

개정3판

Visual
Studio
2017

쉽게 풀어쓴

C언어 EXPRESS



천인국 지음

CHAPTER 2: 프로그램 작성 과정



프로그램 개발 과정

요구사항 분석

무엇을
만들 것인가를
결정한다.



설계

알고리즘을
설계한다.



구현

개발 도구를
사용하여
소스 코드를
작성한다.



테스팅

여러가지
경우에 대하여
실행하여 본다.



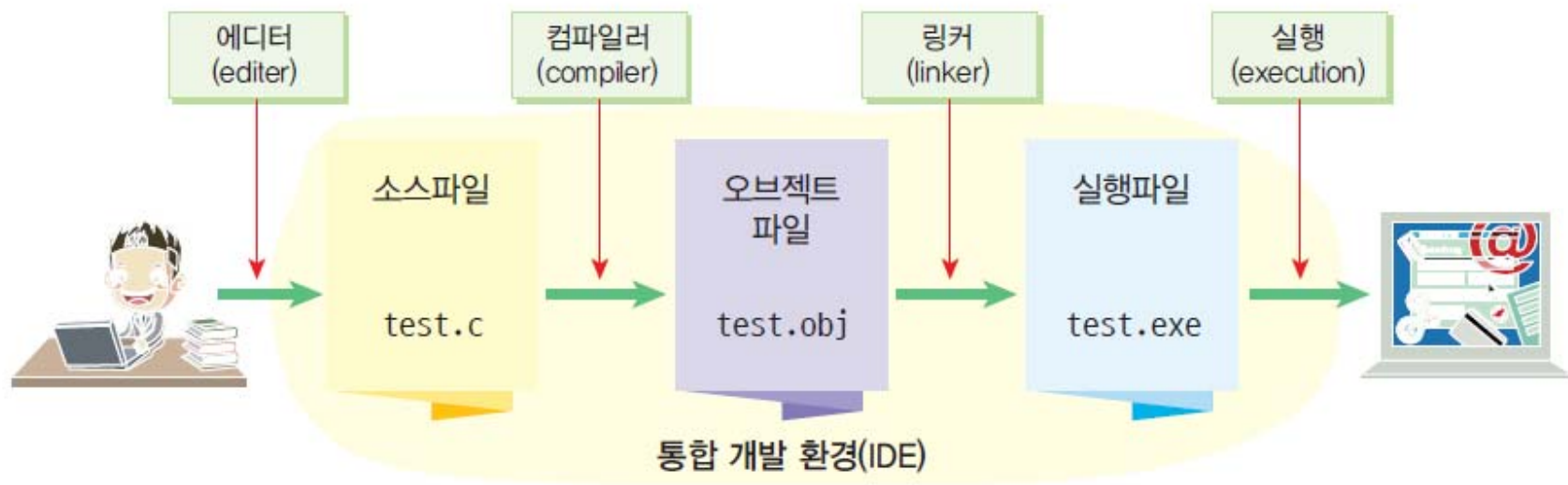
유지보수

사용자의 추가
요구사항을
반영한다.





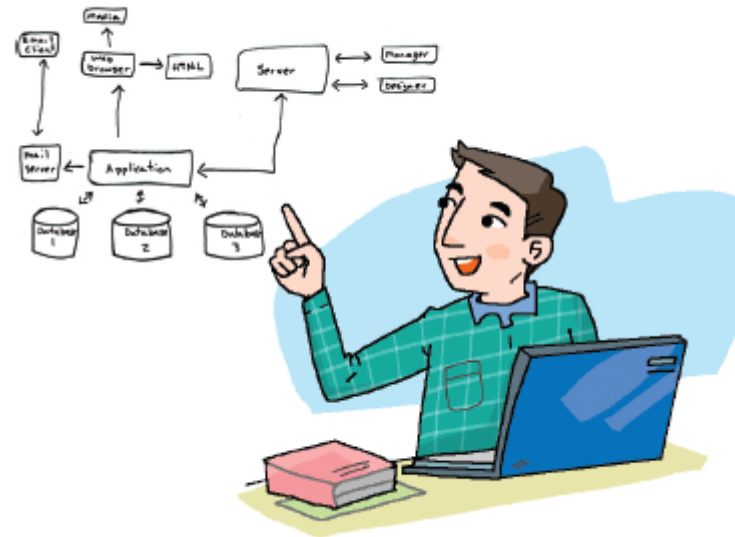
프로그램 개발 과정





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





소스 작성

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

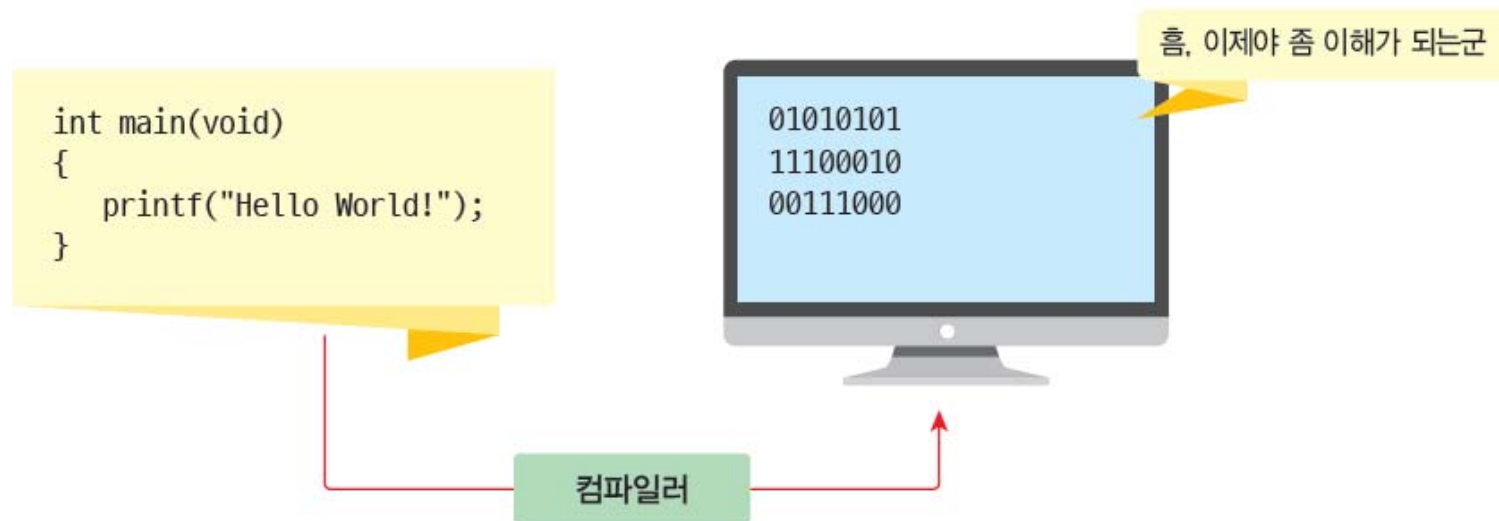


```
int main(void)
{
    printf("Hello World!");
}
```



컴파일

- 소스 프로그램을 오브젝트 파일로 변환하는 작업
- 오브젝트 파일 이름: (예) test.obj





컴파일 오류

- 컴파일 오류(compile error): 문법 오류
 - ▣ (예) He go to school;

The goes to
school.



오류가 있어요. 수정해주세요!

오류발생!!

```
int main(void)
{
    printf(Hello World!)
}
```



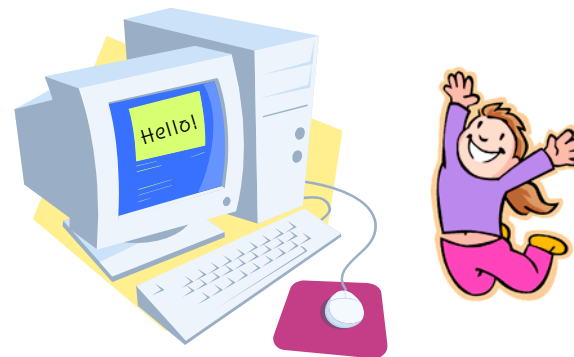
오류가 있어요. 수정해주세요!

오류발생!!



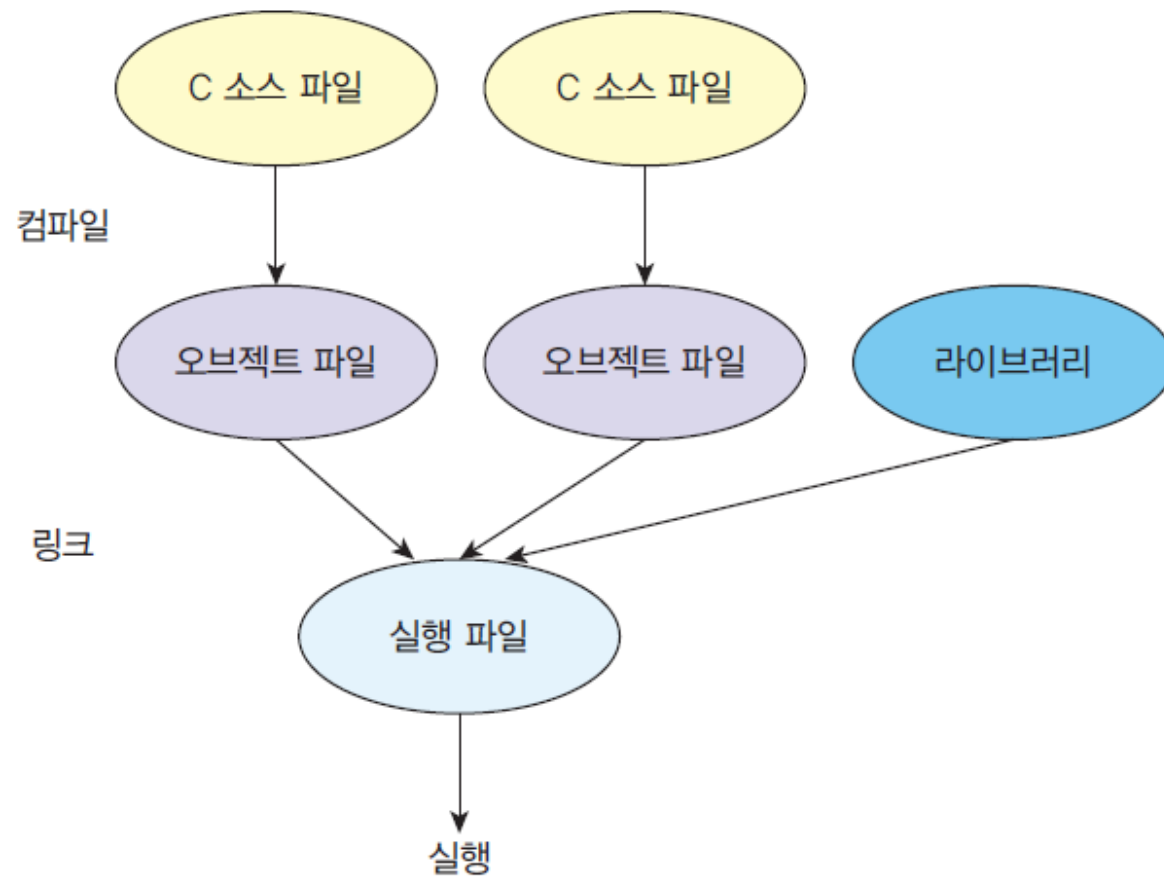
링크

- 컴파일된 목적 프로그램을 라이브러리와 연결하여 실행 프로그램을 작성하는 것
- 실행 파일 이름: (예) test.exe
- *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 시간 오류(run time error):

- ▣ (예) 0으로 나누는 것
- ▣ 잘못된 메모리 주소에 접근하는 것

□ 논리 오류(logical error):

- ▣ 문법은 틀리지 않았으나 논리적으로 정확하지 않는 것
- ▣ ① 그릇1과 그릇2를 준비한다.
② 그릇1에 밀가루, 우유, 계란을 넣고 잘 섞는다.
③ 그릇2를 오븐에 넣고 30분 동안 350도로 굽는다.





디버깅

- 소스에 존재하는 오류를 잡는 것





디버깅의 유래

- 1945년 마크 II 컴퓨터가 릴레이 장치에 낱아든 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 Autom started
1000 stopped - autom ✓
1500 HP - MC 2.13047695
PRO 2 2.13047695
conv 2.13047695
Relays 6-2 in 0.24 full speed speed test
1m relay 10.00 test.

1100 Started Cosine Tape (Sine check)
1525 Started Multi-Angle Test.

1545 Relay #70 Panel F (motion relay).

1700 Autom started.
1700 closed door.

First actual case of bug being found.



소프트웨어의 유지 보수

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





통합 개발 환경

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





통합 개발 환경의 예

- 비주얼 스튜디오: 마이크로소프트
- 이클립스(eclipse): 오픈 소스 프로젝트
- Dev-C++: 오픈 소스 프로젝트





비주얼 스튜디오 버전

- 커뮤니티(Visual Studio Community) 버전은 “기업 외 응용 프로그램 빌드 개발자를 위한 완벽한 기능의 확장 가능한 무료 도구”이다.
- 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

모든 개발, 앱 및 플랫폼에서 생산성 극대화

Visual Studio 2017의 새로운 기능

Visual Studio 다운로드

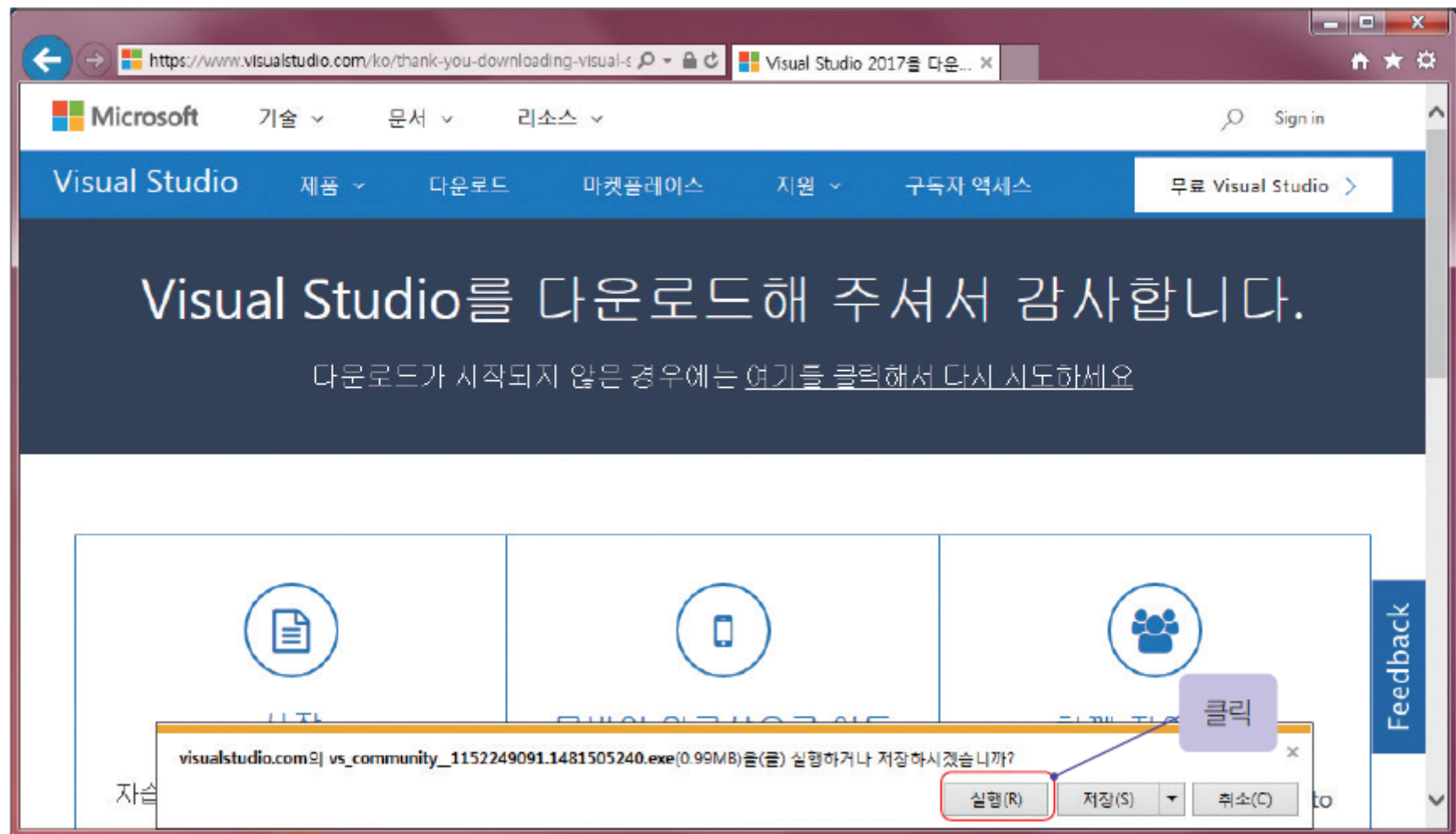
- Community 2017
- Professional 2017
- Enterprise 2017

Feedback

Quick Launch (Ctrl+Q)



비주얼 스튜디오 설치





비주얼 스튜디오 설치

Visual Studio
설치 중 — Visual Studio Community 2017 — 15.2 (26430.6)

워크로드 개별 구성 요소 언어 팩

Windows (3)

- ☐ 유니버설 Windows 플랫폼 개발
C#, VB, JavaScript 또는 선택적으로 C++를 사용하여 유니버설 Windows 플랫폼 응용 프로그램 개발을 만듭니다.
- ☒ C++를 사용한 데스크톱 개발
Visual C++ 도구 집합, ATL 및 MFC, C++/CLI 같은 선택적 기능을 사용하여 클래식 Windows 기반 응용 프로그램을 빌드합니다.
- ☐ .NET 데스크톱 개발
.NET Framework 및 WPF, Windows Forms 및 콘솔 응용 프로그램 개발을 만듭니다.

클라우드 및 웹 (7)

- ☐ ASP.NET 및 웹 개발
ASP.NET, ASP.NET Core, HTML, JavaScript 및 CSS를 사용하여 웹 응용 프로그램을 빌드합니다.
- ☐ Azure 개발
클라우드 앱을 개발하고 리소스를 만들기 위한 Azure SDK, 도구 및 프로젝트입니다.
- ☐ Python 개발
Python에 대한 편집, 디버깅, 대화형 개발 및 소스 제어입니다.
- ☐ Node.js 개발
비동기 이벤트 구동 JavaScript 런타임인 Node.js를 사용하여 확장 가능한 네트워크 응용 프로그램을 빌드합니다.
- ☐ 데이터 저장소 및 처리
데이터베이스 및 데이터 처리 응용 프로그램을 만듭니다.
- ☐ 데이터 과학 및 분석 응용 프로그램
데이터 과학 및 분석 응용 프로그램을 만듭니다.

원지
C:\Program Files (x86)\Microsoft Visual Studio\2017\Community

요약

▼ C++를 사용한 데스크톱 개발

포함됨

- ☒ Visual C++ 핵심 데스크톱 기능

옵션

 - ☒ VC++ 2017 v141 도구 집합(x86,x64)
 - ☒ C++ 프토폴링 도구
 - ☒ 데스크톱 C++ x86 및 x64용 Windows...
 - ☒ CMake용 Visual C++ 도구
 - ☒ Visual C++ ATL 지원
 - ☐ Windows 8.1 SDK 및 UCRT SDK
 - ☐ C++용 Windows XP 지원
 - ☐ MFC 및 ATL 지원(x86 및 x64)
 - ☐ C++/CLI 지원
 - ☐ Clang/C2(실험적)
 - ☐ 표준 라이브러리 모듈

1 계속하면 선택한 Visual Studio 버전에 대한 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 사용이 허가됩니다. 계속하면 이러한 라이선스에도 동의하게 됩니다.

설치 크기: 5.14GB

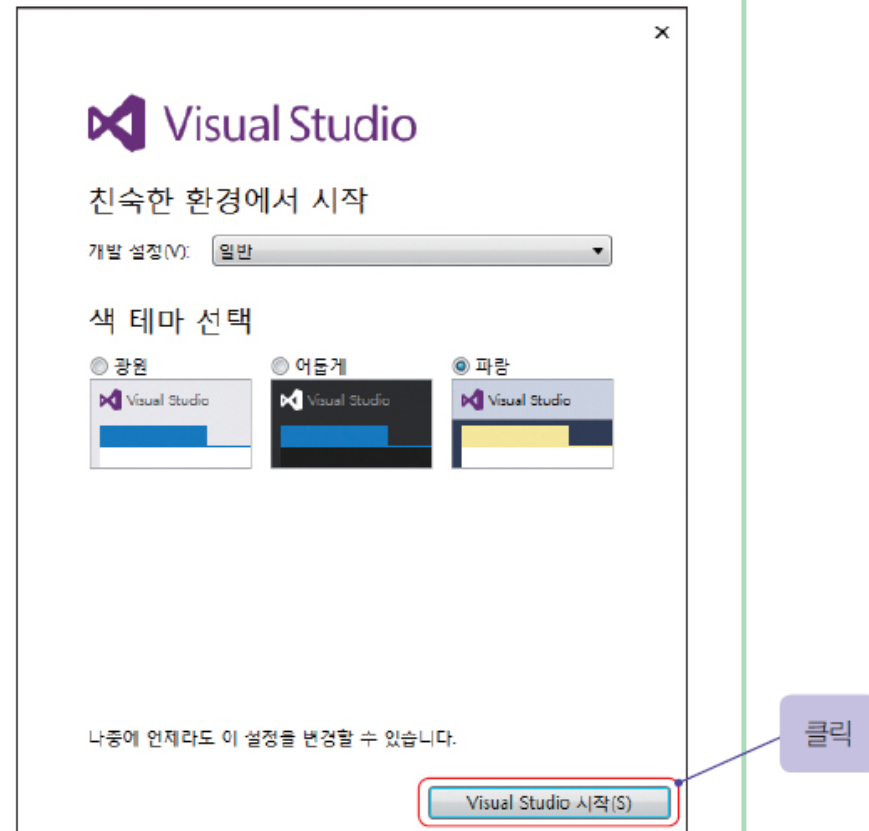
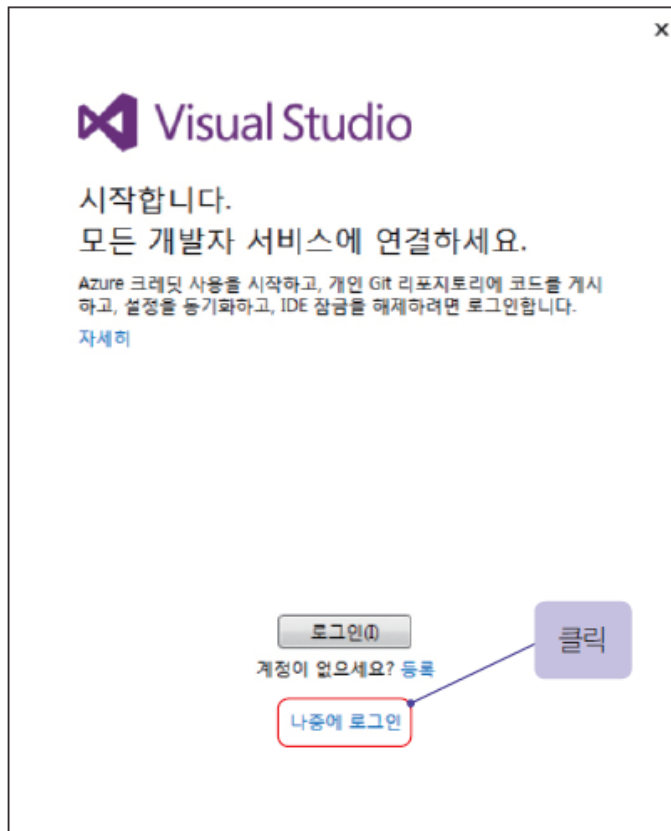
설치

클릭

클립보드에 복사되었습니다.



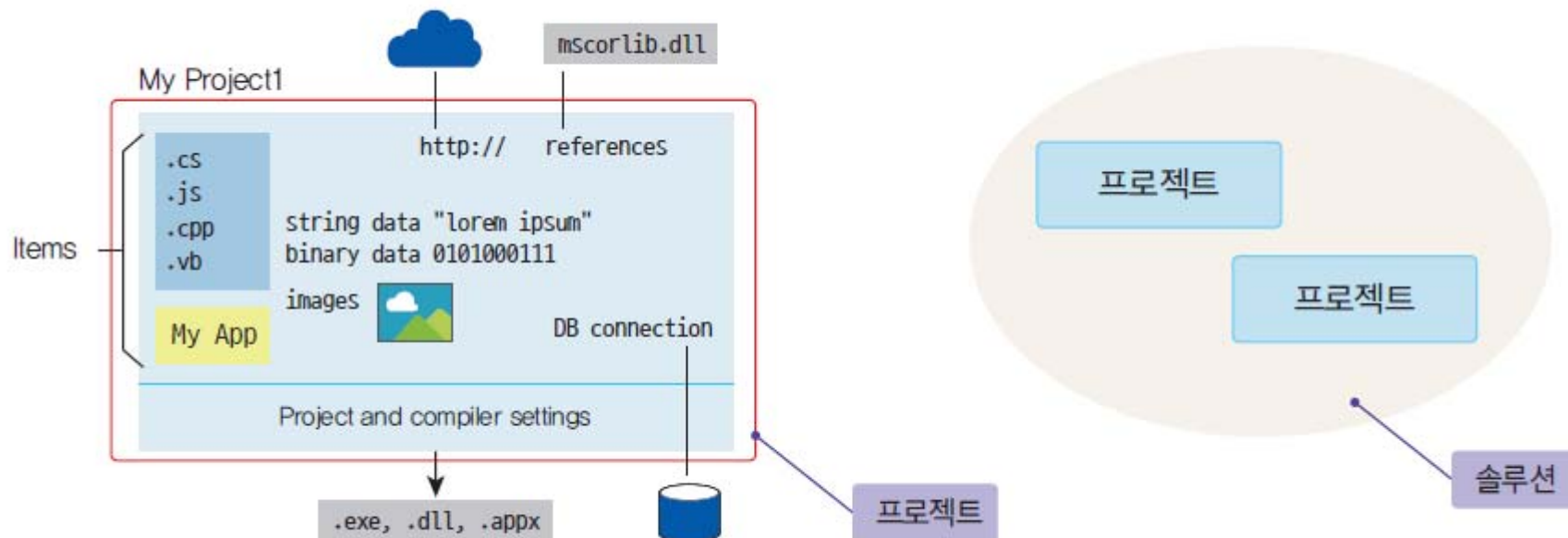
비주얼 스튜디오 시작





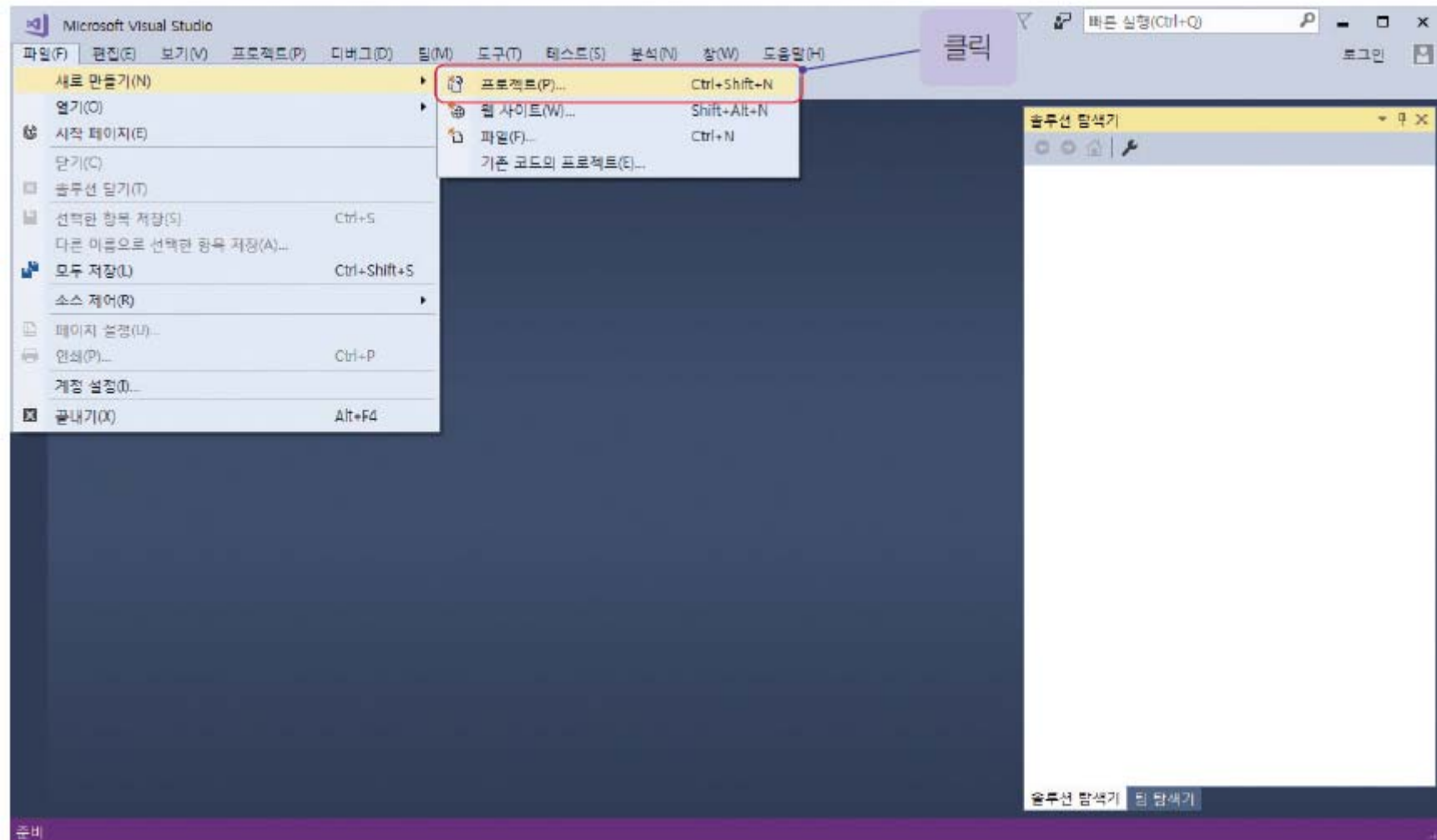
워크스페이스와 프로젝트

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



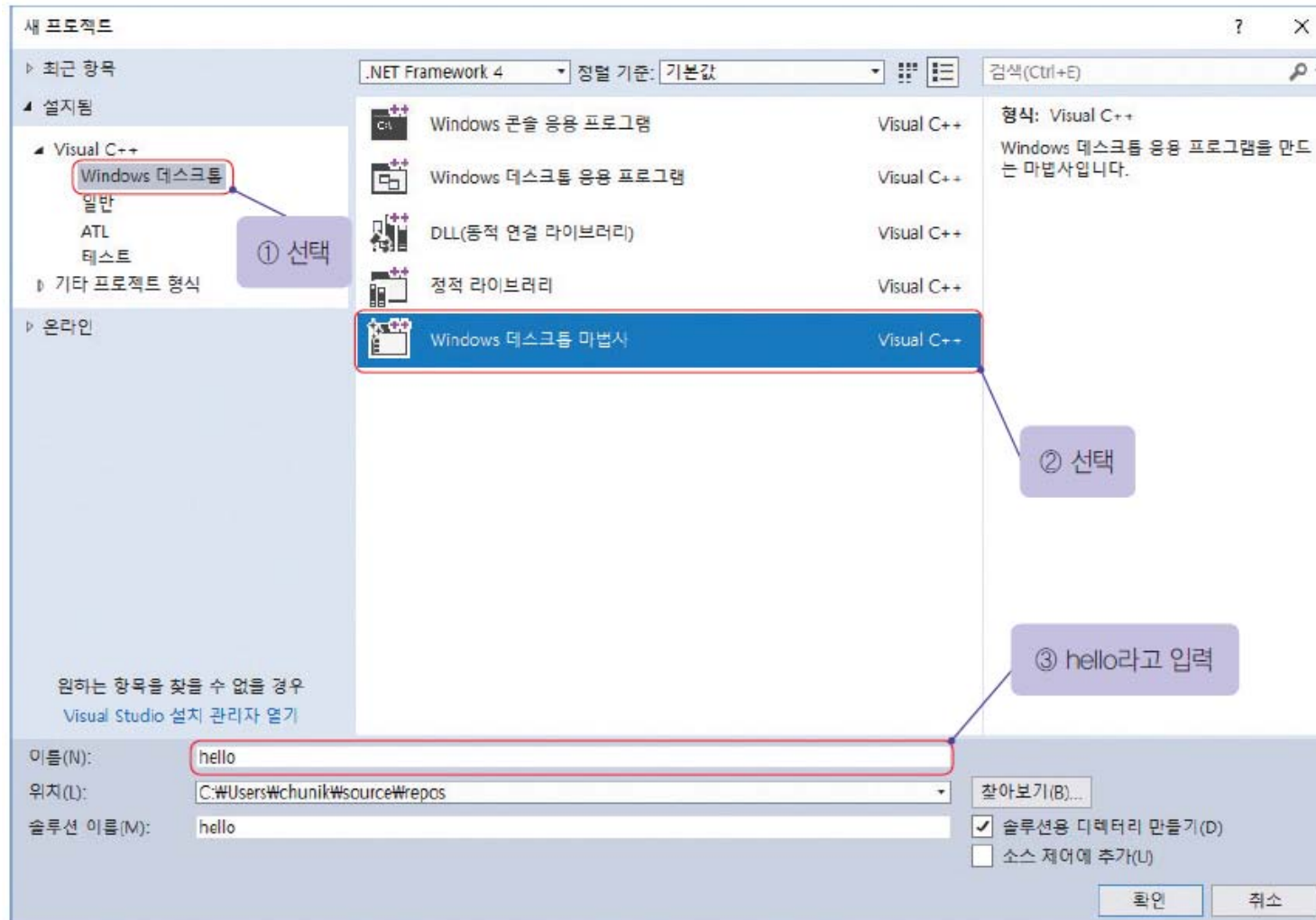


프로젝트 생성하기





프로젝트 생성하기





프로젝트 생성하기

Windows 데스크톱 프로젝트

응용 프로그램 종류(T):
콘솔 응용 프로그램(.exe)

다음에 대한 일반 헤더 추가:
☐ ATL(A)
☐ MFC(M)

추가 옵션:
☒ 빈 프로젝트(E)
☐ 내보내기 기호(X)
☐ 미리 컴파일된 헤더(P)
☐ SDL(Security Development Lifecycle) 검사(C)

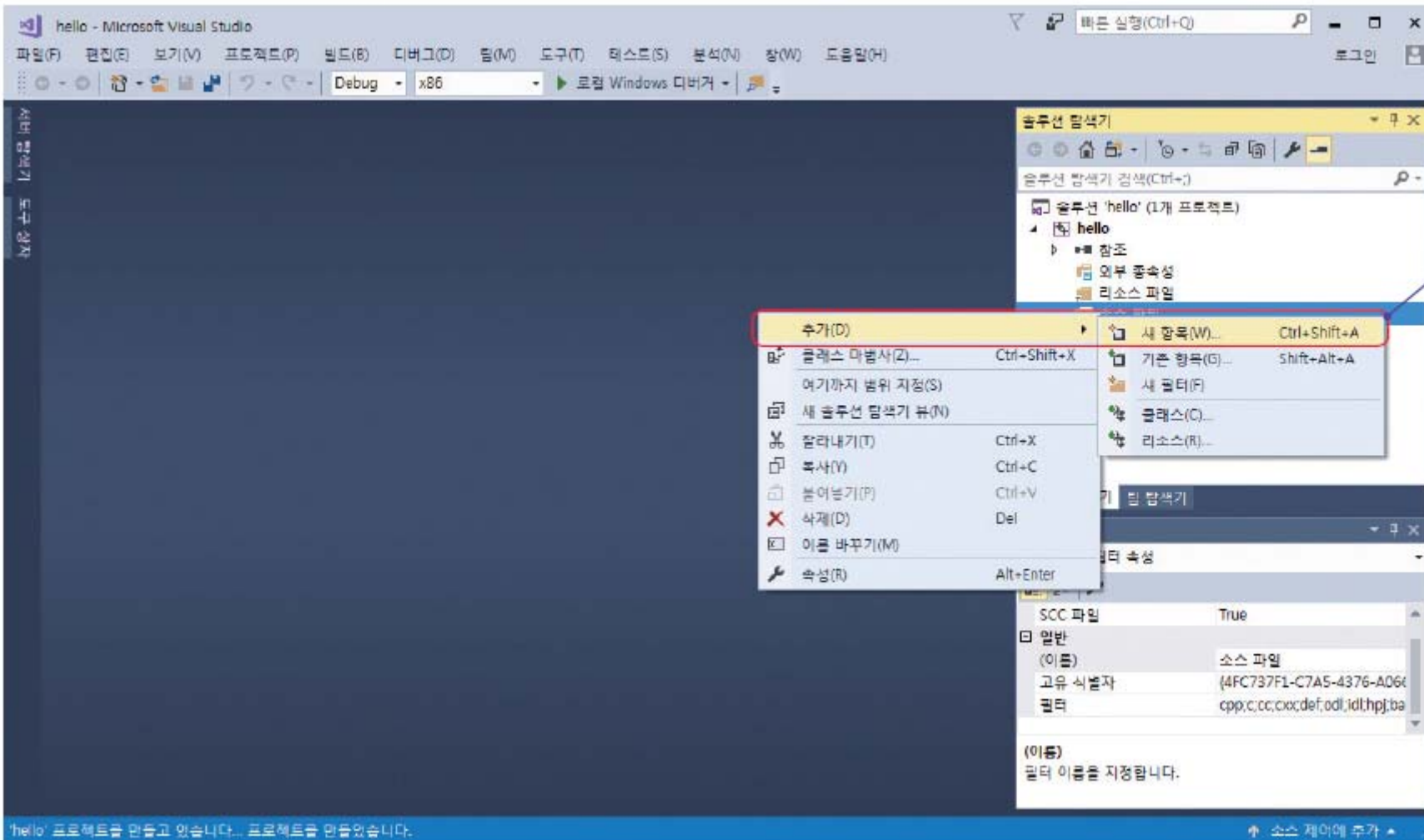
클릭

확인 취소

미리 컴파일된 헤더는 해제하고
빈 프로젝트는 체크한다.

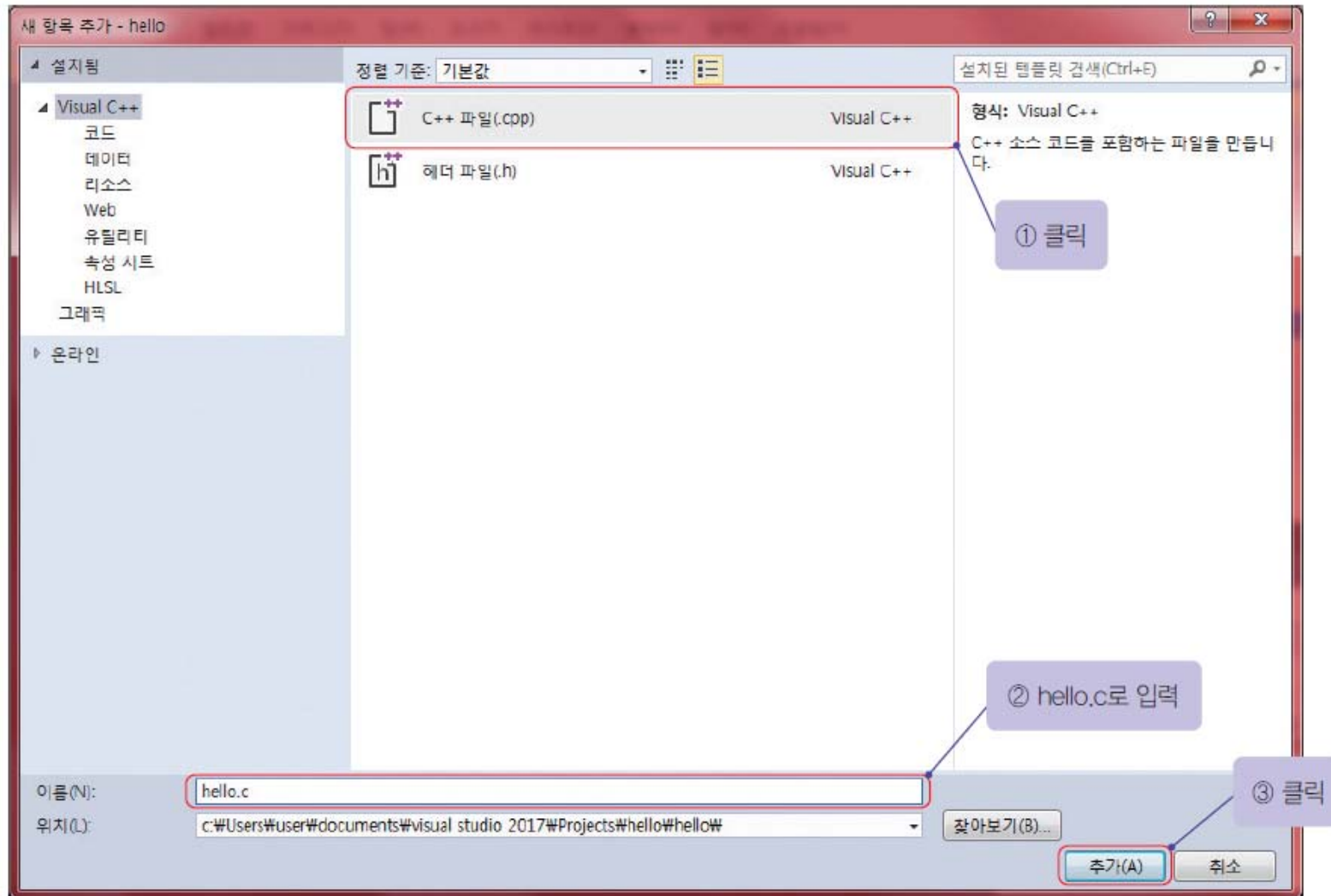


소스 파일 생성하기





소스 파일 생성하기





프로그램 입력

The screenshot displays the Microsoft Visual Studio IDE with a C program named 'hello.c' open. The code is as follows:

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

Annotations on the image include:

- A red box around the code is labeled "철자에 주의하여 입력한다." (Be careful of typos when entering).
- A purple box labeled "프로젝트" (Project) points to the 'hello' project in the Solution Explorer.
- A purple box labeled "솔루션" (Solution) points to the 'hello' solution in the Solution Explorer.

The Solution Explorer on the right shows the project structure:

- 솔루션 'hello' (1개 프로젝트)
 - hello
 - 참조
 - 외부 종속성
 - 리소스 파일
 - 소스 파일
 - hello.c
 - 헤더 파일

The Properties window at the bottom right shows the properties for the 'main' VCodeFunction:

속성	값
(Name)	main
File	c:\Users\wigchun\Documents
FullName	main
IsDefault	False

The status bar at the bottom shows the current line is 6, column is 21, and the character count is 11.



기호는 정확하게 입력

문장의 끝에는 세미콜론



컴파일하기

The screenshot shows the Visual Studio IDE with the 'hello.c' file open. The menu 'Build (B)' is open, and the 'Build' option is highlighted. A callout box points to the 'Build' option with the text: ① 컴파일, 링크하여서 실행 파일을 생성한다.

The 'Output' window at the bottom shows the compilation results:

```
1>----- 빌드 시작: 프로젝트: hello, 구성: Debug Win32 -----
1>hello.c
1>hello.vcxproj -> c:\Users\Wigchun\documents\visual studio 2017\Projects\hello\Debug\hello.exe
1>hello.vcxproj -> c:\Users\Wigchun\documents\visual studio 2017\Projects\hello\Debug\hello.pdb (Partial PDB)
===== 빌드: 성공 1, 실패 0, 최신 0, 생략 0 =====
```

A second callout box points to the 'Output' window with the text: ② 오류가 없다면 이런 메시지가 나온다.

The 'Solution Explorer' on the right shows the project structure:

- 리소스 파일
- 소스 파일
 - hello.c
- 헤더 파일

The 'Properties' window on the right shows the properties for the 'main' file:

(Name)	main
File	c:\Users\Wigchun\documents\visual studio 2017\Projects\hello\Debug\hello.exe
FullName	main
IsDefault	False



프로그램 실행 하기

hello - Microsoft Visual Studio

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

Debug

창(W)
그래픽(C)
디버깅 시작(S) F5
디버그하지 않고 시작(H) Ctrl+F5
성능 프로파일러(F) Alt+F2
프로세스에 연결(P) Ctrl+Alt+P

프로그램의 출력

선택

프로그램의 실행이 끝나면
항상 나오는 메시지

```
#include <stdio.h>

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

C:\Windows\system32\cmd.exe

Hello World! 계속하려면 아무 키나 누르십시오 . . .

출력

출력 보기 선택(S): 빌드

모든 목록 출력

준비 줄: 1 열: 1 문자: 1 INS 소스 제어에 추가



장간 점검



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



첫 번째 프로그램의 설명

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

```
int main(void)
```

```
{
```

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

```
    return 0;
```

```
}
```





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;
```

```
}
```

여기다 원하는 작업을
수행하는 문장을 적어준다.

프로그램



간략한 소스 설명

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

헤더파일을 포함한다.

```
int main(void)
```

메인 함수 시작

```
{
```

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

화면에 "Hello World!"를 출력

```
    return 0;
```

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.

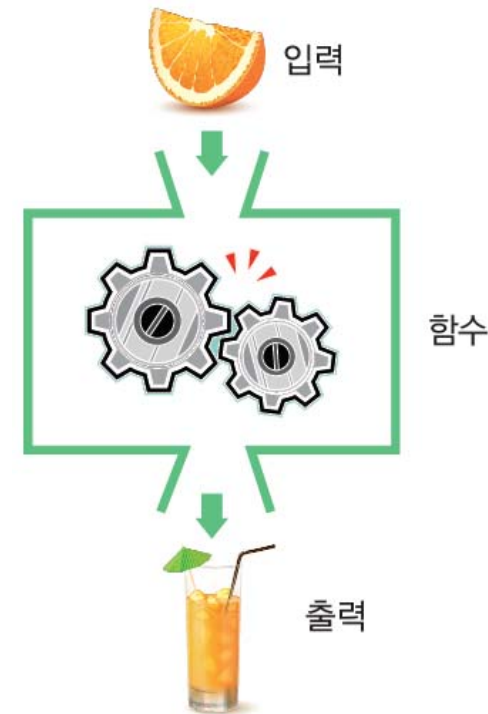
`#include <stdio.h>`

- 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 $y = x^2 + 1$
- (참고) 수학적 함수
- 프로그램 = 함수의 집합





main() 함수

- 함수의 반환값

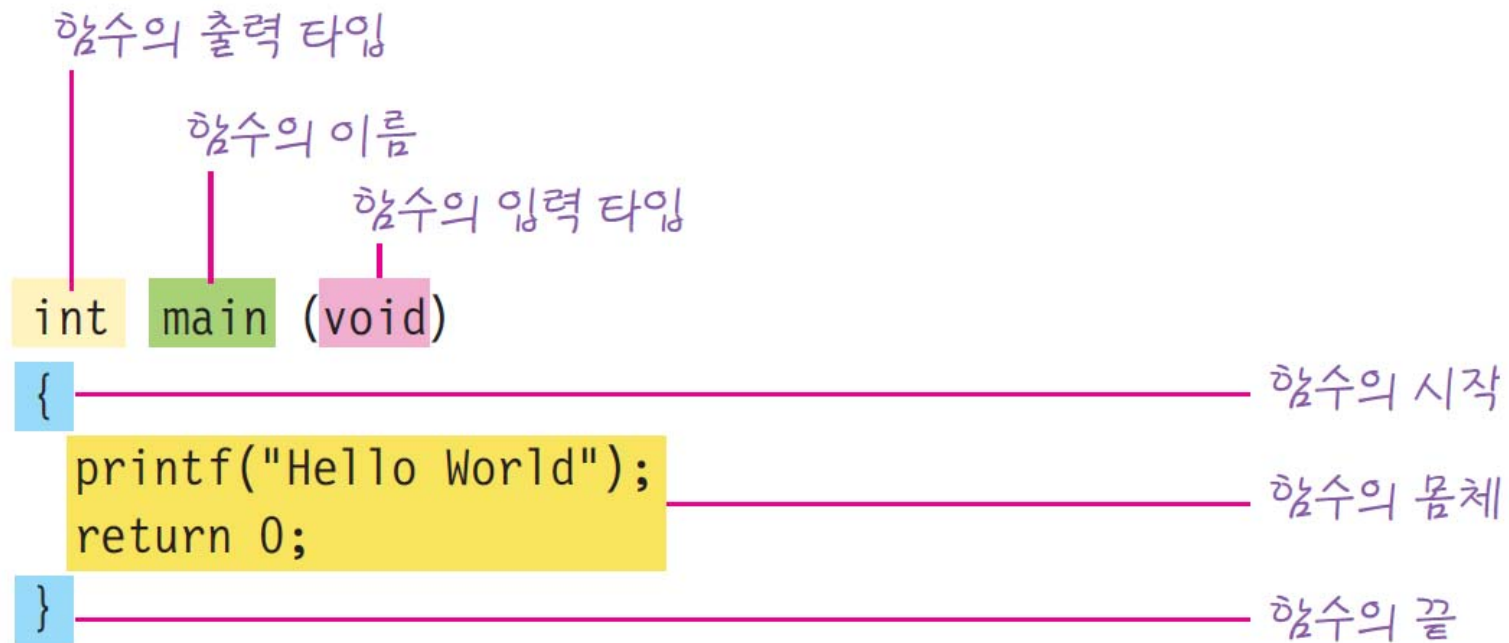
- 함수의 입력은 없음!

int main(void)

- 함수의 이름
- main()은 가장 먼저 수행되는 함수



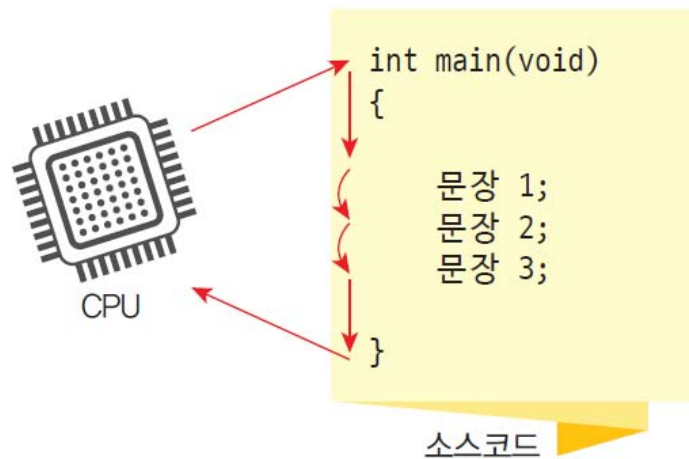
함수의 간단한 설명





문장

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





printf() 호출

- printf()는 컴파일러가 제공하는 함수로서 출력을 담당한다

printf("Hello World!");

- 큰따옴표 안의 문자열이 화면에 출력된다.





함수의 반환값

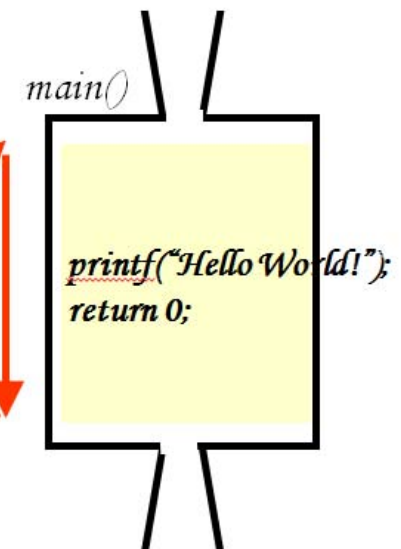
- return은 함수의 결과값을 외부로 반환

return 0;

- 반환값은 0

운영 체제

운영 체제





장간 점검

- 문장의 끝에 추가하여야 하는 기호는?
- C프로그램에 반드시 있어야 하는 함수는?
- `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

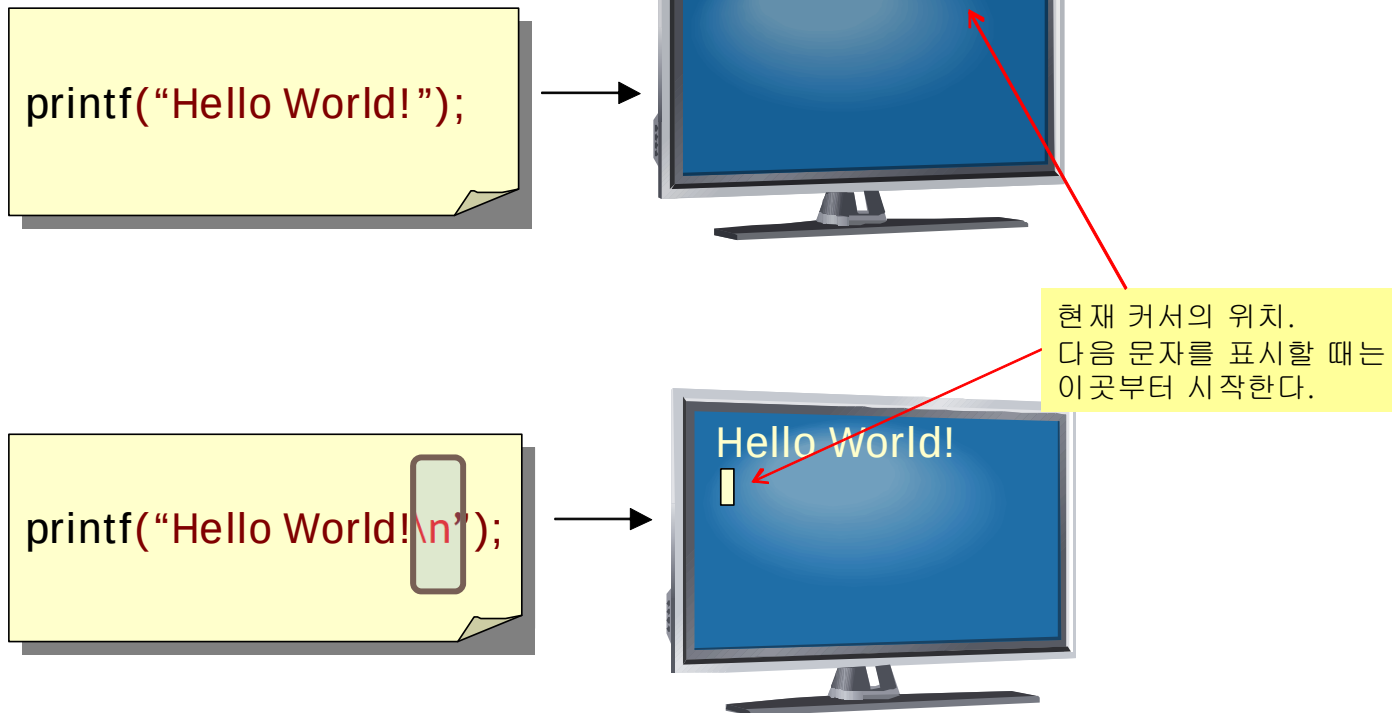


Hello World! Kim ChulSoo



줄바꿈 문자

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





잘바꿈 문자 2개를 사용하면?

```
printf("Hello \nWorld! \n");
```





변경된 프로그램

- 줄바꿈 문자를 포함하면 우리가 원하던 결과가 된다



```
#include <stdio.h>

int main(void)
{

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

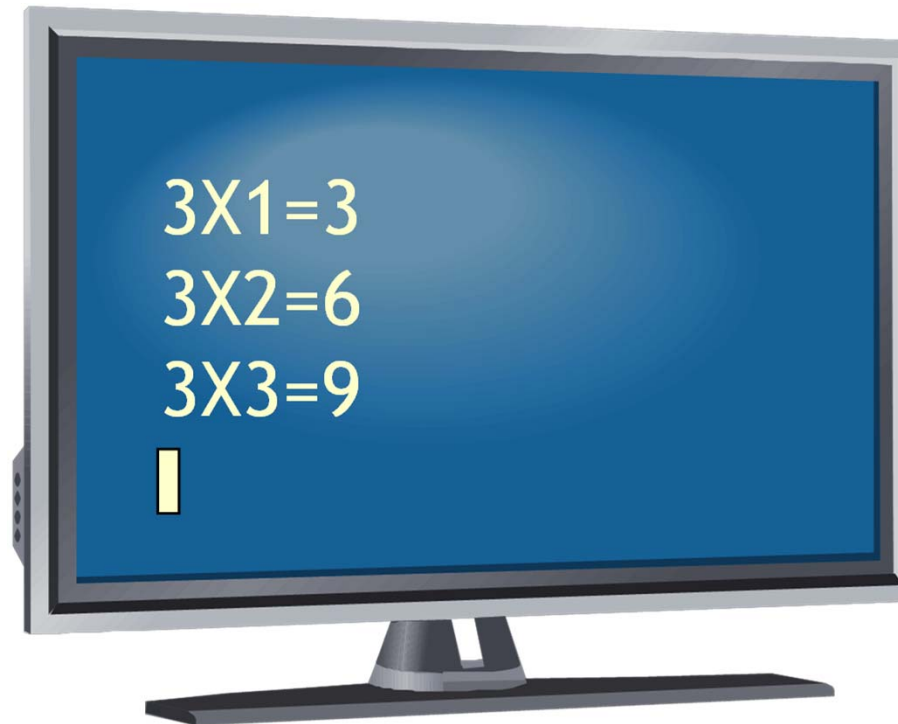
    return 0;
}
```





응 프로그램 #2

- 다음과 같은 출력을 가지는 프로그램을 제작하여 보자.





- 역시 문장들은 순차적으로 수행된다는 점을 이용한다.

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

```
int main(void)
```

```
{
```

```
    printf("3 X 1 = 3\n");
```

```
    printf("3 X 2 = 6\n");
```

```
    printf("3 X 3 = 9\n");
```

```
    return 0;
```

```
}
```

3개의 문장은
순차적으로 실행된다.



장간 점검

- 화면에 새로운 줄을 만드는데 사용되는 특수한 기호는?
- “사과”, “오렌지”, “포도”를 한 줄에 하나씩 출력하는 프로그램을 작성하여 보자.
- 구구단 3단 전체를 출력하는 프로그램을 작성하여 보자.





lab: 간단한 계산을 해보자

- 덧셈과 뺄셈, 곱셈, 나눗셈 계산을 하는 프로그램을 작성해보자.

📌 실행결과

$2+5=5$

$2-3=-1$

$2*3=6$

$2/3=0$



solution

```
#include <stdio.h>

int main(void)
{
    printf("2+5=%d\n", 2 + 3);
    printf("2-3=%d\n", 2 - 3);
    printf("2*3=%d\n", 2 * 3);
    printf("2/3=%d\n", 2 / 3);
    return 0;
}
```



오류 수정 및 디버깅

- 컴파일이나 실행 시에 오류가 발생할 수 있다.
- 에러와 경고
 - ▣ 에러(error): 심각한 오류
 - ▣ 경고(warning): 경미한 오류





오류의 종류

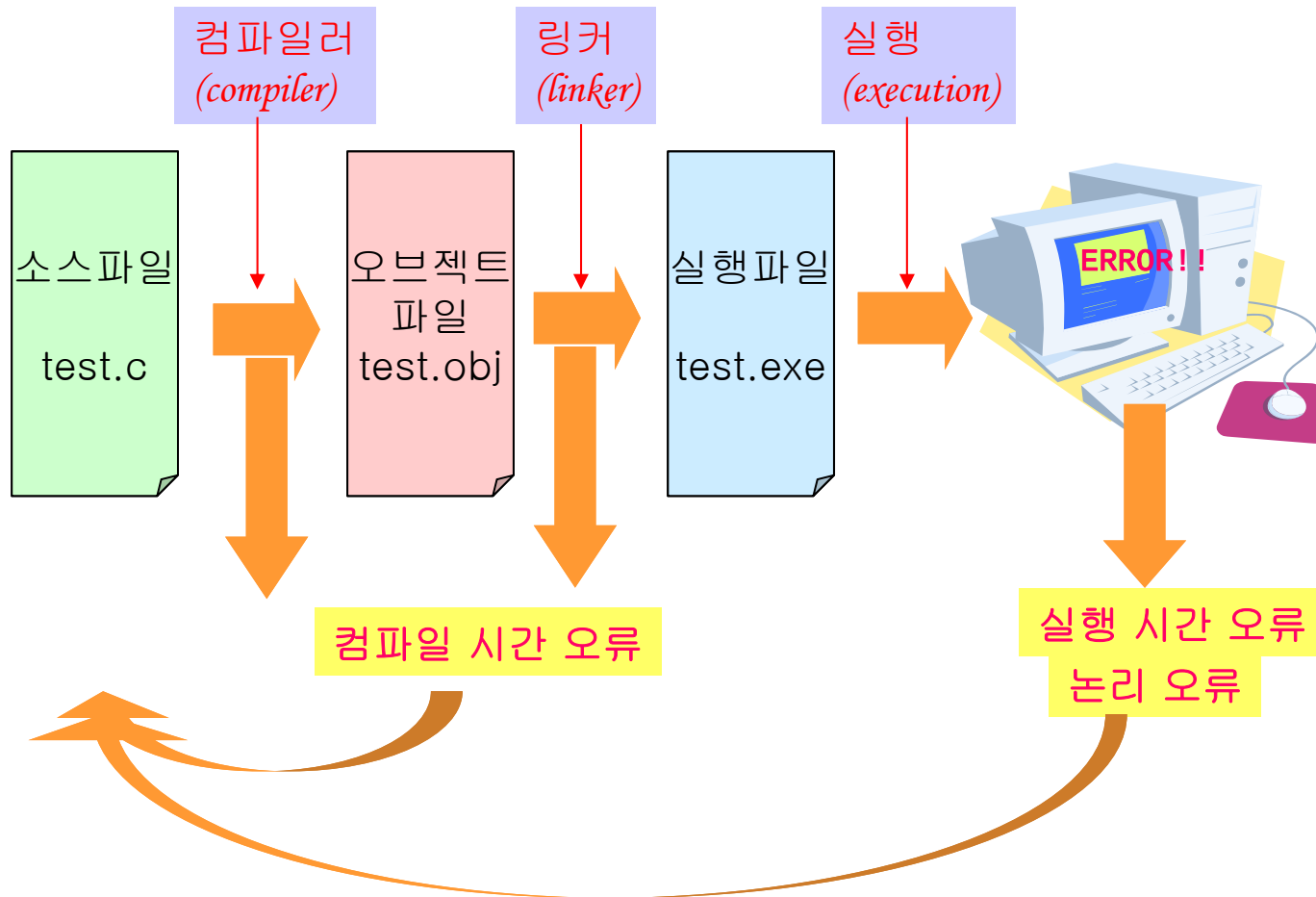
□ 오류의 종류

- ▣ 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) 디버그(D) 팀(M) 도구(T) 테스트(S) 분석(N) 창(W) 도움말(H) 로그인

Debug x86 로컬 Windows 디버거

error1.c (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	구문 오류: ';'이(가) 'return' 앞에 없습니다.	add	error1.c	6	

오류가 발견된 소스 파일

오류가 발견된 줄번호

오류 목록 출력

준비 줄: 7 열: 2 문자: 2 INS

게시



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

add (전역 범위) main(void)

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

문자열을 표시할 때, "와"를 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

속성

main VCodeFunction

C++

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

C++

준비 줄: 6 열: 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.

오류 목록

전체 솔루션 2 오류 1 경고 0 메시지 빌드 + IntelliSense

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

함수를 찾지 못했음.

솔루션 탐색기

솔루션 탐색기 검색(Ctrl+;)

솔루션 'add' (1개 프로젝트)

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

속성

main VCodeFunction

속성	값
(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("Hey!");
```

```
    printf("Good Morning");
```

```
    return 0;
```

```
}
```

줄이 바뀌지
않았음!

Hey!Good Morning



논리 오류가 수정된 프로그램

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

```
int main(void)
```

```
{
```

```
    printf("Hey! \n");
```

```
    printf("Good Morning \n");
```

```
    return 0;
```

```
}
```

논리 오류
수정!!



Hey!
Good Morning
|



디버깅

□ 디버깅: 논리 오류를 찾는 과정

아무래도 이부분이 수상해..



프로그램의 실행결과

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





디버거(debugger)

hello (디버깅) - Microsoft Visual Studio

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

프로세스: [144720] hello.exe

hello.c

```
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

디버그 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 디버그 대상(H)
- 프로파일러
- ▶ 한 단계씩 코드 실행(I) F11
- 프로세서 단위 실행(O) F10
- 프로세서 다가기(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...

한 문장 단위로 실행한다.

진단 도구

진단 및 세션: 0 초(178 ms)(가) 선택됨

CPU (모든 프로세서에 대한 비율(%))

요약 이벤트 메모리 사용량 CPU 사용량

이벤트

표시(1/1)

스냅숏 만들기

스택

hello.exe!main() 줄 4

부 코드

프레임은 슬바르지 않거나 누락되었거나 kernel32.dll에 대해 로드된 기호가 없음

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;
    }
}
```

hello.c x
(전역 범위)
main(void)
100 %
{
 /* 첫 번째 프로그램 */
 #include <stdio.h>
 int main(void)
 {
 printf("Hello World!\n");
 printf("Good Morning\n");
 return 0;
 }
}

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

hello.c x
(전역 범위)
main(void)
100 %
{
 /* 첫 번째 프로그램 */
 #include <stdio.h>
 int main(void)
 {
 printf("Hello World!\n");
 printf("Good Morning\n");
 return 0;
 }
}

C:\Users\chun#documents#visual studio 2010#Projects#hello#Debug#hello.exe
Hello World!
Good Morning



작업 점검

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





mini project

□ 오류를 수정해보자!

```
#include <stdio.h>

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



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.

print가 아니고 printf이어야 한다.

문자열에는 따옴표를 붙인다.



Mini Project

□ 오류를 수정해보자!

```
#include <stdio.h>

int main(void)
{
    printf("안녕하세요 ? \n");
    printf("이번 코드에는 많은 오류가 있다네요\n");
    printf("제가 다 고쳐보겠습니다.\n");
    return 0;
}
```



Q & A

