Studio IDE screenshot showing the 'File' menu with 'New Item' highlighted. A red box highlights the 'New Item' option, and a yellow arrow points to it. The 'Add' submenu is also visible, showing options like 'Add Class', 'Add File', etc.

File (F) 편집(E) 보기(V) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H) 검색 (Ctrl+Q) hello

Debug x86 로컬 Windows 디버거

서버 탐색기 도구 상자

솔루션 탐색기

솔루션 탐색기 검색(Ctrl+;) hello

솔루션 'hello' (1/1개 프로젝트)

- hello
  - 참조
    - 외부 종속성
    - 리소스 파일
    - 소스 파일

새 항목(W)... Ctrl+Shift+A 추가(D)

기존 항목(G)... Shift+Alt+A 클래스 마법사(Z)... Ctrl+Shift+X

새 필터(F) 여기까지 범위 지정(S)

클래스(C)... 새 솔루션 탐색기 뷰(N)

리소스(R)... 잘라내기(T) Ctrl+X

복사(Y) Ctrl+C

붙여넣기(P) Ctrl+V

삭제(D) Del

이름 바꾸기(M) F2

속성 속성(R) Alt+Enter

고급

SCC 파일	True
--------	------

일반

(이름)	소스 파일
고유 식별자	{4FC737F1-C7A5-4376-A066-2...}
필터	cpp;c;cc;cxx;c++;def;idl;hpj;t...

(이름)  
필터 이름을 지정합니다.

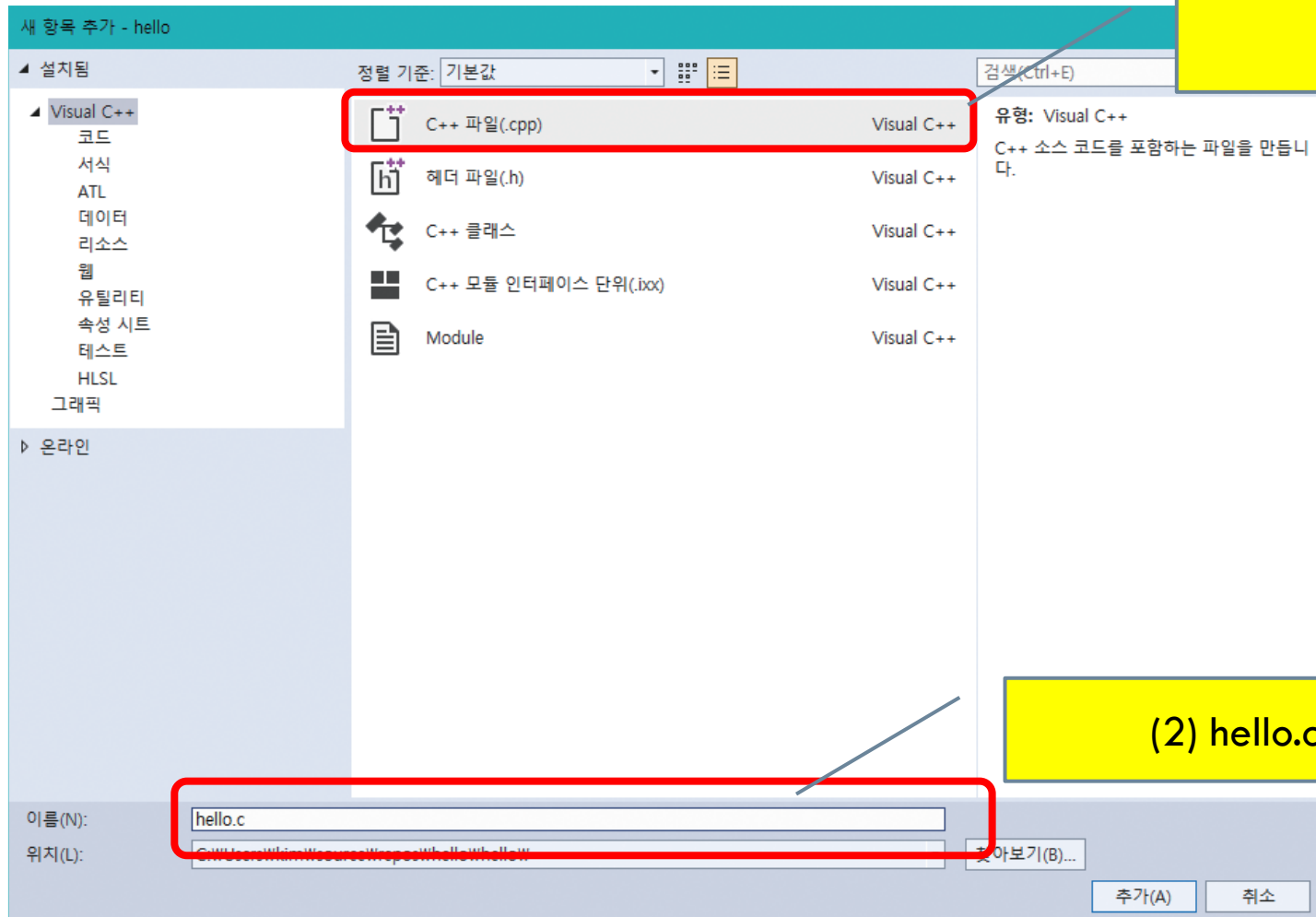
이 항목은 미리 보기를 지원하지 않습니다.

소스 제어에 추가



# 소스 파일 생성하기

(1) 클릭



(2) hello.c로 입력



# 프로그램 입력

철자에 주의하여 입력

```
파일(F)  편집(E)  보기(V)  Git(G)  프로젝트(P)  빌드(B)  디버그(D)  테스트(S)  분석(N)  도구(T)  확장(X)  창(W)  도움말(H)  검색
Debug  x86  로컬 Windows 디버거
hello.c*  X
hello  (전역 범위)
1  #include <stdio.h>
2
3  int main(void)
4  {
5      printf("Hello World!");
6      return;
7  }
```

133 % 문제자가 검색되지 않음 줄: 1 문자: 1 탭 CRLF

출력

출력 보기 선택(S): 빌드

다시 빌드 시작...

1>----- 모두 다시 빌드 시작: 프로젝트: hello, 구성: Debug Win32 -----

1>hello.c

1>hello.vcxproj -> C:\Users\k\source\repos\hello\Debug\hello.exe

===== 모두 다시 빌드: 성공 1, 실패 0, 생략 0 =====

솔루션 탐색기

솔루션 탐색기 검색(Ctrl+,)

솔루션 'hello' (1/1개 프로젝트)

- hello
  - 참조
  - 외부 종속성
  - 리소스 파일
  - 소스 파일
    - hello.c
  - 헤더 파일

솔루션 탐색기 Git 변경 내용

속성



기호는 정확하게 입력

문장의 끝에는 세미콜론



# 컴파일하기

The screenshot shows the Visual Studio IDE with the 'Build' menu open. A yellow lightning bolt points to the 'Build' menu. The 'Build' menu is highlighted with a red box, and the 'Build' option is also highlighted with a red box. The output window at the bottom shows the build process for the 'hello' project.

**Build Menu Options:**

- 솔루션 빌드(B) Ctrl+Shift+B
- 솔루션 다시 빌드(R)
- 솔루션 정리(C)
- 솔루션의 전체 프로그램 데이터베이스 파일 빌드
- 솔루션에서 코드 분석 실행(Y) Alt+F11
- hello 빌드(U) Ctrl+B
- hello 다시 빌드(E)
- hello 정리(N)
- hello에서 코드 분석 실행(A)
- 프로젝트만(J)
- 일괄 빌드(T)...
- 구성 관리자(O)...
- 컴파일(M) Ctrl+F7
- 파일에서 코드 분석 실행(F) Ctrl+Shift+Alt+F7

**Output Window:**

```
출력 보기 선택(S): 빌드
1>----- 모두 다시 빌드 시작: 프로젝트: hello, 구성: Debug Win32 -----
1>hello.c
1>hello.vcxproj -> C:\Users\wkim\source\repos\hello\Debug\hello.exe
===== 모두 다시 빌드: 성공 1, 실패 0, 생략 0 =====
```

모두 다시 빌드했습니다.



# 프로그램 실행 하기

Microsoft Visual Studio 디버깅 콘솔

선택한다.

프로그램의 출력

```
#include <stdio.h>

int main(void)
{
    printf("Hello World!");
    return 0;
}
```

출력 보기 선택

발드 시작...

1>----- 발드

1>hello.c

1>hello.vcxproj

===== 발드

C:\Users\kkim\source\repos\hello\hello\Debug\hello.exe (프로세스 2280개)이(가) 종료되었습니다(코드: 0개).

이 창을 닫으려면 아무 키나 누르세요...



## 작업 점검



- ① 에디터, 컴파일러, 링커, 실행, 디버깅 등의 기능이 하나의 프로그램 안에 들어 있는 것을 무엇이라고 하는가?
- ② 비주얼 스튜디오에서 새로운 프로젝트를 생성하는 메뉴는 무엇인가?
- ③ 비주얼 스튜디오에서 프로젝트에 속하는 소스 파일을 컴파일하여 실행 파일을 생성하는 메뉴는 무엇인가?
- ④ C언어에서는 대문자와 소문자를 구별하는가?
- ⑤ 비주얼 스튜디오를 이용하여서 **sample.c**라는 소스 파일을 컴파일하였을 때 생성되는 파일들은 무엇인가?
- ⑥ 비주얼 스튜디오를 사용하여 소스 프로그램을 편집하는 경우, 메모장 같은 다른 텍스트 에디터를 사용하여도 되는가?





# 첫 번째 프로그램의 설명

*hello.c*

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    printf("Hello World!");
```

```
    return 0;
```

```
}
```



Hello World!



# 프로그램 == 작업 지시서

\*화면에 "Hello World!"를  
표시한다.

작업 지시서

```
#include <stdio.h>


int main(void)
{
    printf("Hello World!");
    return 0;
}
```

프로그램



# 작업을 적어주는 위치

```
#include <stdio.h>

int main(void)
{
    
    return 0;
}
```

여기다가 원하는 작업을  
수행하는 문장을 적어  
준다.

프로그램



# 간략한 소스 설명

```
#include <stdio.h>

int main(void)
{
    printf("Hello World!");
    return 0;
}
```

• 헤더 파일을 포함한다.

• 메인 함수 시작

• 화면에 "Hello World!" 를 출력

• 외부로 0값을 반환

• 메인 함수 종료

프로그램



# 헤더 파일 포함

- `#include`는 소스 코드 안에 특정 파일을 현재의 위치에 포함

- 주의!: 전처리기 지시자 문장 끝에는 세미콜론(;)을 붙이면 안 된다.

`#include <stdio.h>`

- 헤더 파일(header file): 컴파일러가 필요로 하는 정보를 가지고 있는 파일
- stdio.h: standard input output header file

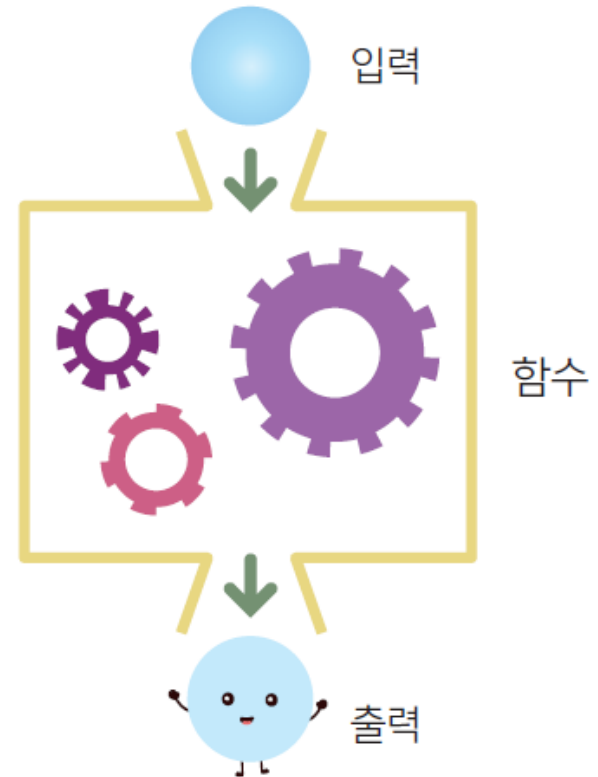


# 함수

- 함수(function): 특정한 작업을 수행하기 위하여 작성된 독립적인 코드
- (참고) 수학적 함수

$$y = x^2 + 1$$

- 프로그램 = 함수의 집합





# 함수의 간단한 설명

함수의 출력 타입

int

함수의 이름

main

함수의 입력 타입

(void)

{

함수의 시작

```
printf("Hello World");  
return 0;
```

함수의 몸체

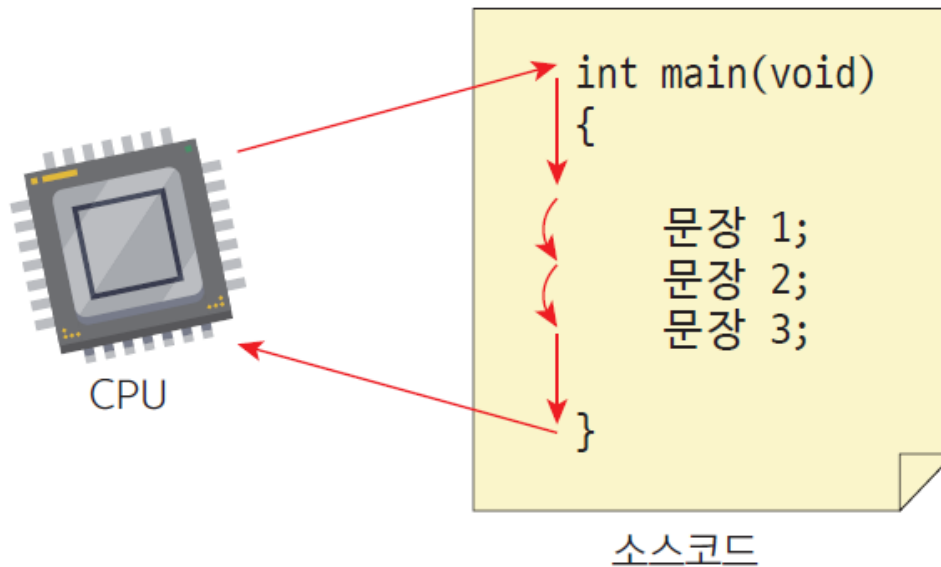
}

함수의 끝



# 문장(명령문)

- 함수는 여러 개의 문장으로 이루어진다.
- 문장들은 순차적으로 실행된다.
- 문장의 끝에는 반드시 ;이 있어야 한다.



소스 코드의 문장들은  
기본적으로 차례대로  
실행됩니다.







# printf() 호출

- printf()는 컴파일러가 제공하는 함수로서 출력을 담당한다

printf("Hello World!");



Hello World!

- 큰따옴표 안의 문자열이 화면에 출력된다.



# 함수의 반환값

- return은 함수의 결과값을 외부로 반환

return 0;

- 반환값은 0



## 장간 점검

1. 문장의 끝에 추가하여야 하는 기호는?
2. C프로그램에 반드시 있어야 하는 함수는?
3. printf()가 하는 기능은 무엇인가?





# 응 프로그램 #1

- 다음과 같은 출력을 가지는 프로그램을 제작하여 보자.





# 첫 번째 버전

- 문장들은 순차적으로 실행된다는 사실 이용

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    printf("Hello World!");
```

```
    printf("Kim ChulSoo");
```

```
    return 0;
```

```
}
```

2개의 문장은 순차적으로  
실행된다

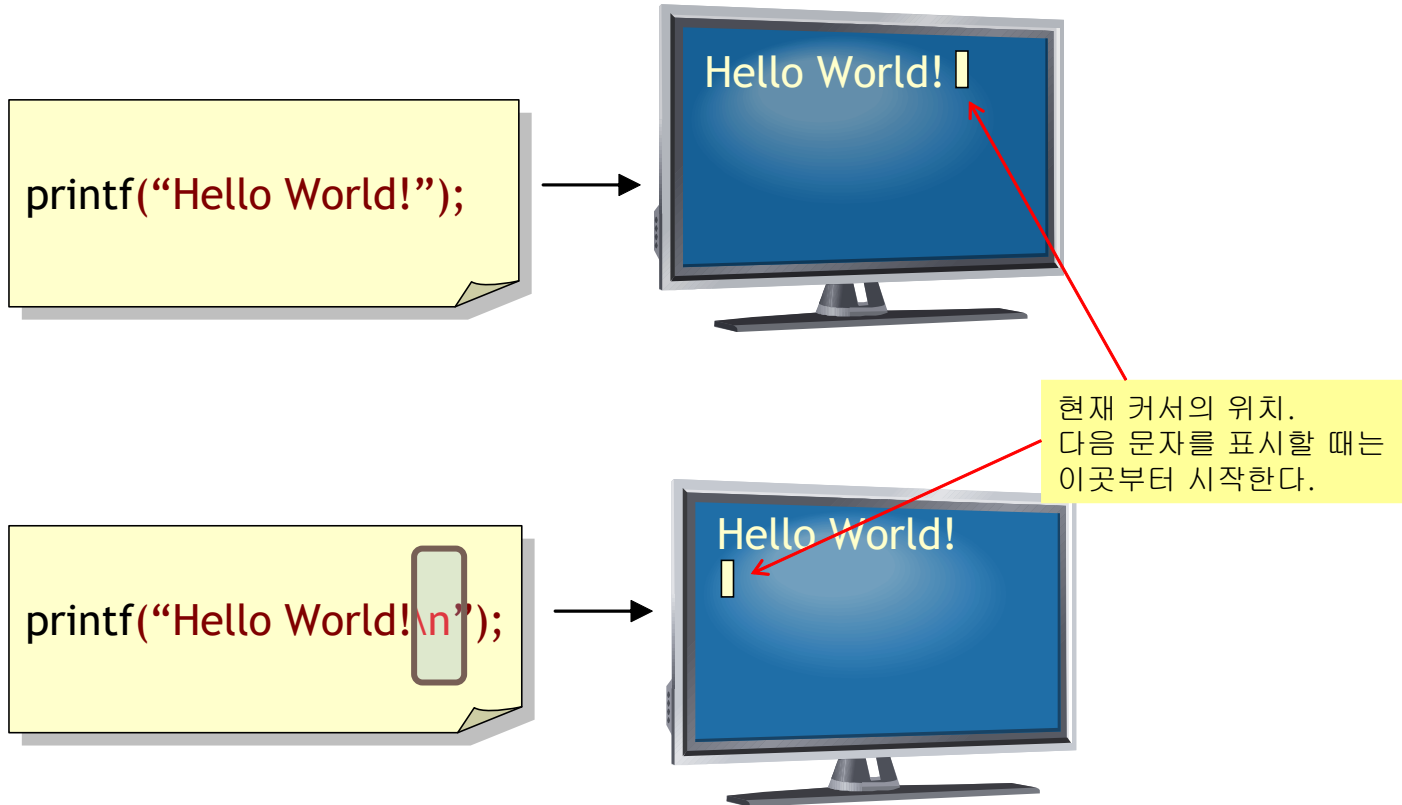


Hello World! Kim ChulSoo



# 줄바꿈 문자

- 줄바꿈 문자인 `\n`은 화면에서 커서는 다음줄로 이동하게 한다.





# 변경된 프로그램

- 줄바꿈 문자를 추가하면 우리가 원하던 결과가 된다.

```
#include <stdio.h>

int main(void)
{

    printf("Hello World!\n");
    printf("Kim ChulSoo\n");

    return 0;
}
```





# Lab: 간단한 계산을 해보자

- 덧셈과 뺄셈, 곱셈, 나눗셈 계산을 하는 프로그램을 작성해보자.

```
결과값=5  
결과값=-1  
결과값=6  
결과값=5
```





# Solution

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    printf("결과값=%d\n", 2 + 3);
```

```
    printf("결과값=%d\n", 2 - 3);
```

```
    printf("결과값=%d\n", 2 * 3);
```

```
    printf("결과값=%d\n", 2 / 3);
```

```
    return 0;
```

```
}
```

파이썬의 문자열 포매팅과 다르게 문자열 뒤 %로 구분 하는게 아님

파이썬 예시 )print("2+3 = %d" %2+3)

printf : print 함수에 format을 사용하겠다 라는 의미로 생각!



# Lab: 구구단을 출력해보자.

- 구구단 중에서 9단의 일부를 출력하는 프로그램을 작성해보자.

```
9 X 1 = 9  
9 X 2 = 18  
9 X 3 = 27  
9 X 4 = 36  
9 X 5 = 45
```



# Solution

```
#include <stdio.h>

int main(void)
{
    printf("9 X 1 = %d\n", 9*1);
    printf("9 X 2 = %d\n", 9*2);
    printf("9 X 3 = %d\n", 9*3);
    printf("9 X 4 = %d\n", 9*4);
    printf("9 X 5 = %d\n", 9*5);
    return 0;
}
```



9단 전체를 출력하도록 코드를 수정해보자.



# 오류 수정 및 디버깅

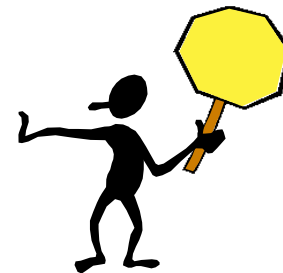
- 컴파일이나 실행 시에 오류가 발생할 수 있다.
- 에러와 경고
  - 에러(error): 심각한 오류 -> 컴파일, 링크 불가능
  - 경고(warning): 경미한 오류 -> 컴파일, 링크 가능 / 잠재적 문제 발생가능





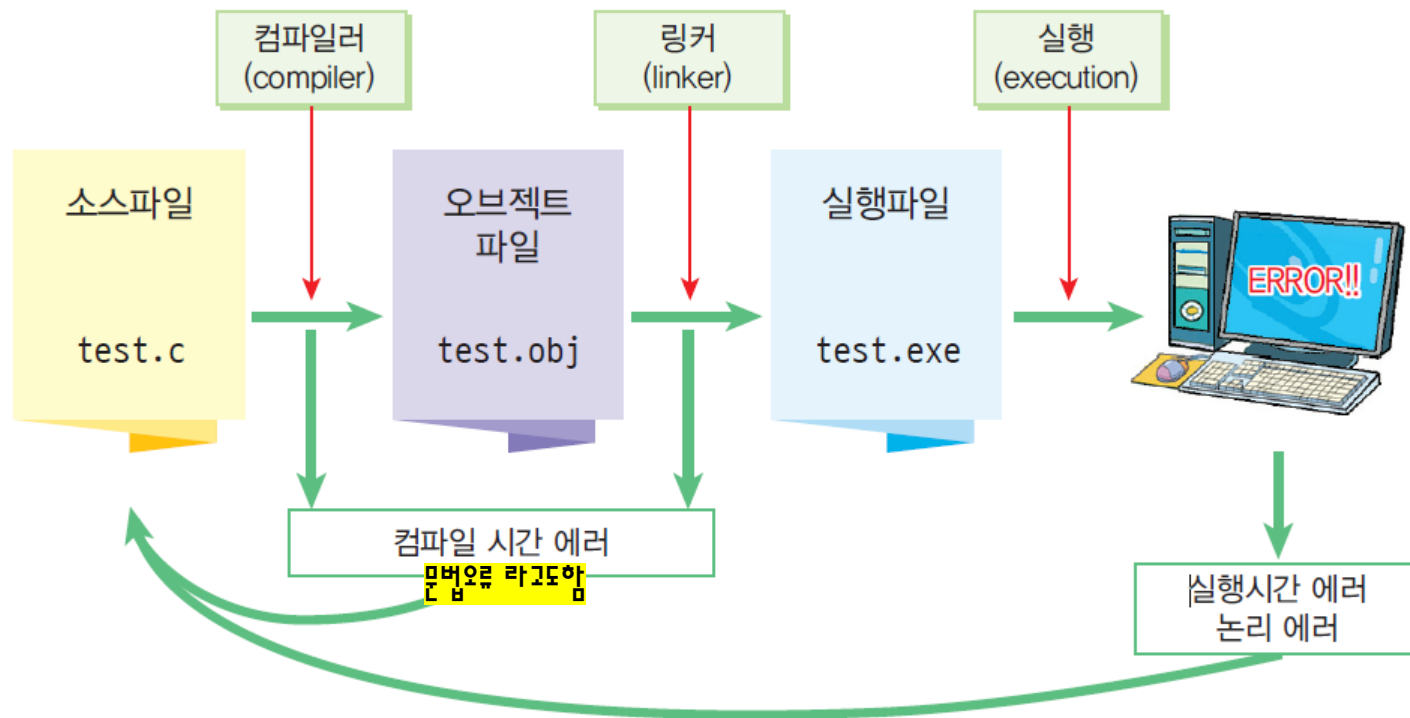
# 오류의 종류

- 오류의 종류
  - 컴파일 시간 오류: 대부분 문법적인 오류
  - 실행 시간 오류: 실행 중에 0으로 나누는 연산 같은 오류
  - 논리 오류: 논리적으로 잘못되어서 결과가 의도했던 대로 나오지 않는 오류





# 오류 수정 과정





# 오류 #1

add - Microsoft Visual Studio

빠른 실행(Ctrl+Q)

파일(F) 편집(E) 보기(V) 프로젝트(P) 빌드(B) 디버깅(D) 팀(M) 도구(T) 테스트(S) 분석(N) 창(W) 도움말(H) 로그인

Debug x86 로컬 Windows 디버거

error1.c\* x

add (전역 범위) main(void)

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     printf("Hello World!");
6     return 0;
7 }
```

;이 생략되었다.

retrun 앞에 ;를 빠뜨렸다는 의미이다.

오류 목록

전체 솔루션 2 오류 0 경고 0 메시지 빌드 + IntelliSense 검색 오류 목록

코드	설명	프로젝트	파일	줄	비표시 오류(Suppr...)
C2143	';'가 필요합니다. 구문 오류: ';'이(가) 'return' 앞에 없습니다.	add	error1.c	6	
		add	error1.c	6	

오류가 발견된 소스 파일

오류가 발견된 줄번호

솔루션 탐색기

솔루션 탐색기 검색(Ctrl+;)

솔루션 'add' (1개 프로젝트)

- add
  - 참조
  - 외부 종속성
  - 리소스 파일
  - 소스 파일
    - error1.c
  - 헤더 파일

속성

main VCodeFunction

일 C++

(Name)	main
File	d:\sources\Wch
FullName	main

C++

준비 줄: 7 열: 2 문자: 2 INS



# 오류 #2

add - Microsoft Visual Studio

빠른 실행(Ctrl+Q)

파일(F) 편집(E) 보기(V) 프로젝트(P) 빌드(B) 디버그(D) 팀(M) 도구(T) 테스트(S) 분석(N) 창(W) 도움말(H) 로그인

Debug x86 로컬 Windows 디버거

error3.c

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     printf("Hello World!");
6     return 0;
7 }
```

문자열을 표시할 때, "와 "을 빠뜨렸다.

오류 목록

전체 솔루션 5 오류 2 경고 0 메시지 빌드 + IntelliSense 검색 오류 목록

코드	설명	프로젝트	파일	줄	비표시 오류(Suppr...)
abc	식별자 "Hello"이(가) 정의되어 있지 않습니다.	add	error3.c	5	
abc	')'가 필요합니다.	add	error3.c	5	
C2065	'Hello': 선언되지 않은 식별자입니다.	add	error3.c	5	
C4047	'함수': 'const char *const '의 간접 참조 수준이 'int'과(와) 다릅니다.	add	error3.c	5	

오류 목록 출력

솔루션 탐색기

솔루션 탐색기 검색(Ctrl+;)

솔루션 'add' (1개 프로젝트)

- add
  - 참조
  - 외부 종속성
  - 리소스 파일
  - 소스 파일
    - error3.c
  - 헤더 파일

솔루션 탐색... 팀 탐색기 클래스 뷰

속성

main VCCodeFunction

C++

속성	값
(Name)	main
File	d:\sources\Wch
FullName	main

C++

줄: 6 알: 14 문자: 11 INS

↑ 게시





# 오류 #3

add - Microsoft Visual Studio

빠른 실행(Ctrl+Q)

파일(F) 편집(E) 보기(V) 프로젝트(P) 빌드(B) 디버그(D) 팀(M) 도구(T) 테스트(S) 분석(N) 창(W) 도움말(H) 로그인

Debug x86 로컬 Windows 디버거

error3.c

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     print("Hello World!");
6     return 0;
7 }
```

printf이어야 한다.

함수를 찾지 못했음.

오류 목록

전체 솔루션 2 오류 1 경고 0 메시지 빌드 + IntelliSense 검색 오류 목록

코드	설명	프로젝트	파일	줄	비표시 오류(Suppr...)
C4013	'print'이(가) 정의되지 않았습니다. extern은 int형을 반환하는 add 것으로 간주합니다.	add	error3.c	5	
LNK2019	_print 외부 기호(참조 위치: _main 함수)에서 확인하지 못했습니다.	add	error3.obj	1	
LNK1120	1개의 확인할 수 없는 외부 참조입니다.	add	add.exe	1	

오류 목록 출력

속성

main VCodeFunction

C++

(Name)	main
File	d:\sources\Wch
FullName	main

C++

빌드 실패

↑ 게시



## 노리 오류

- 다음과 같은 출력을 가지는 프로그램을 작성하여 보자.





# 논리 오류가 존재하는 프로그램

```
#include <stdio.h>

int main(void)
{
    printf("마트에서 사올 품목");
    printf("=====");
    printf("사과, 우유, 빵");
    printf("=====");
    return 0;
}
```

줄이 바뀌지  
않았음!

마트에서 사올 품목=====사과,  
우유, 빵=====



# 논리 오류가 수정된 프로그램

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    printf("마트에서 사올 품목\n");
```

```
    printf("=====\n");
```

```
    printf("사과, 우유, 빵\n");
```

```
    printf("=====\n");
```

```
    return 0;
```

```
}
```

논리 오류  
수정!!



마트에서 사올 품목

=====

사과, 우유, 빵

=====



## 그래서 논리오류가 뭔데?

간단히 말해서 코드가 정상적으로 컴파일 및 실행이 되지만,  
개발자(작성자)가 의도한 기능 또는 동작을 하지 않는 것을 의미한다.

사실상 가장 수정하기 어려운 부분이라 생각.

-> 개발자의 이도에 따라 오류 여부가 갈릴 수 있기 때문.



# 디버깅

- 디버깅: 논리 오류를 찾는 과정

아무래도 이 부분이  
수상해..



프로그램의 실행결과

논리 에러를 발견하는 것은  
수사관이 범죄 흔적을 이용하여  
범인을 찾는 것과 같습니다.





# 디버거(debugger)

- 프로그램을 한 문장씩 실행하면서 오류의 원인을 찾는 도구

hello (디버깅) - Microsoft Visual Studio

파일(F) 편집(E) 보기(V) 프로젝트(P) 빌드(B) 디버거(D) 팀(M) 도구(T) 테스트(S) 분석(N) 창(W) 도움말(H)

프로세스: [144720] hello.exe

hello.c x

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     printf("Hello World!");
6     printf("Good Mornin");
7     return 0;
8 }
```

현재 실행되고 있는 위치

디버거 메뉴:

- 장(W)
- 그래픽(C)
- ▶ 계속(C) F5
- 모두 중단(K) Ctrl+Alt+Break
- 디버깅 중지(E) Shift+F5
- 모두 분리(L)
- 모두 종료(M)
- 다시 시작(R) Ctrl+Shift+F5
- 코드 변경 내용 적용(A) Alt+F10
- 성능 프로파일러(F...) Alt+F2
- 프로세스에 연결(P...) Ctrl+Alt+P
- 기타 디버거 대상(H) ▶
- 프로파일러
- ▶ 한 단계씩 코드 실행(I) F11
- 프로시저 단위 실행(O) F10
- 프로시저 나가기(T) Shift+F11
- 간략한 조사식(Q...) Shift+F9
- 중단점 설정/해제(G) F9
- 새 중단점(B) ▶
- 모든 중단점 삭제(A) Ctrl+Shift+F9
- 모든 DataTips 지우기(A)
- DataTips 내보내기(X)...
- DataTips 가져오기(I)...
- 다른 이름으로 덤프 저장(V)...
- 옵션(O)...
- hello 속성...

진단 도구

진단 및 세션: 0 초(178 ms이(가) 선택됨)

CPU (모든 프로세서에 대한 비율(%))

요약 이벤트 메모리 사용량 CPU 사용량

이벤트

이벤트 표시(1/1)

메모리 사용량

스냅샷 만들기

스택

hello.exe!main() 줄 4

부 코드

프레임은 슬라브지 않거나 누락되었거나 kernel32.dll에 대해 로드된 기호가 없음

중단점 예외 설정 명령 창 직접 실행 창 출력

준비

소스 제어에 추가



## 디버거의 명령어 정의

- F5 (Go): 실행
- F10 (Step Over): 한 문장씩 실행(함수도 하나의 문장 취급)
- F11 (Step Into): 한 문장씩 실행(함수 안으로 진입)
- F9 (Breakpoint): 현재 문장에 중단점을 설정





# 디버거의 실행 과정

F10을 누를 때마다  
한 문장씩 실행된다.

F10을 누를 때마다  
한 문장씩 실행된다.

```
hello.c x
(전역 범위)
- main(void)
/* 첫 번째 프로그램 */
#include <stdio.h>

int main(void)
{
    printf("Hello World!\n");
    printf("Good Morning\n");
    return 0;
}
```

100 %

C:\Users\chun#documents#visual studio 2010#Projects#hello#Debug#hello.exe

Hello World!

Good Morning



## 작업 점검

1. 프로그램을 편집하여 컴파일, 링크를 한 다음, 실행시켰는데 자신이 기대한 대로 결과가 나오지 않았다. 이때는 어떻게 하여야 하는가?
2. 비교적 경미한 오류를 무엇이라고 하는가?





# 참고사항: 디버거 기능

Start Debug→Go( <b>F5</b> )	디버깅 모드로 프로그램을 실행한다.
Restart( <b>Ctrl</b> + <b>Shift</b> + <b>F5</b> )	프로그램을 재실행한다.
Stop Debugging( <b>Shift</b> + <b>F5</b> )	디버깅을 중단한다.
Break Execution	프로그램 실행 중에 이 버튼을 누르면 현재 위치에서 실행이 중단된다.
Step Into( <b>F11</b> )	하나의 문장을 실행한다. 만약 문장에 함수 호출이 있으면 그 함수로 들어간다.
Step Over( <b>F10</b> )	하나의 문장을 실행한다. 만약 문장에 함수 호출이 있어도 함수도 들어가지 않는다.
Step Out( <b>Shift</b> + <b>F11</b> )	현재 실행중인 함수를 빠져 나온다.
Run to Cursor( <b>Ctrl</b> + <b>F11</b> )	현재 커서 위치까지 실행한다.
Quick Watch( <b>Shift</b> + <b>F9</b> )	현재 사용 중인 변수를 입력하여 그 변수의 값을 볼 수 있다.
Watch	보고 싶은 변수를 입력한다.
Variables	현재 사용되는 변수값이 표시된다.
Registers	CPU 안의 레지스터의 상태를 보여준다.
Memory	메모리를 16진수와 문자열로 표시한다.
Call Stack	함수의 호출 순서를 볼 수 있다.
Disassembly	변환된 어셈블리 코드를 보여준다.
<b>F9</b>	현재 위치에 중단점을 설정한다. 디버거가 중단점을 만나면 실행을 중지한다.



# 과제

- 아래 코드의 오류를 수정해보자!
- 올바르게 수정하고 **LMS**에 올라온 양식에 맞춰 업로드

```
#include <stdio.h>

int Main(void)
(
    printf(안녕하세요?\n);
    printf(이번 코드에는 많은 오류가 있다네요\n)
    print(제가 다 고쳐보겠습니다.\n);
    return 0;
)
```



# 과제



## Solution bug.c

```
1 #include <stdio.h>
2
3 int Main(void)
4 (
5     printf(안녕하세요? \n);
6     printf(이번 코드에는 많은 오류가 있다네요 \n);
7     print(제가 다 고쳐보겠습니다.\n);
8     return 0;
9 )
```

main

(가 아니라 {이어야 한다.

문장의 끝에는 ;가 있어야 한다.

문자열에는 따옴표를 붙인다.

print가 아니고 printf 이어야 한다.



# Q & A

