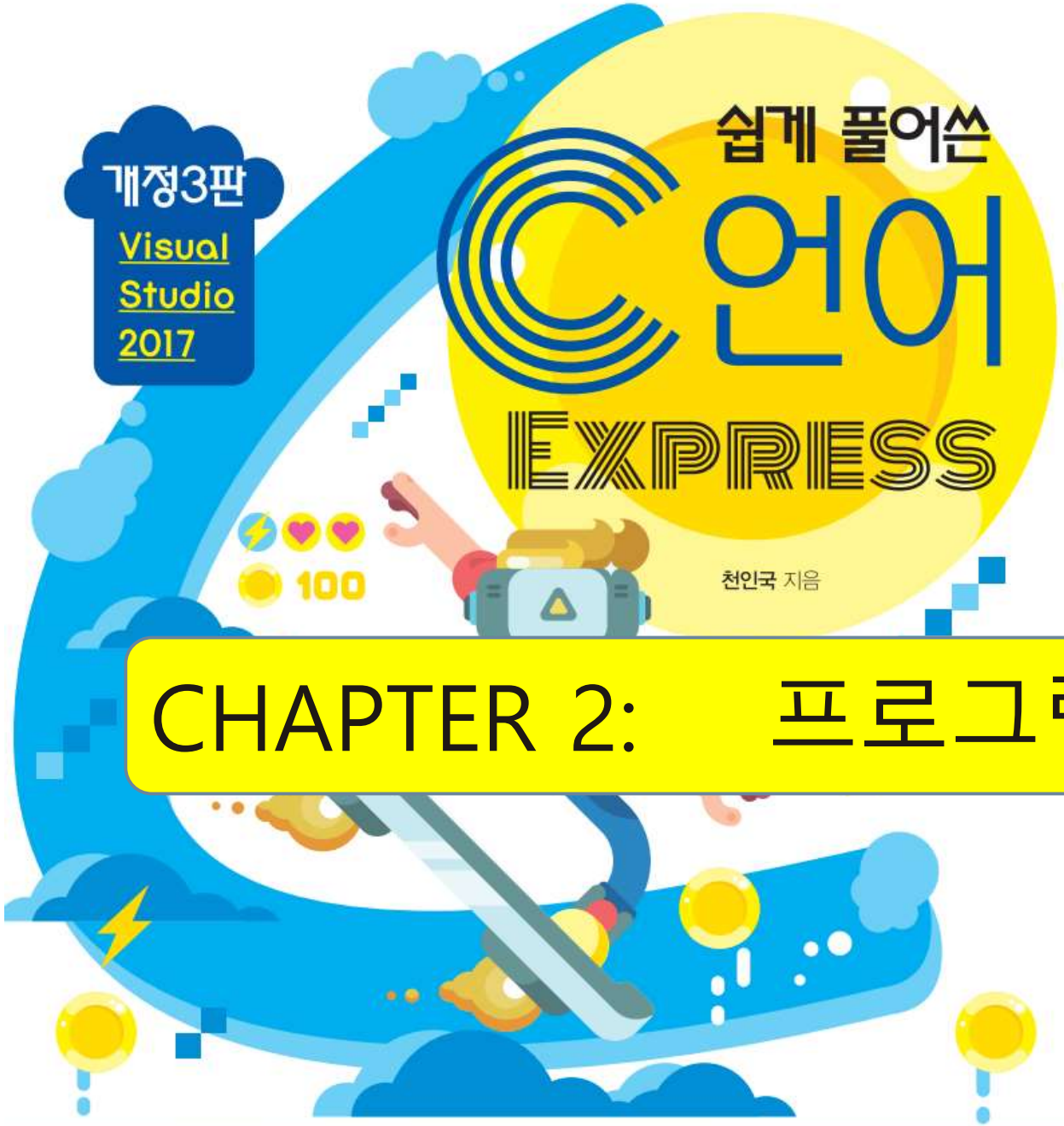


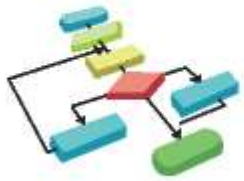
CHAPTER 2: 프로그램 작성 과정

프로그램 개발 과정

요구사항 분석 → **설계** → **구현** → **테스팅** → **유지보수**

무엇을 만들 것인가를 결정한다.

알고리즘을 설계한다.

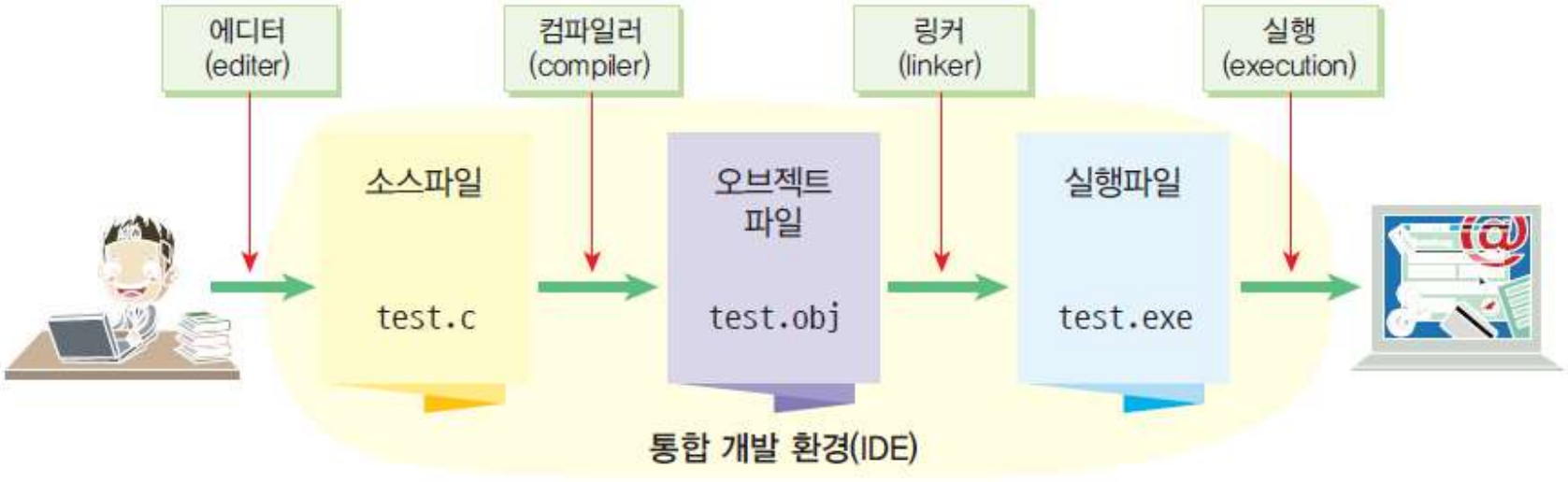


개발 도구를 사용하여 소스 코드를 작성한다.

여러가지 경우에 대하여 실행하여 본다.

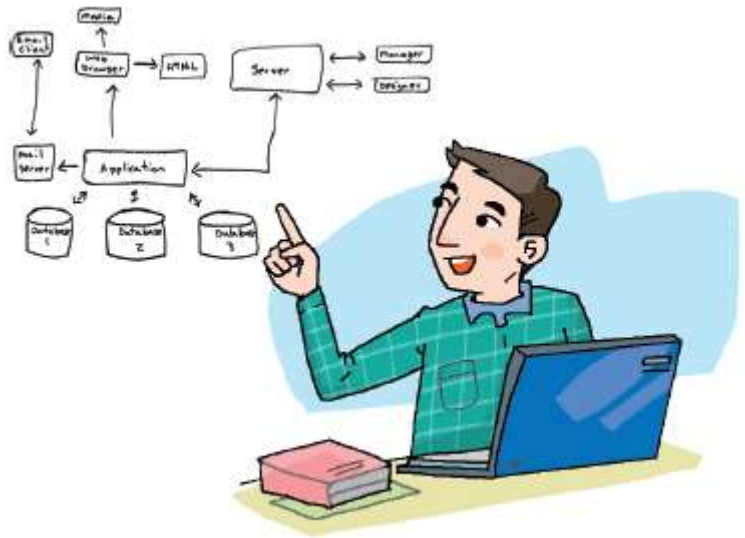
사용자의 추가 요구사항을 반영한다.

프로그램 개발 과정


에디터
(editer)
컴파일러
(compiler)
링커
(linker)
실행
(execution)
소스파일
test.c
오브젝트
파일
test.obj
실행파일
test.exe
통합 개발 환경(IDE)

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



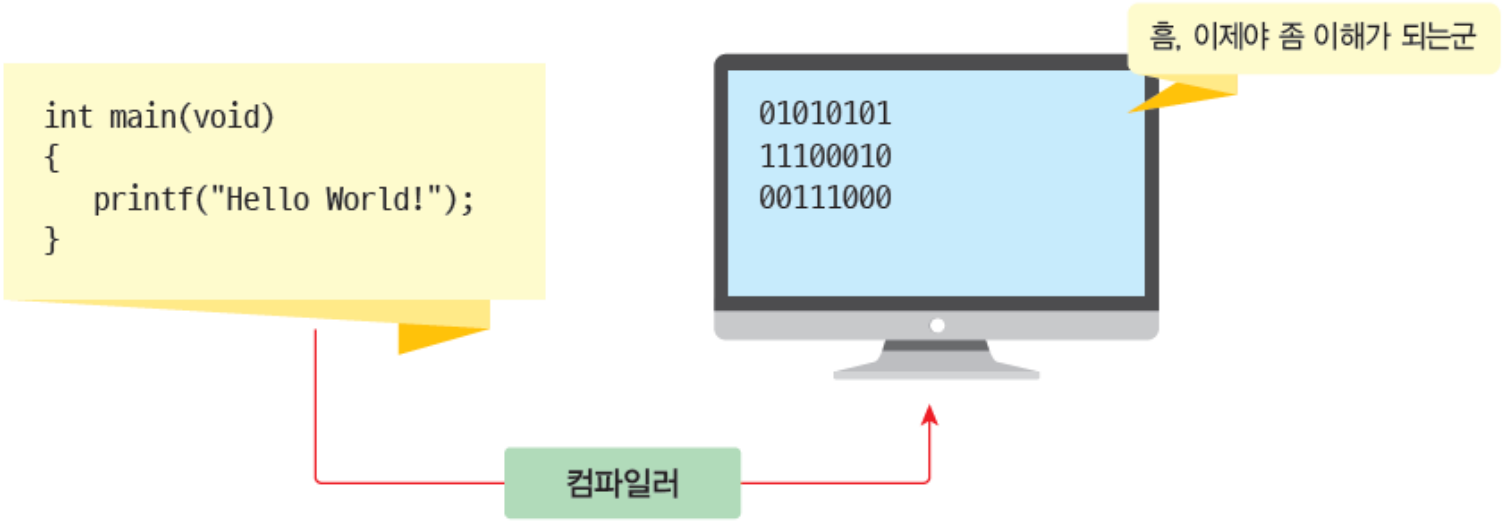
소스 작성

- 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!");
}
```

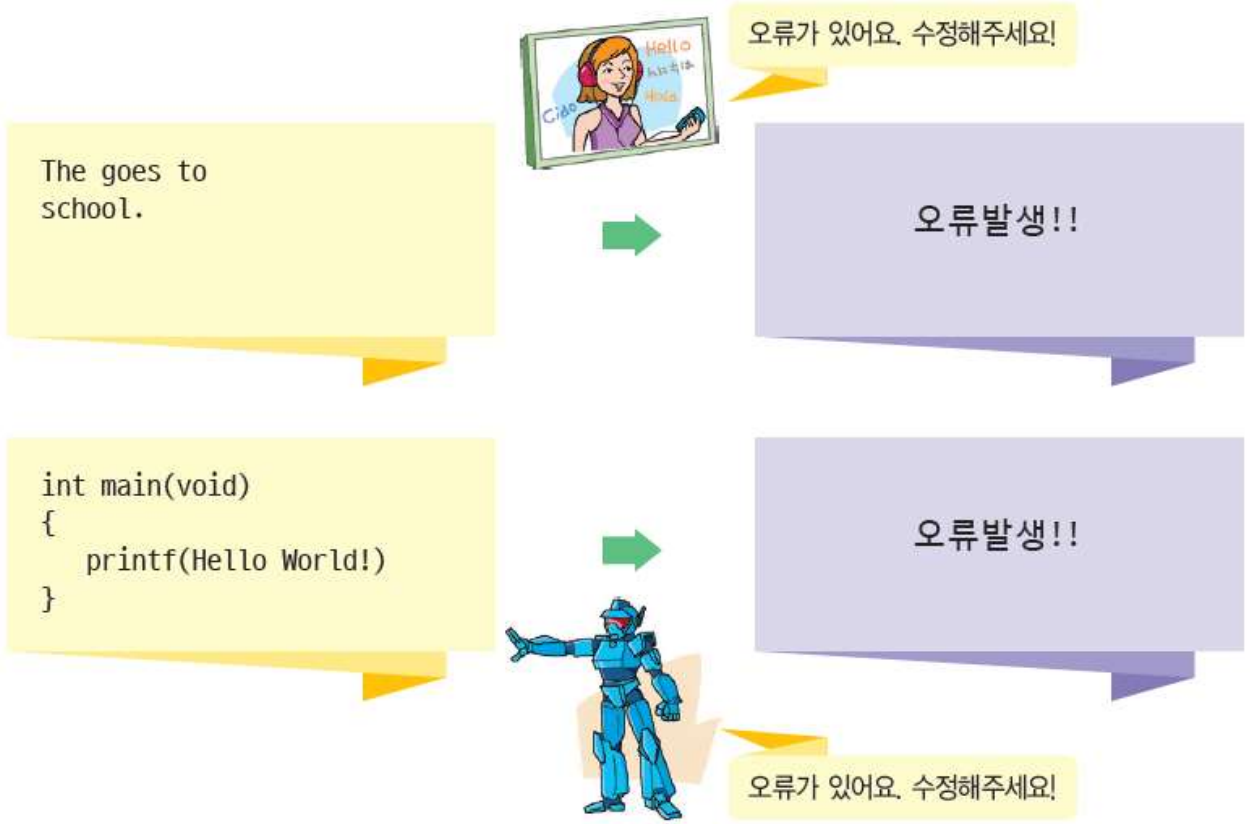
컴파일

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



컴파일 오류

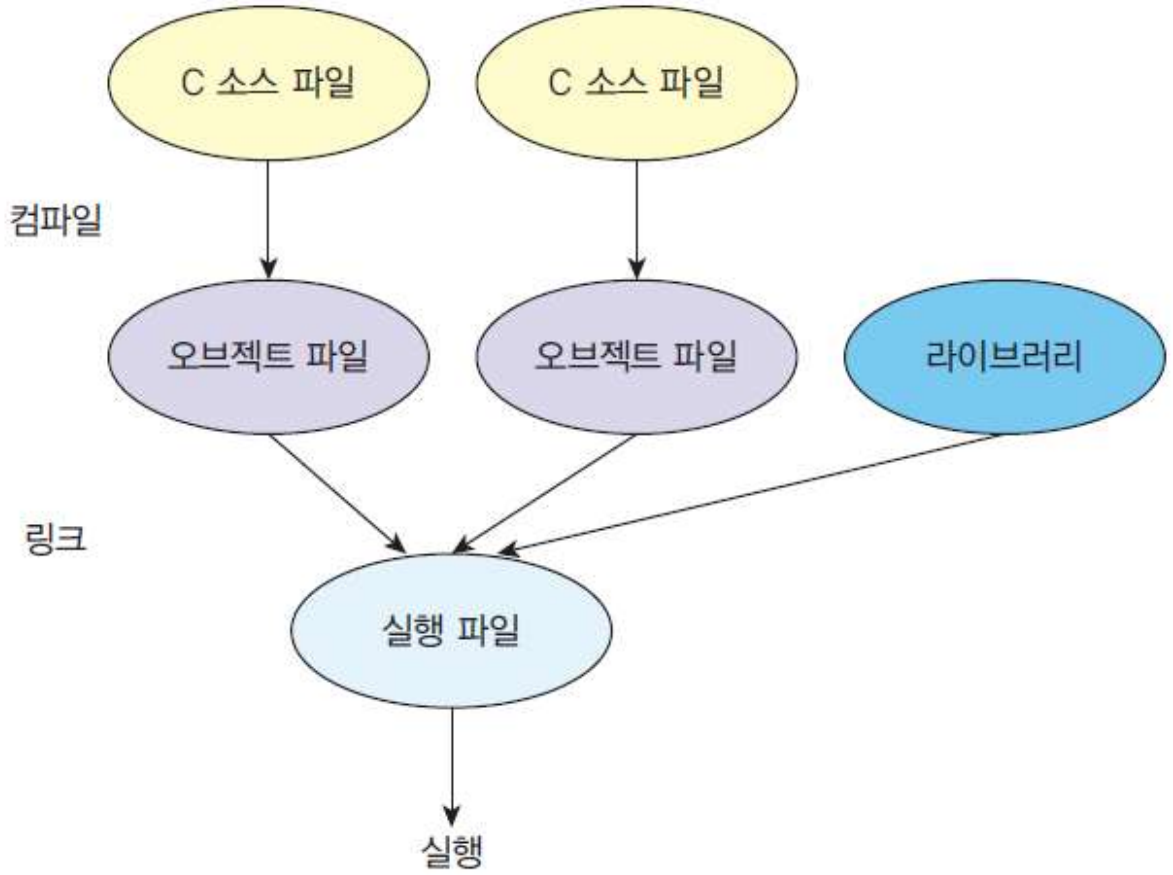
- 컴파일 오류(complie error): 문법 오류
  - (예) He go to school;



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
  - (예)

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

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% 이상을 차지

통합 개발 환경

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

통합 개발 환경의 예

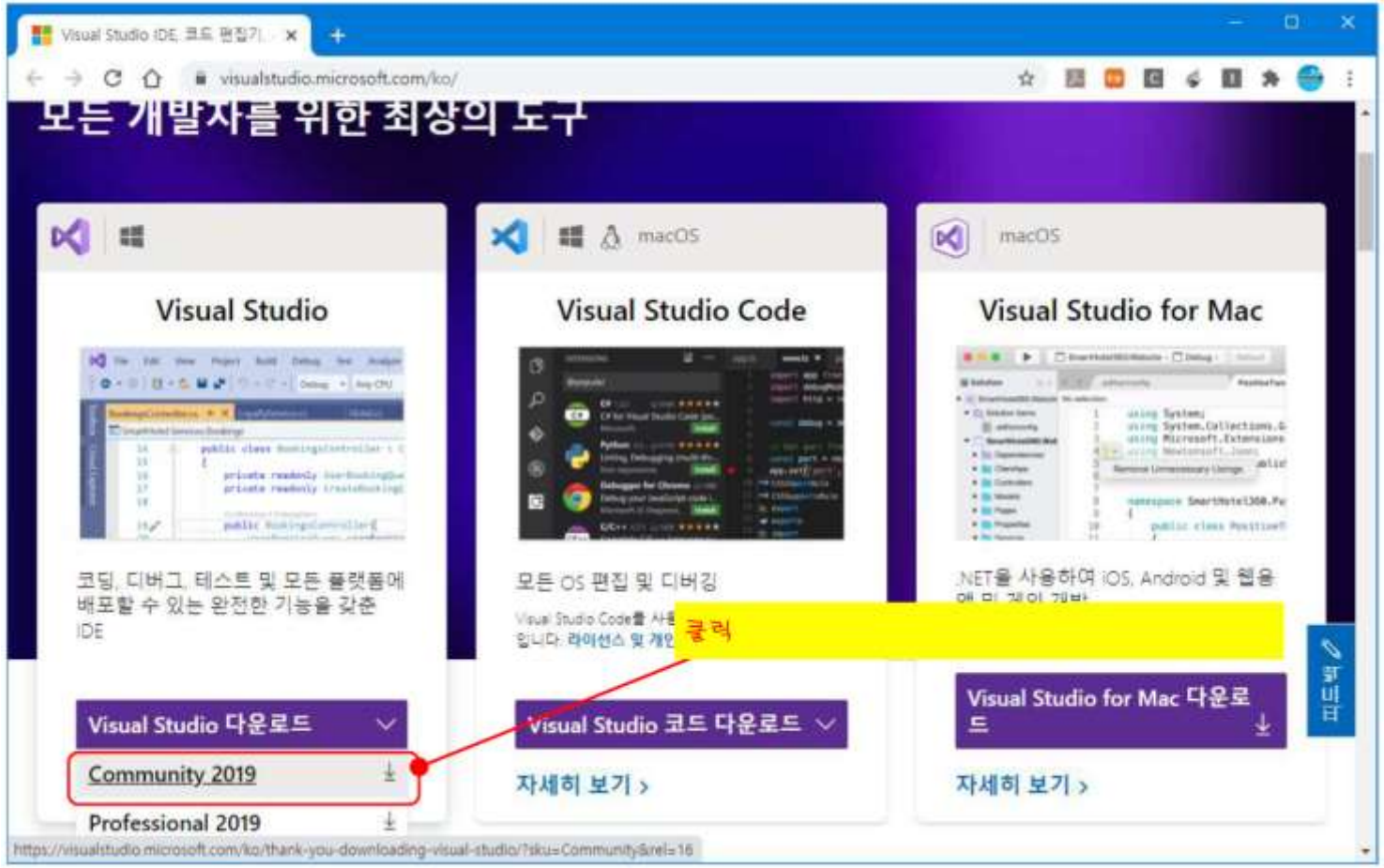
- 비주얼 스튜디오: 마이크로소프트
- 이클립스(eclipse): 오픈 소스 프로젝트
- Dev-C++: 오픈 소스 프로젝트

비주얼 스튜디오 버전

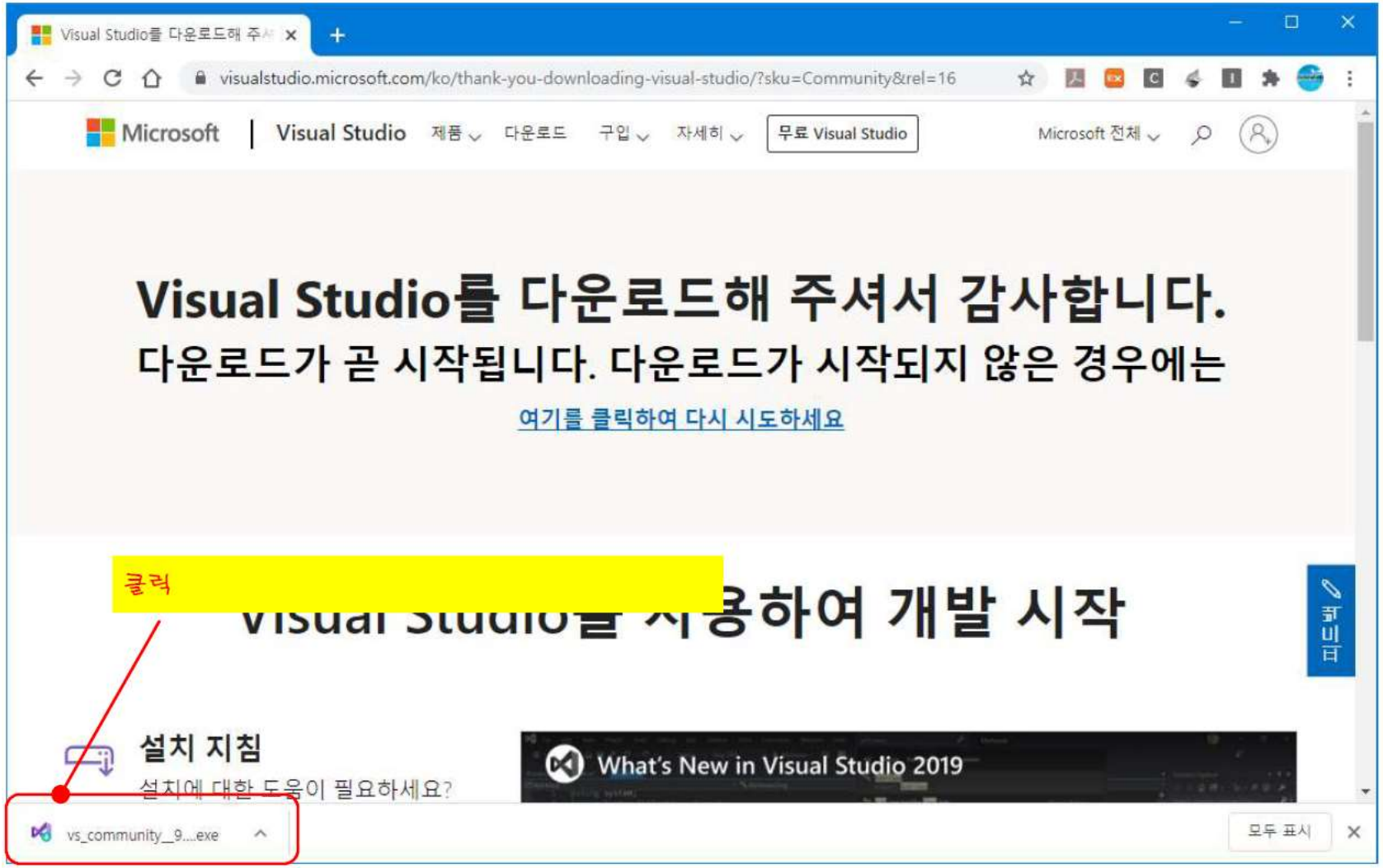
- 커뮤니티(Visual Studio Community) 버전은 “기업 외 응용 프로그램 빌드 개발자를 위한 완벽한 기능의 확장 가능한 무료 도구”이다.
- 프로페셔널 버전(Visual Studio Professional)은 “개별 개발자 또는 소규모 팀을 위한 전문적인 개발자 도구 및 서비스”라고 되어 있다.
- 엔터프라이즈 버전(Visual Studio Enterprise)은 “고급 테스트 및 DevOps를 포함해서 어떠한 크기나 복잡한 프로젝트까지 개발 팀을 위한 고급 기능이 포함된 엔터프라이즈급 솔루션”라고 표시되어 있다.

비주얼 스튜디오 설치

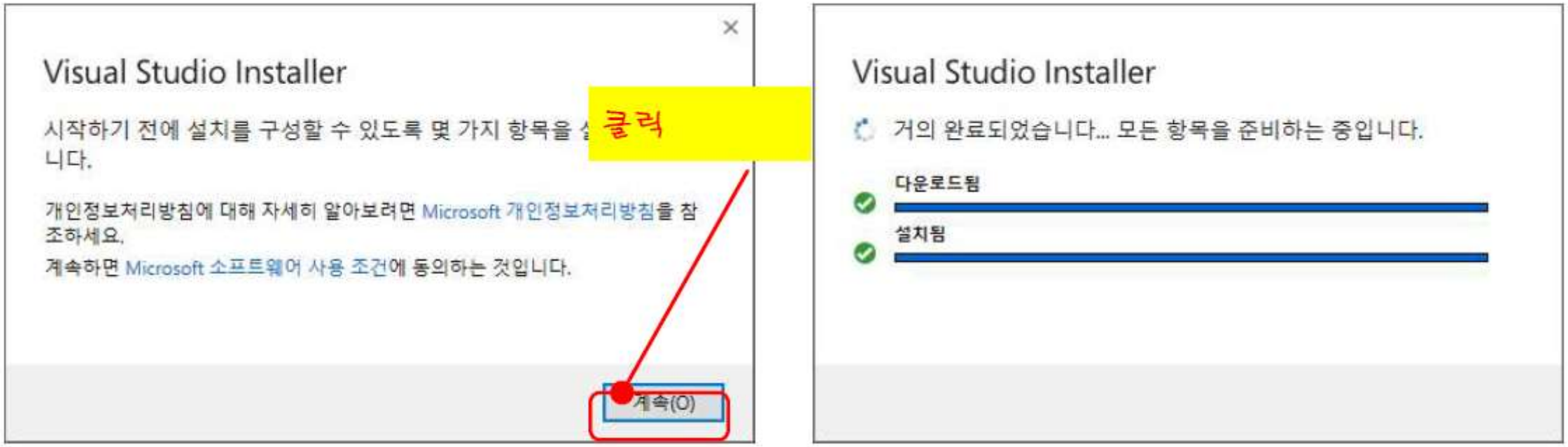
https://visualstudio.Microsoft.com/ko/ 접속



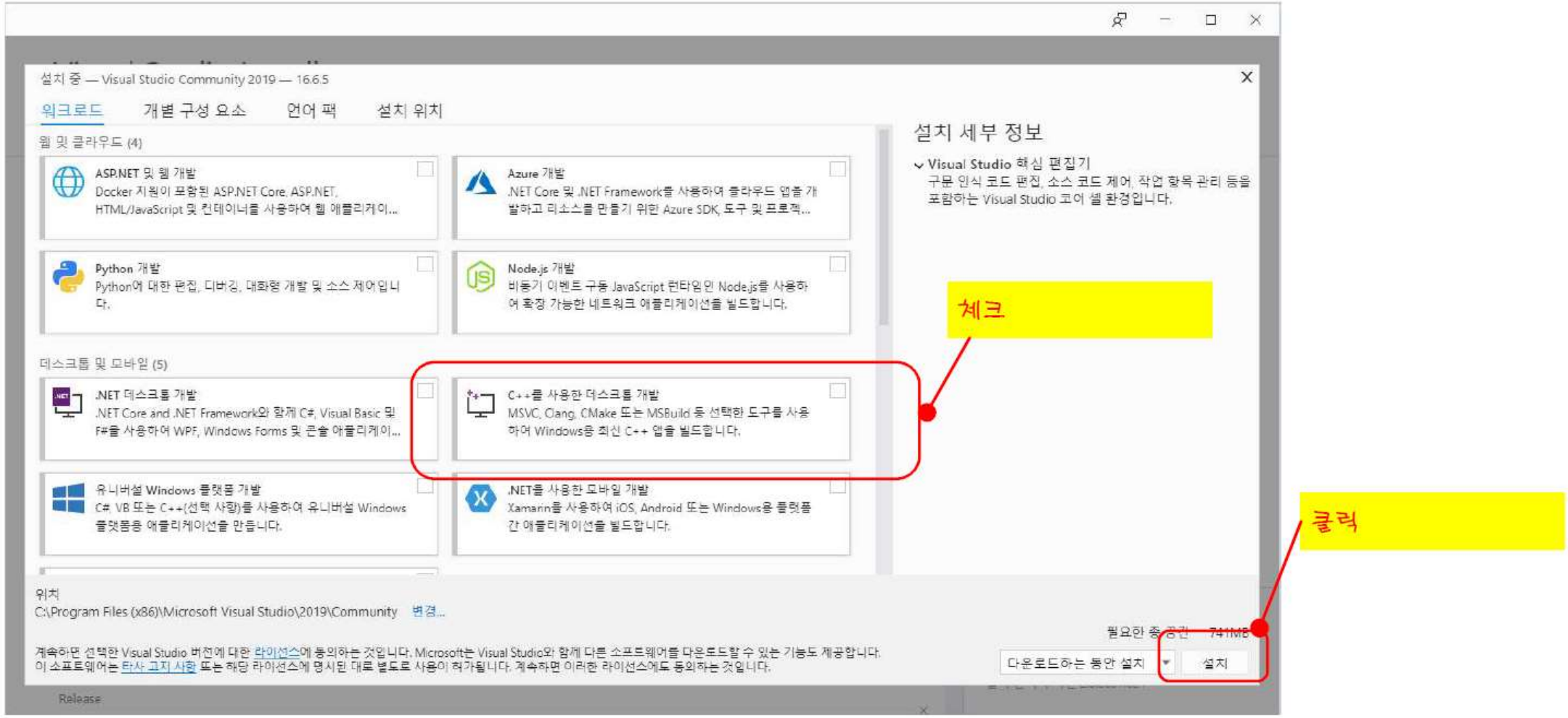
비주얼 스튜디오 설치


Visual Studio를 다운로드해 주셔
visualstudio.microsoft.com/ko/thank-you-downloading-visual-studio/?sku=Community&rel=16
Microsoft | Visual Studio
제품
다운로드
구입
자세히
무료 Visual Studio
Microsoft 전체
Visual Studio를 다운로드해 주셔서 감사합니다.
다운로드가 곧 시작됩니다. 다운로드가 시작되지 않은 경우에는
여기를 클릭하여 다시 시도하세요
클릭
용하여 개발 시작
피드백
설치 지침
설치에 대한 도움이 필요하세요?
What's New in Visual Studio 2019
vs_community__9....exe
모두 표시

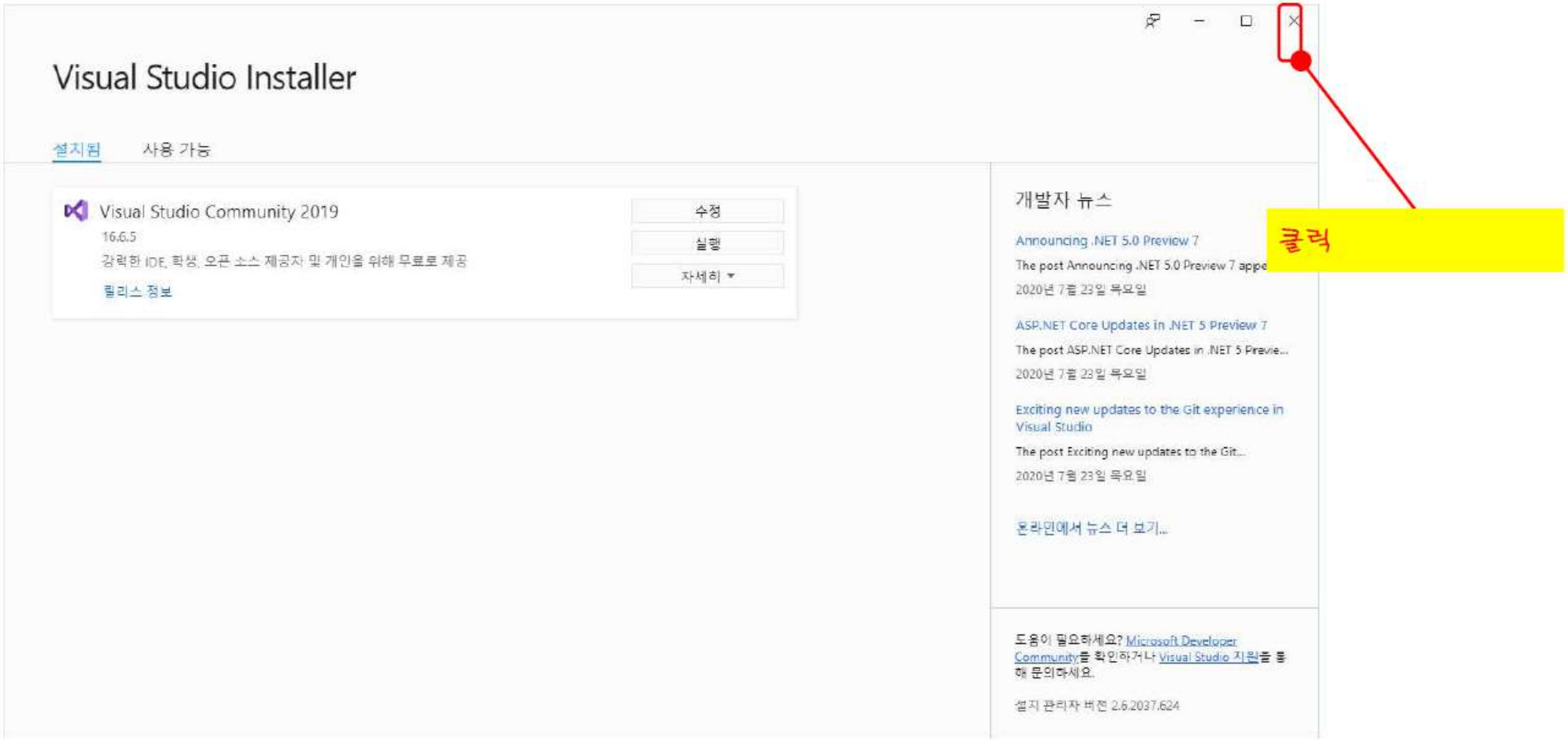
비주얼 스튜디오 설치


Visual Studio Installer
시작하기 전에 설치를 구성할 수 있도록 몇 가지 항목을 설
니다.
개인정보처리방침에 대해 자세히 알아보려면 Microsoft 개인정보처리방침을 참
조하세요.
계속하면 Microsoft 소프트웨어 사용 조건에 동의하는 것입니다.
클릭
계속(O)
Visual Studio Installer
거의 완료되었습니다... 모든 항목을 준비하는 중입니다.
다운로드됨
설치됨

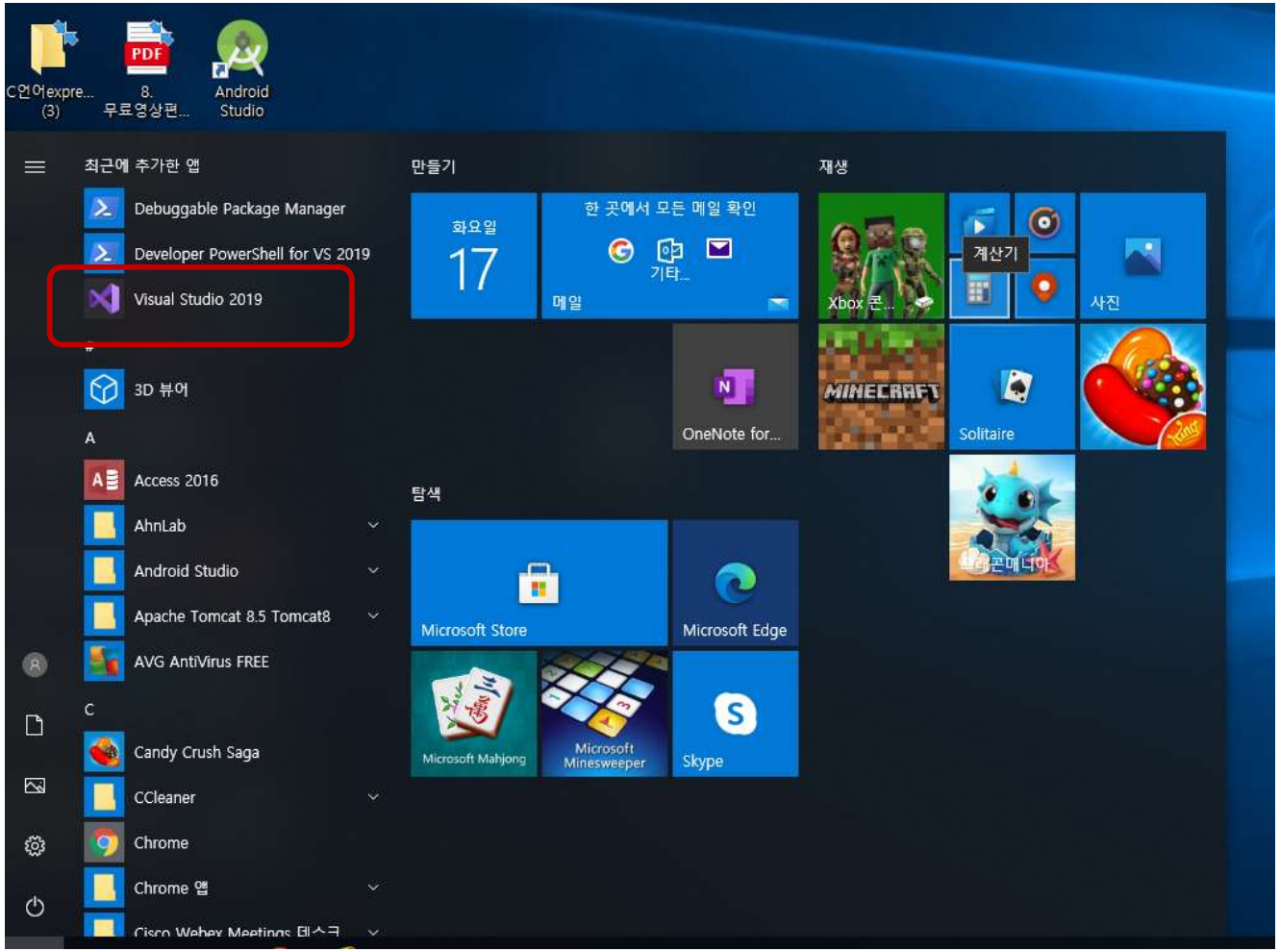
비주얼 스튜디오 설치


설치 중 — Visual Studio Community 2019 — 16.6.5
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 애플리케이...
Azure 개발
.NET Core 및 .NET Framework를 사용하여 클라우드 앱을 개발하고 리소스를 만들기 위한 Azure SDK, 도구 및 프로젝...
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 애플리케이션을 빌드합니다.
데스크톱 및 모바일 (5)
.NET 데스크톱 개발
.NET Core and .NET Framework와 함께 C#, Visual Basic 및 F#을 사용하여 WPF, Windows Forms 및 콘솔 애플리케이...
C++를 사용한 데스크톱 개발
MSVC, Clang, CMake 또는 MSBuild 등 선택한 도구를 사용하여 Windows용 최신 C++ 앱을 빌드합니다.
유니버설 Windows 플랫폼 개발
C# VB 또는 C++(선택 사항)를 사용하여 유니버설 Windows 플랫폼용 애플리케이션을 만듭니다.
.NET을 사용한 모바일 개발
Xamarin을 사용하여 iOS, Android 또는 Windows용 플랫폼 간 애플리케이션을 빌드합니다.
설치 세부 정보
Visual Studio 핵심 편집기
구문 인식 코드 편집, 소스 코드 제어, 작업 항목 관리 등을 포함하는 Visual Studio 코어 셸 환경입니다.
체크
위치
C:\Program Files (x86)\Microsoft Visual Studio\2019\Community
변경...
계속하면 선택한 Visual Studio 버전에 대한 라이선스에 동의하는 것입니다. Microsoft는 Visual Studio와 함께 다른 소프트웨어를 다운로드할 수 있는 기능도 제공합니다. 이 소프트웨어는 타사 고지 사항 또는 해당 라이선스에 명시된 대로 별도로 사용이 허가됩니다. 계속하면 이러한 라이선스에도 동의하는 것입니다.
741MB
다운로드하는 동안 설치
설치
클릭
Release

비주얼 스튜디오 설치

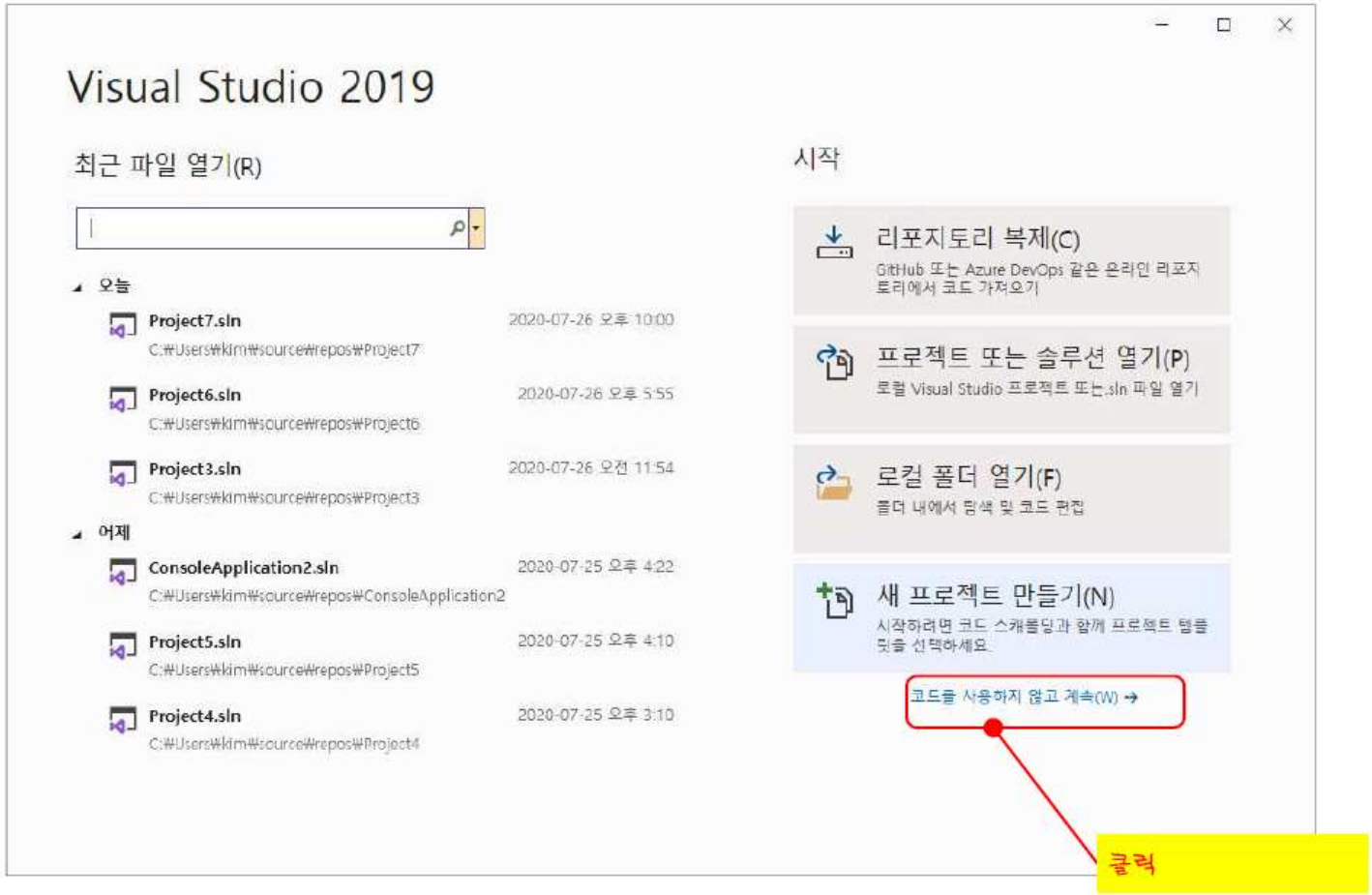

Visual Studio Installer
설치됨
사용 가능
Visual Studio Community 2019
16.6.5
강력한 IDE, 학생, 오픈 소스 제공자 및 개인을 위해 무료로 제공
릴리스 정보
수정
실행
자세히
개발자 뉴스
Announcing .NET 5.0 Preview 7
The post Announcing .NET 5.0 Preview 7 appe
2020년 7월 23일 목요일
ASP.NET Core Updates in .NET 5 Preview 7
The post ASP.NET Core Updates in .NET 5 Previe...
2020년 7월 23일 목요일
Exciting new updates to the Git experience in Visual Studio
The post Exciting new updates to the Git...
2020년 7월 23일 목요일
온라인에서 뉴스 더 보기...
도움이 필요하세요? Microsoft Developer Community를 확인하거나 Visual Studio 지원을 통해 문의하세요.
설치 관리자 버전 2.6.2037.624
클릭

비주얼 스튜디오 설치


C언어expre... (3)
8. 무료영상편...
Android Studio
최근에 추가한 앱
Debuggable Package Manager
Developer PowerShell for VS 2019
Visual Studio 2019
3D 뷰어
A
Access 2016
AhnLab
Android Studio
Apache Tomcat 8.5 Tomcat8
AVG AntiVirus FREE
C
Candy Crush Saga
CCleaner
Chrome
Chrome 앱
만들기
화요일
17
한 곳에서 모든 메일 확인
기타...
메일
OneNote for...
탐색
Microsoft Store
Microsoft Edge
Microsoft Mahjong
Microsoft Minesweeper
Skype
재생
Xbox 콘...
계산기
사진
MINECRAFT
Solitaire

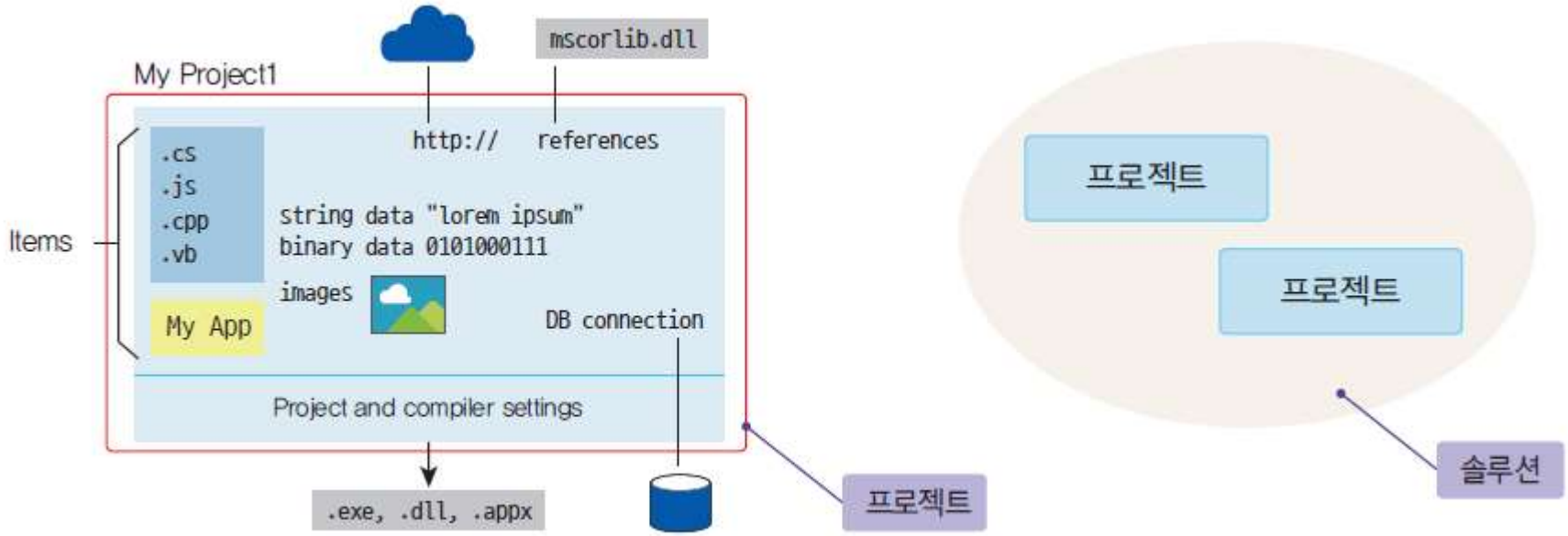
비주얼 스튜디오 시작

윈도우의 [시작] 버튼을 누르고 [Visual Studio 2019]를 찾아서 실행한다. 다음과 같은 화면이 등장한다. "코드를 사용하지 않고 계속(W)"를 클릭한다.



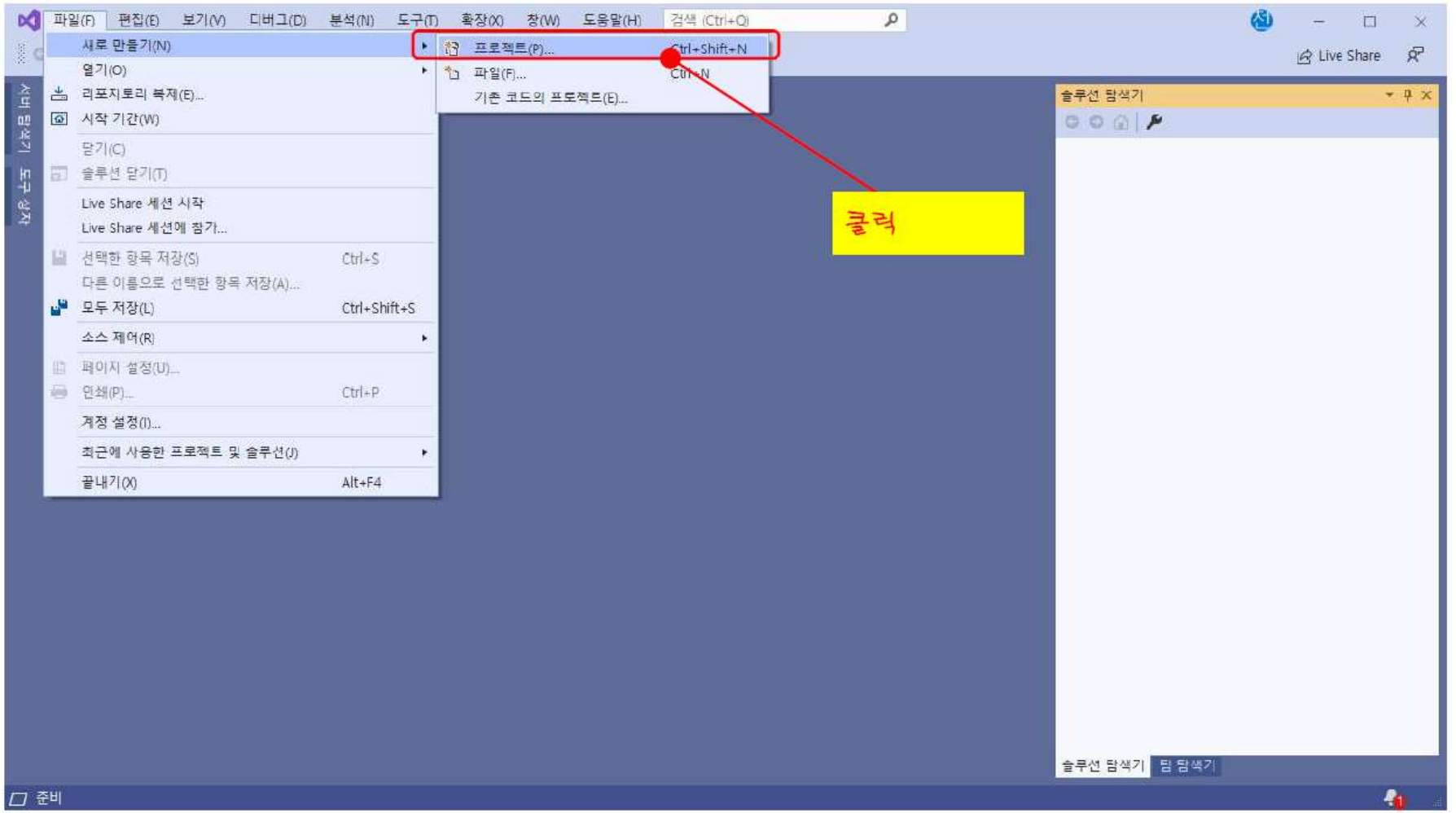
워크스페이스와 프로젝트

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



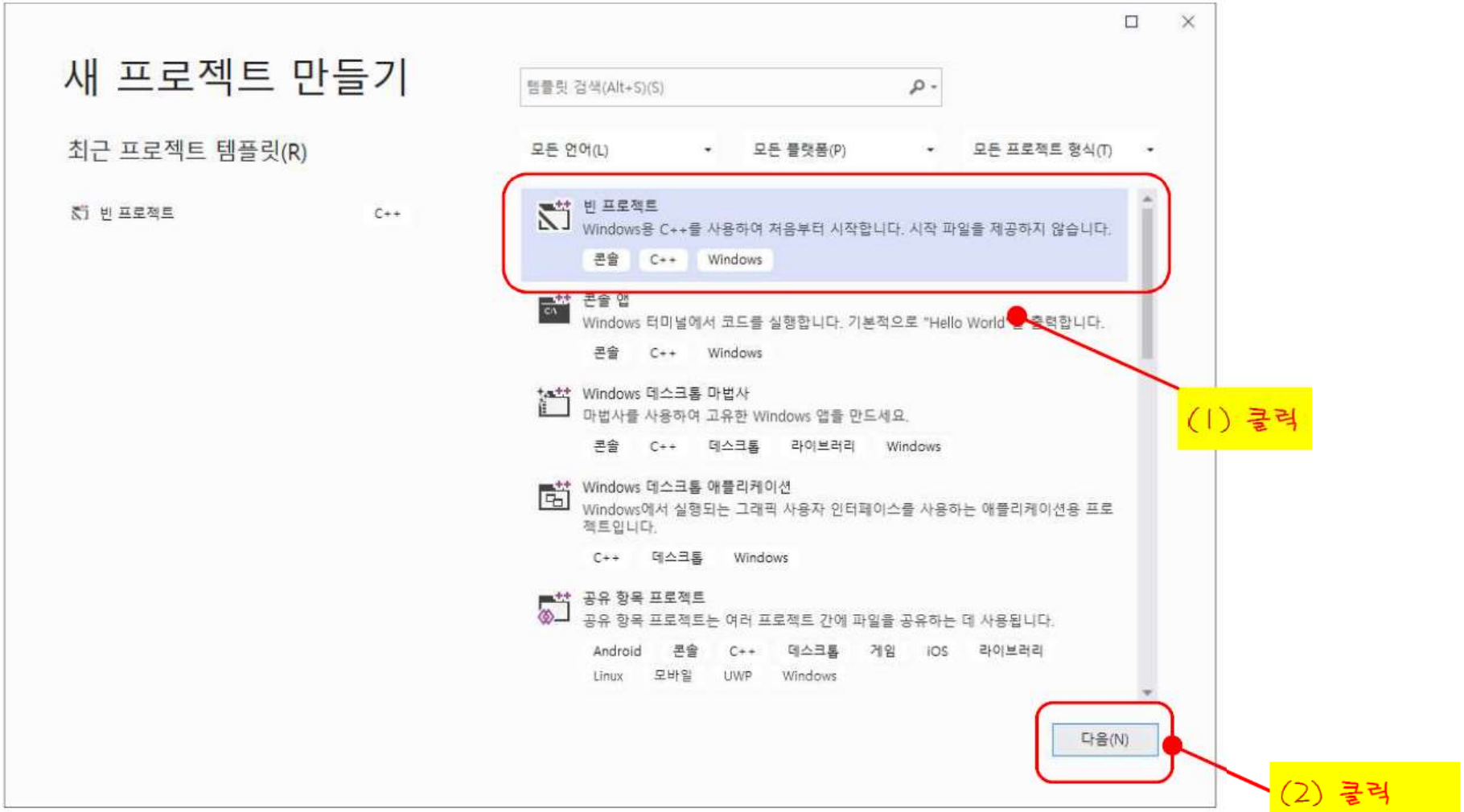
프로젝트 생성하기

(1) 새로운 프로젝트를 만들려면 [파일]->[새로 만들기]->[프로젝트] 메뉴를 선택한다.



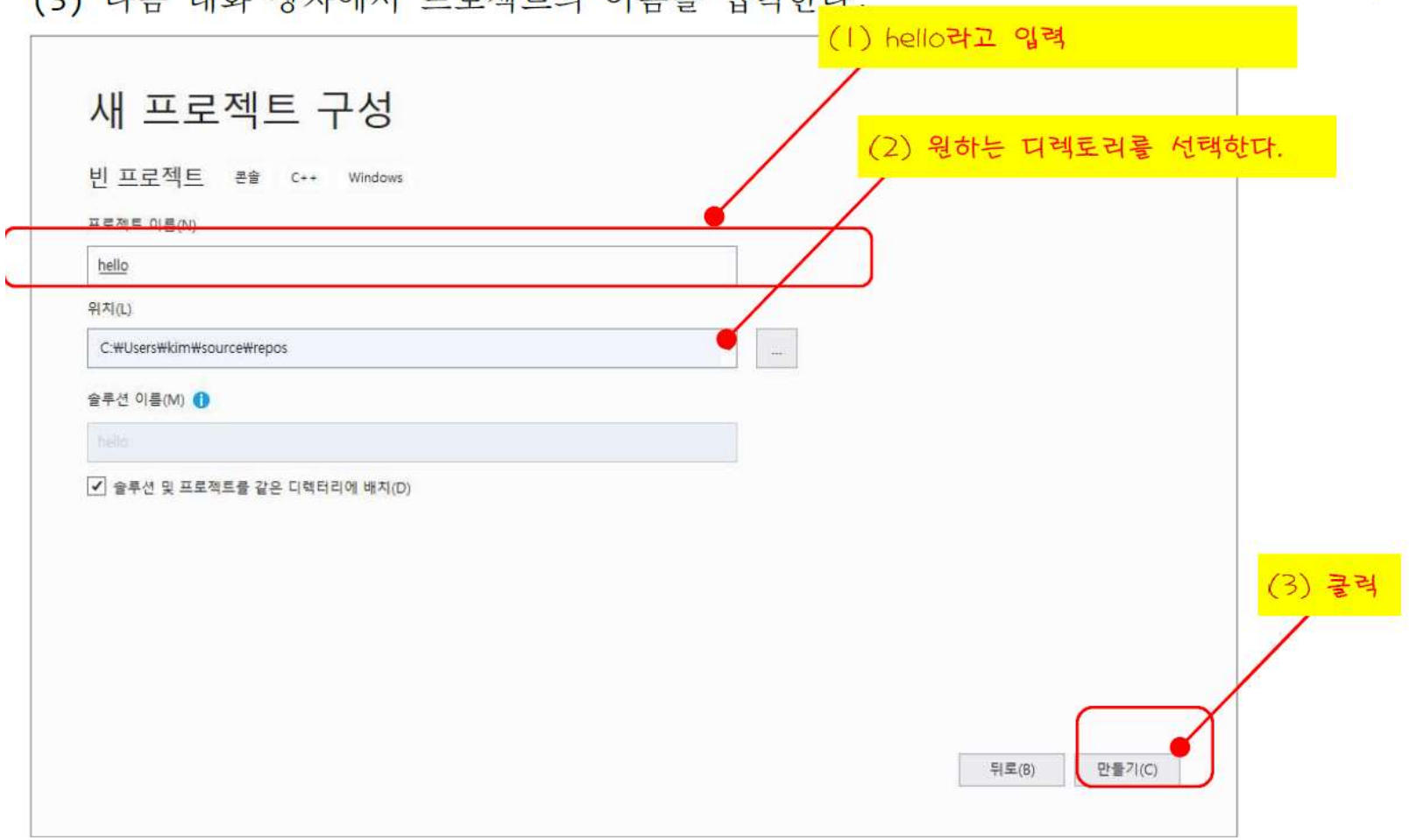
프로젝트 생성하기

(2) 대화 상자에서 "빈 프로젝트"를 선택한다.

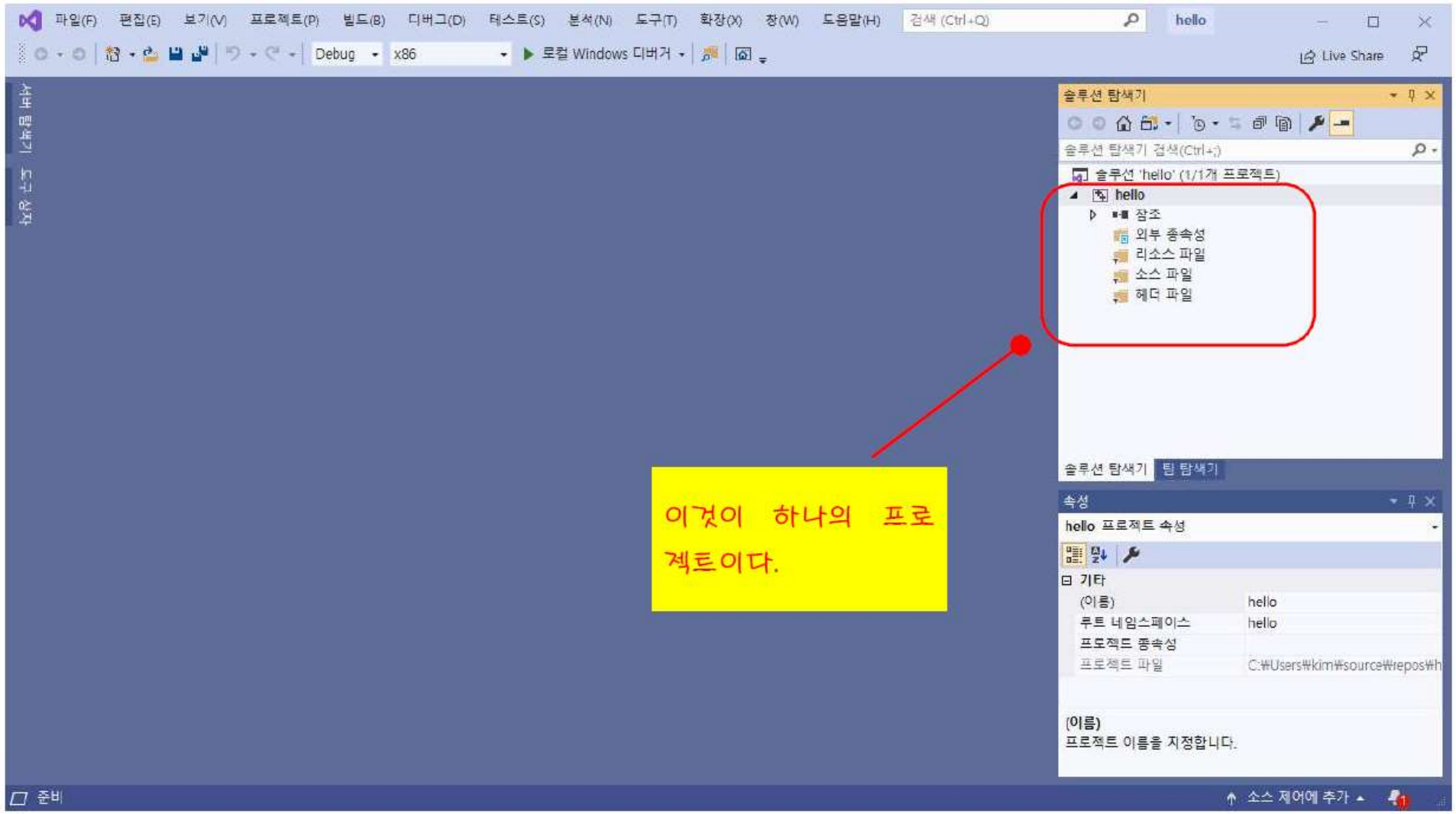


프로젝트 생성하기

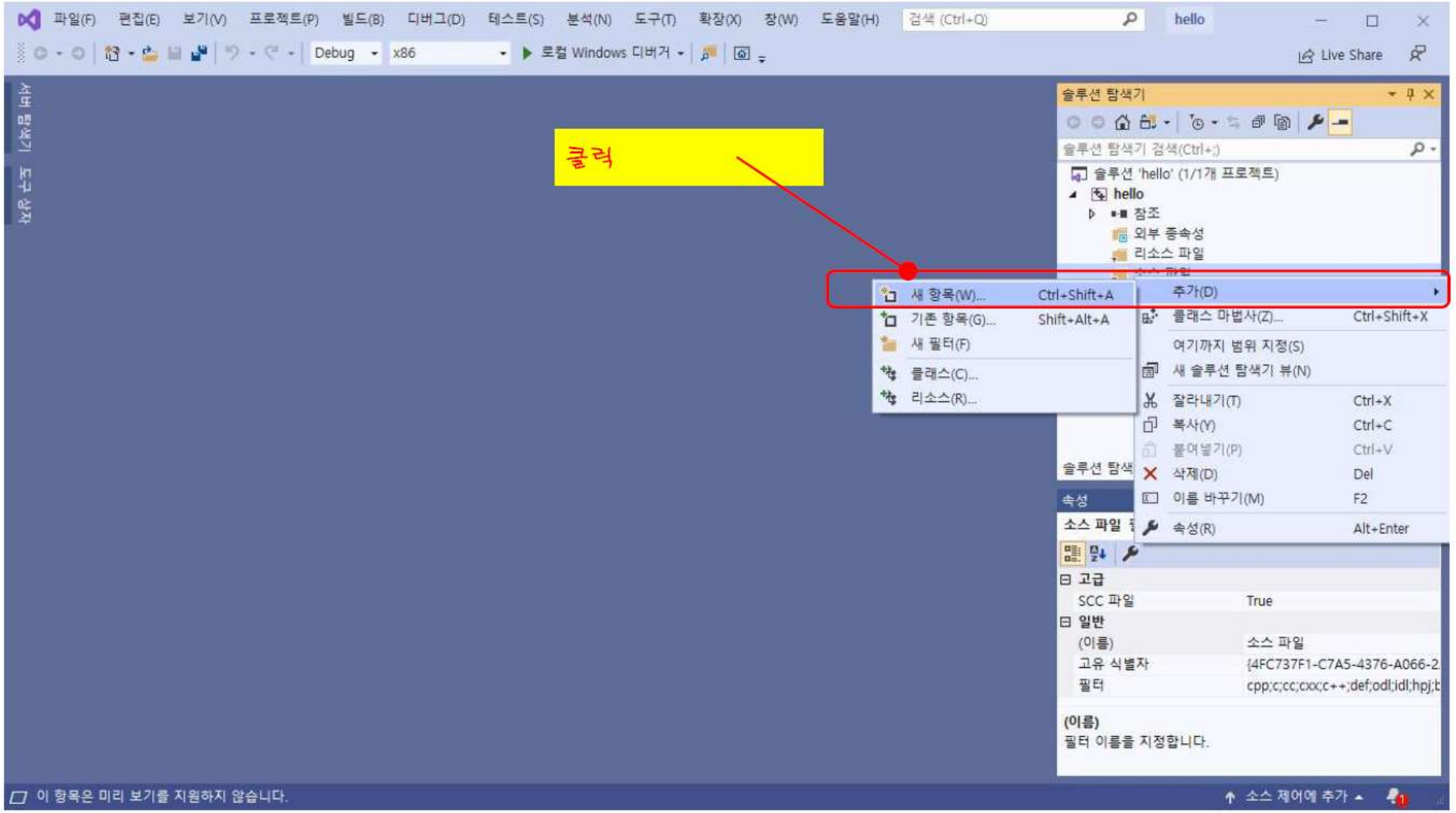
(3) 다음 대화 상자에서 프로젝트의 이름을 입력한다.



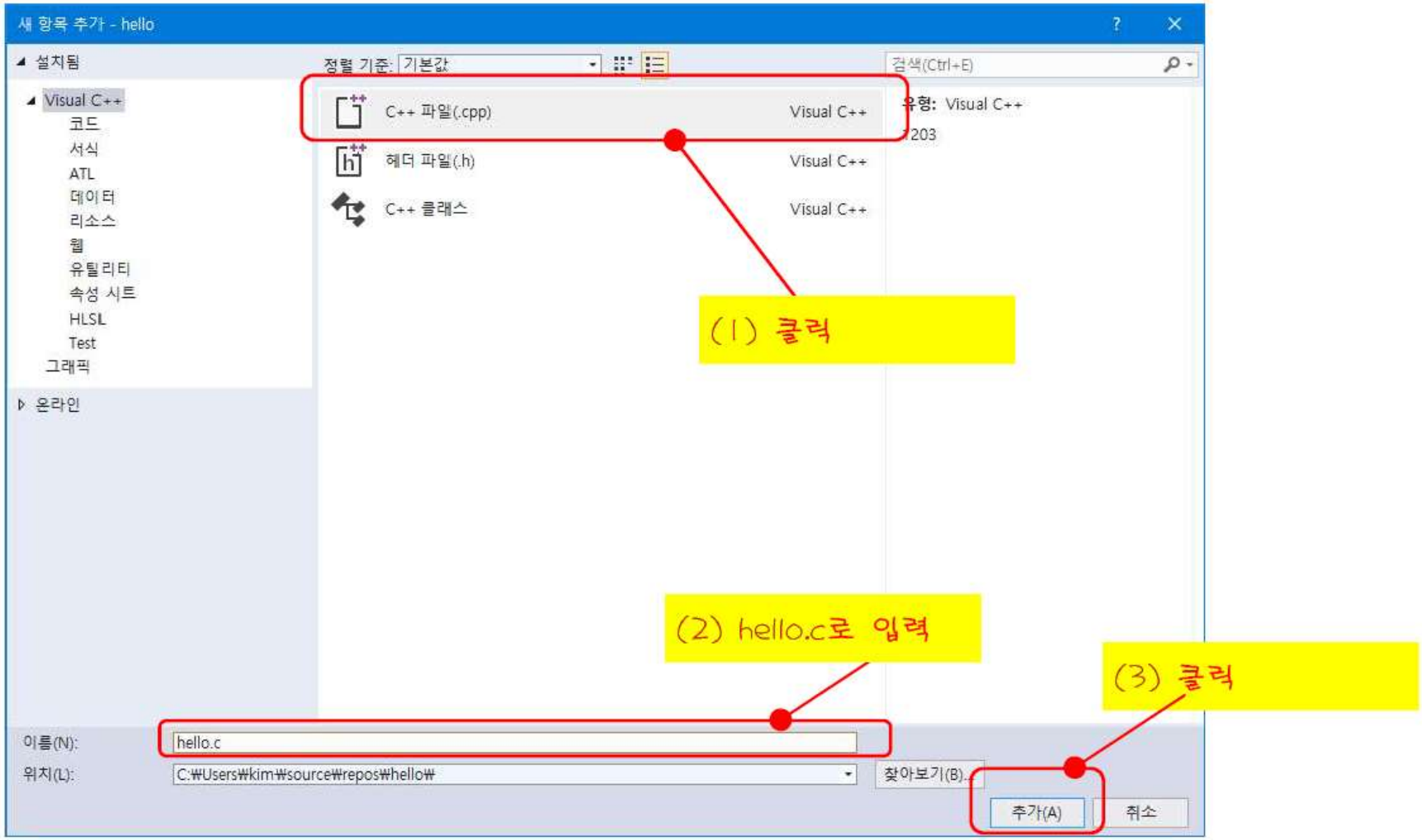
프로젝트 생성하기


파일(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
Live Share
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'hello' (1/1개 프로젝트)
hello
참조
외부 종속성
리소스 파일
소스 파일
헤더 파일
솔루션 탐색기
팀 탐색기
속성
hello 프로젝트 속성
기타
(이름) hello
루트 네임스페이스 hello
프로젝트 종속성
프로젝트 파일 C:₩Users₩kim₩source₩repos₩h
(이름)
프로젝트 이름을 지정합니다.
이것이 하나의 프로젝트이다.
준비
소스 제어에 추가

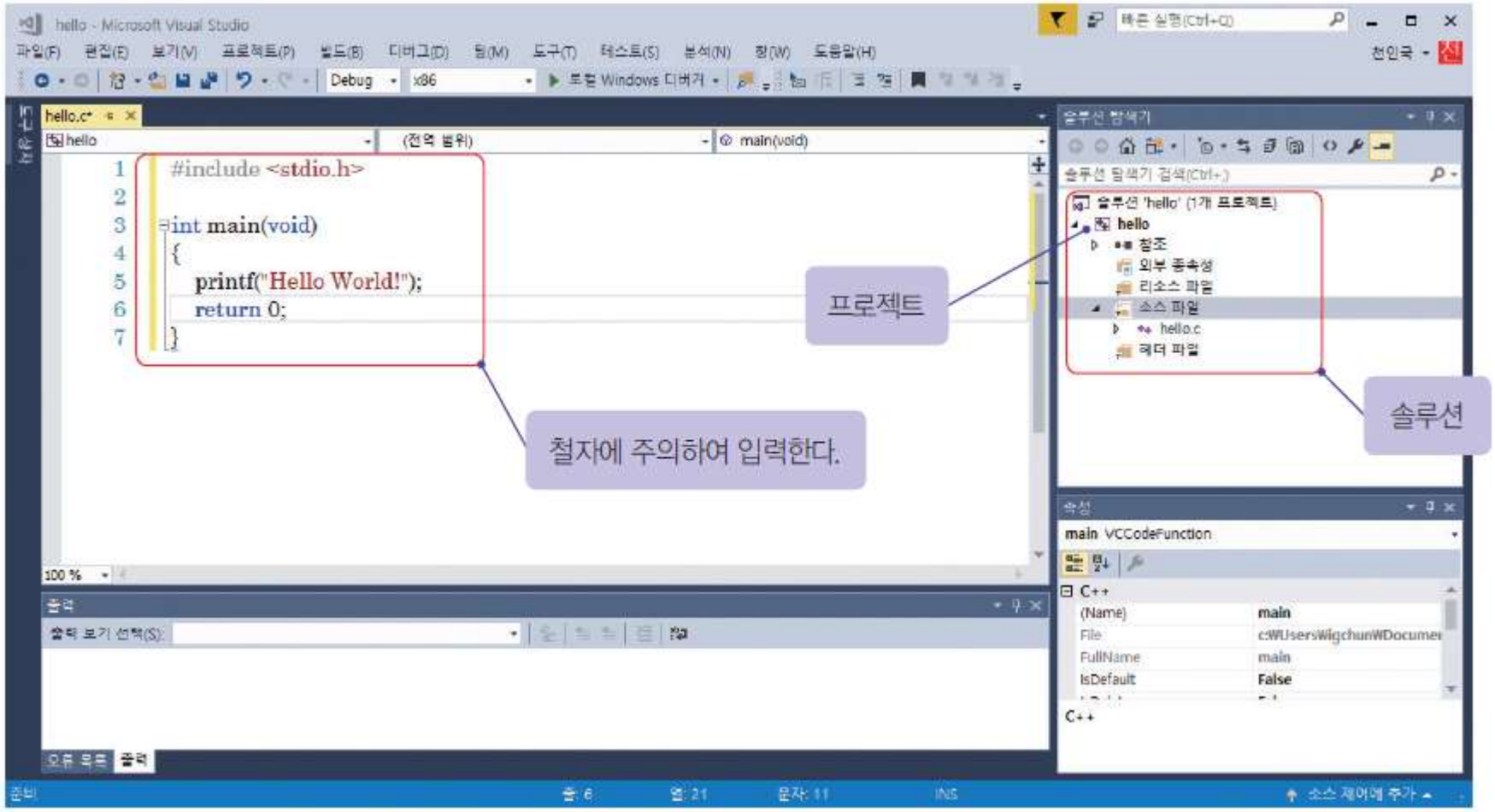
소스 파일 생성하기


파일(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
Live Share
서버 탐색기
도구 상자
클릭
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'hello' (1/1개 프로젝트)
hello
참조
외부 종속성
리소스 파일
새 항목(W)... Ctrl+Shift+A
기존 항목(G)... Shift+Alt+A
새 필터(F)
클래스(C)...
리소스(R)...
추가(D)
클래스 마법사(Z)... Ctrl+Shift+X
여기까지 범위 지정(S)
새 솔루션 탐색기 뷰(N)
잘라내기(T) Ctrl+X
복사(Y) Ctrl+C
붙여넣기(P) Ctrl+V
삭제(D) Del
이름 바꾸기(M) F2
속성(R) Alt+Enter
솔루션 탐색
속성
소스 파일
고급
SCC 파일 True
일반
(이름) 소스 파일
고유 식별자 {4FC737F1-C7A5-4376-A066-2
필터 cpp;c;cc;cxx;c++;def;odl;idl;hpj;
(이름)
필터 이름을 지정합니다.
이 항목은 미리 보기를 지원하지 않습니다.
소스 제어에 추가

소스 파일 생성하기


새 항목 추가 - hello
설치됨
Visual C++
코드
서식
ATL
데이터
리소스
웹
유틸리티
속성 시트
HLSL
Test
그래픽
온라인
정렬 기준: 기본값
검색(Ctrl+E)
C++ 파일(.cpp) Visual C++
헤더 파일(.h) Visual C++
C++ 클래스 Visual C++
유형: Visual C++
(1) 클릭
(2) hello.c로 입력
(3) 클릭
이름(N): hello.c
위치(L): C:₩Users₩kim₩source₩repos₩hello₩
찾아보기(B)...
추가(A)
취소

프로그램 입력


hello - Microsoft Visual Studio
hello.c*
hello
(전역 범위)
main(void)
1 #include <stdio.h>
2
3 int main(void)
4 {
5 printf("Hello World!");
6 return 0;
7 }
프로젝트
솔루션
철자에 주의하여 입력한다.
솔루션 탐색기
솔루션 'hello' (1개 프로젝트)
hello
참조
외부 종속성
리소스 파일
소스 파일
hello.c
헤더 파일
main VCCodeFunction
C++
(Name) main
FullName main
IsDefault False

프로그램 입력

include나 stdio는 붙여쓴다.

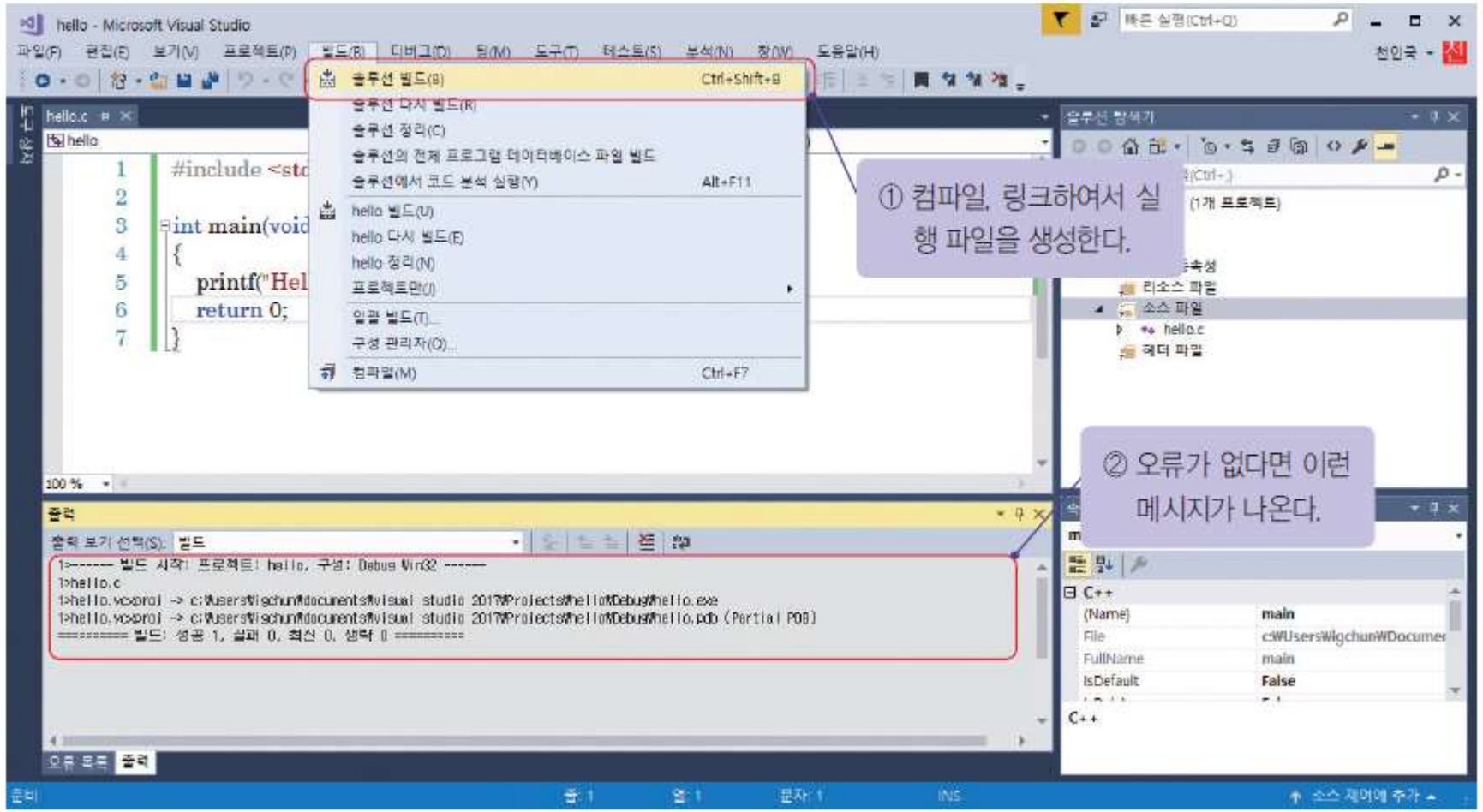
기호는 정확하게 입력

```
#include <stdio.h>

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

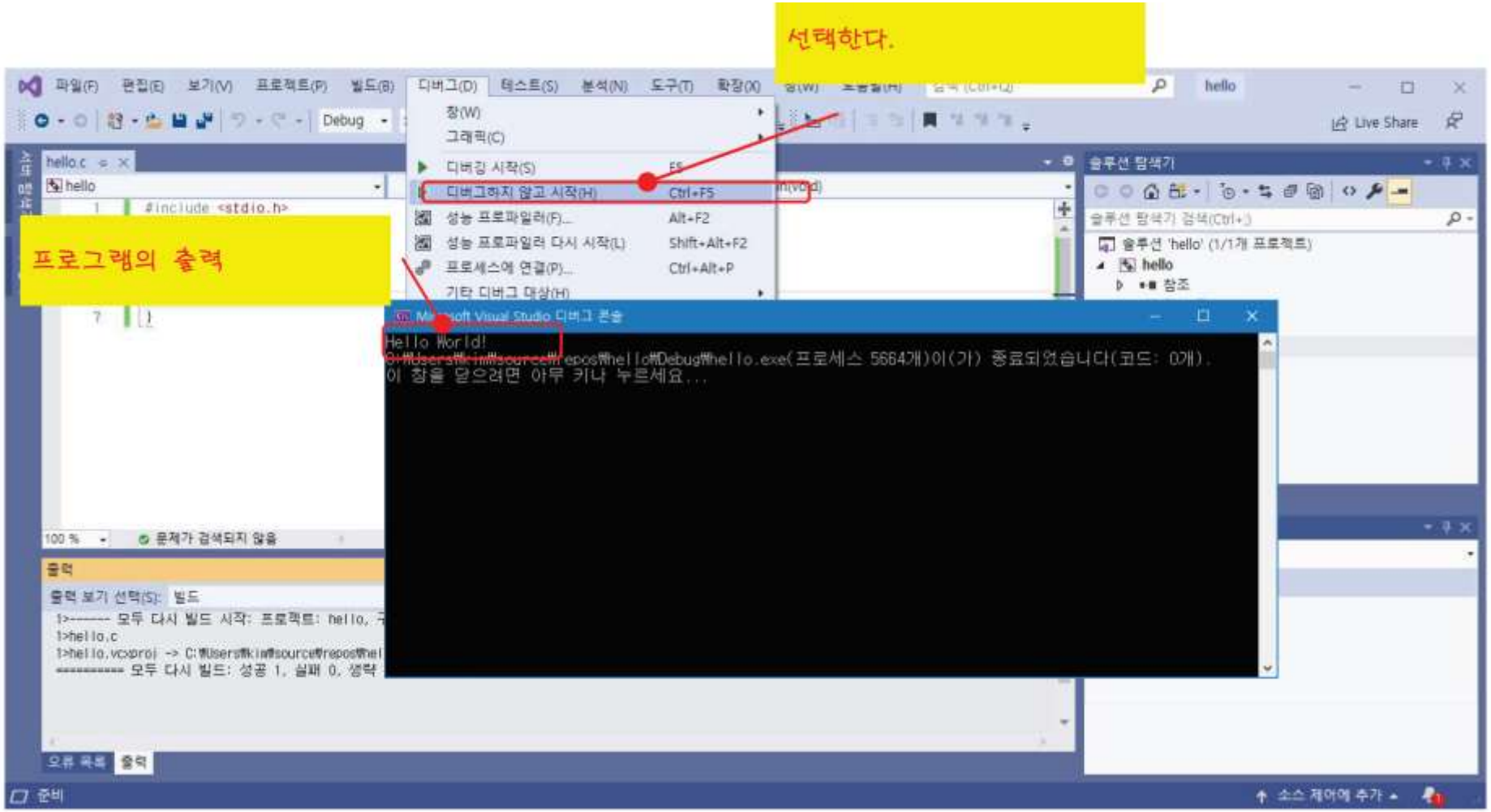
들여쓴다.

문장의 끝에는 세미콜론

컴파일하기


hello - Microsoft Visual Studio
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)
hello 다시 빌드(E)
hello 정리(N)
프로젝트만(J)
일괄 빌드(T)...
구성 관리자(O)...
컴파일(M) Ctrl+F7
① 컴파일, 링크하여서 실행 파일을 생성한다.
② 오류가 없다면 이런 메시지가 나온다.
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 ==========

프로그램 실행 하기


선택한다.
프로그램의 출력
디버그하지 않고 시작(H) Ctrl+F5
Hello World!
이 창을 닫으려면 아무 키나 누르세요...

첫번째 프로그램의 설명


```c
#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>

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

프로그램

- `#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

헤더 파일 포함

- #include는 소스 코드 안에 특정 파일을 현재의 위치에 포함

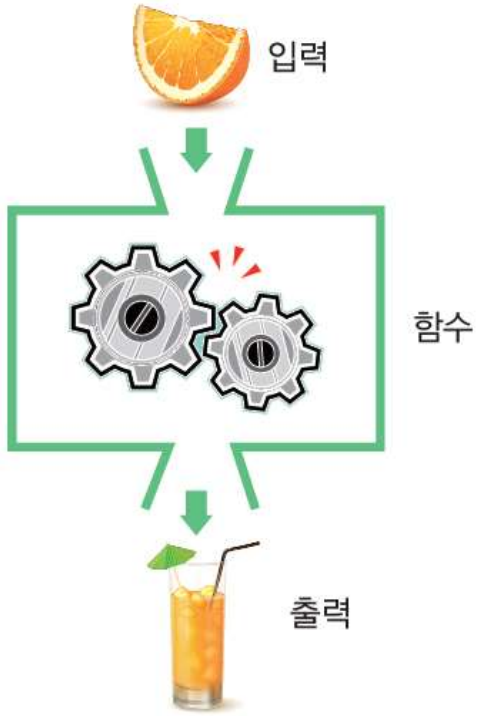
- 주의!: 전처리기 지시자 문장 끝에는 세미콜론(;)을 붙이면 안 된다.

```
#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) 수학적인 함수 $y = x^2 + 1$
- 프로그램 = 함수의 집합



main() 함수

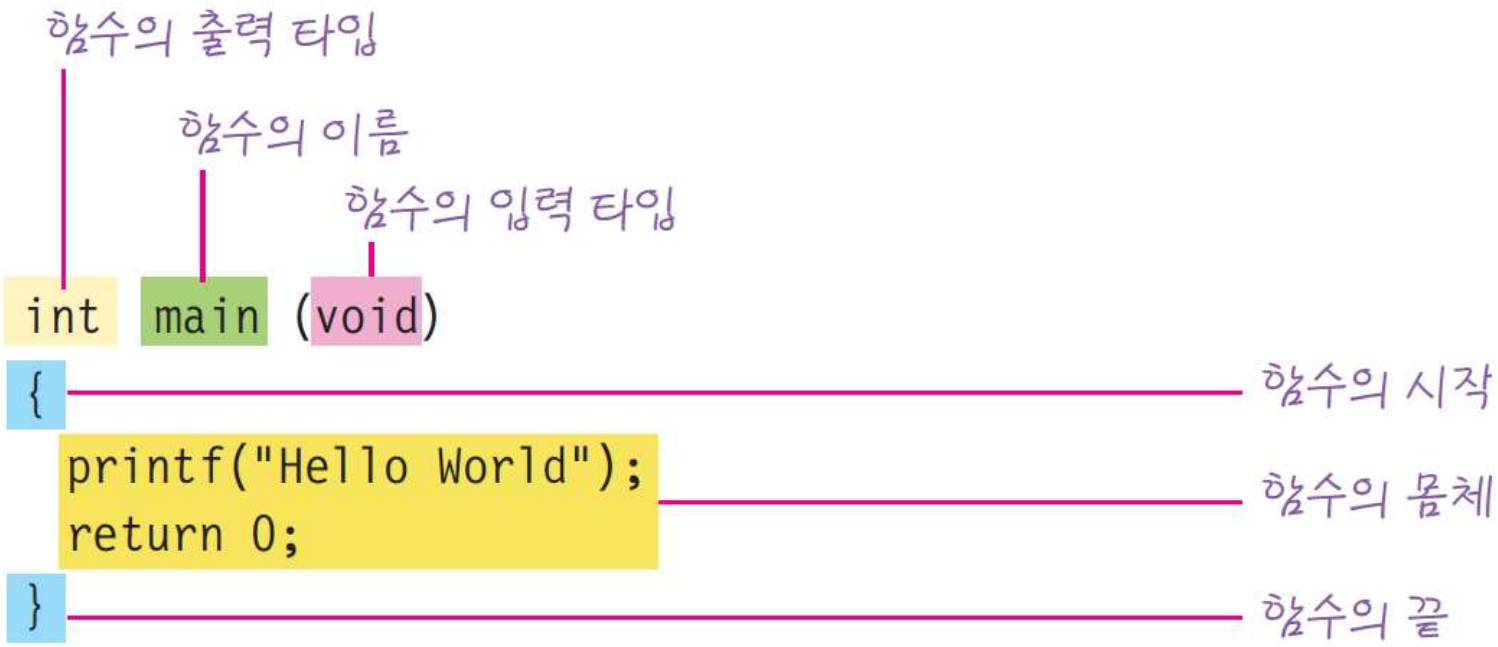
- 함수의 입력은 없음!

- 함수의 반환값

```
int main(void)
```

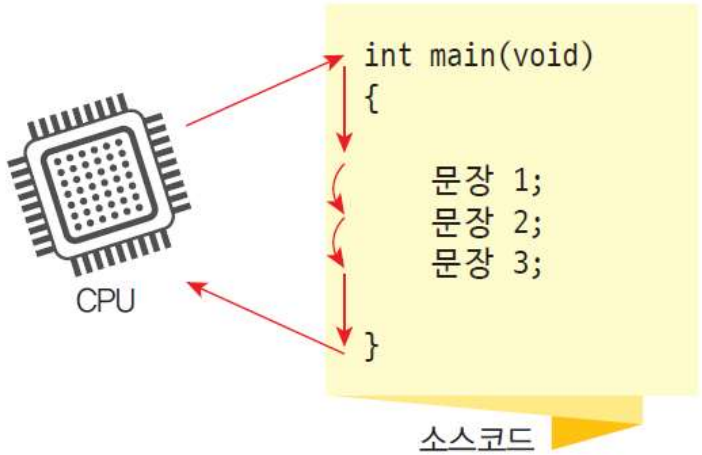
- 함수의 이름
- main()은 가장 먼저 수행되는 함수

함수의 간략한 설명


함수의 출력 타입
함수의 이름
함수의 입력 타입
int main (void)
{
printf("Hello World");
return 0;
}
함수의 시작
함수의 몸체
함수의 끝

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!");
```

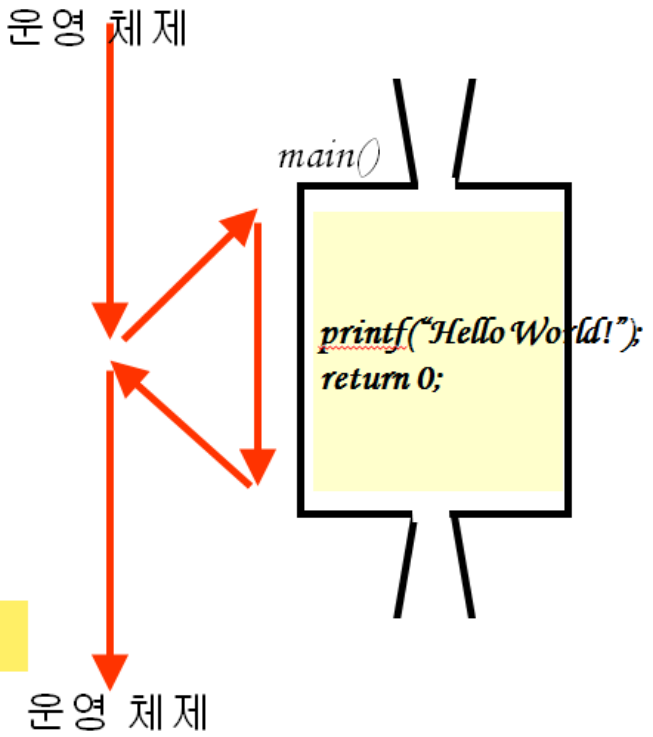
- 큰따옴표 안의 문자열이 화면에 출력된다.

함수의 반환값

- return은 함수의 결과값을 외부로 반환

```
return 0;
```

- 반환값은 0



중간 점검

- 문장의 끝에 추가하여야 하는 기호는?
- C프로그램에 반드시 있어야 하는 함수는?
- printf()가 하는 기능은 무엇인가?

응용 프로그램 #1

- 다음과 같은 출력을 가지는 프로그램을 제작하여 보자.

첫번째 버전

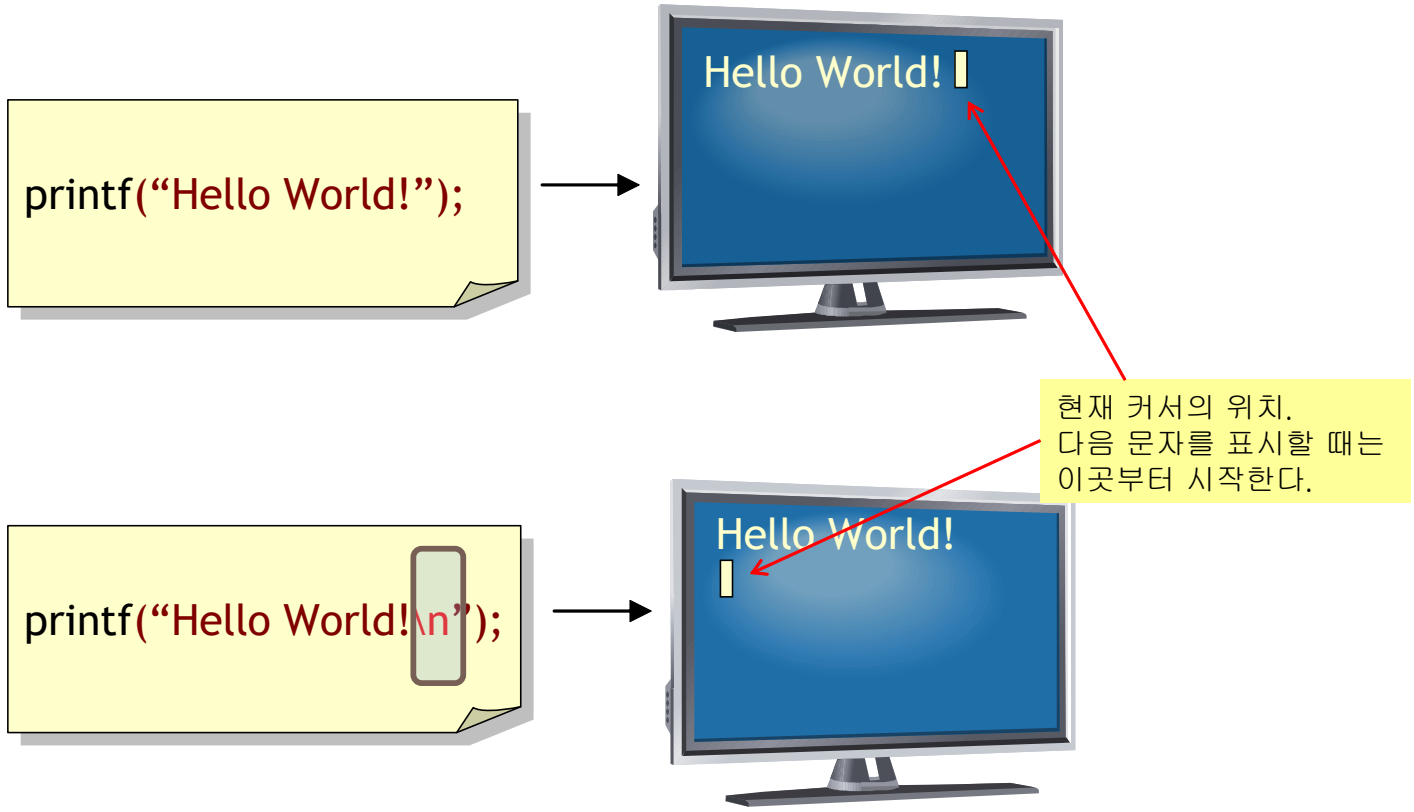
- 문장들은 순차적으로 실행된다는 사실 이용

```
#include <stdio.h>

int main(void)
{
  printf("Hello World!");
  printf("Kim ChulSoo");

  return 0;
}
```

- 2개의 문장은 순차적으로 실행된다

줄바꿈 문자

- 줄바꿈 문자인 ₩n은 화면에서 커서는 다음줄로 이동하게 한다.



줄바꿈 문자 2개를 사용하면?


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

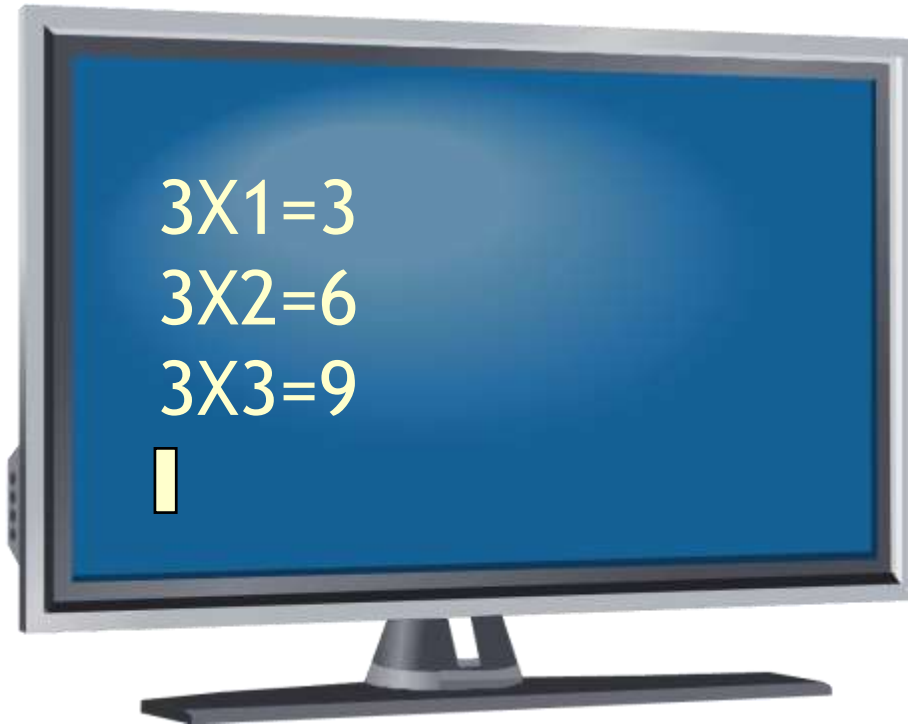
변경된 프로그램

- 줄바꿈 문자를 포함하면 우리가 원하던 결과가 된다.

```
#include <stdio.h>

int main(void)
{

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

  return 0;
}
```

응용 프로그램 #2

- 다음과 같은 출력을 가지는 프로그램을 제작하여 보자.



응용 프로그램

- 역시 문장들은 순차적으로 수행된다는 점을 이용한다.

```c
#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.

중간 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

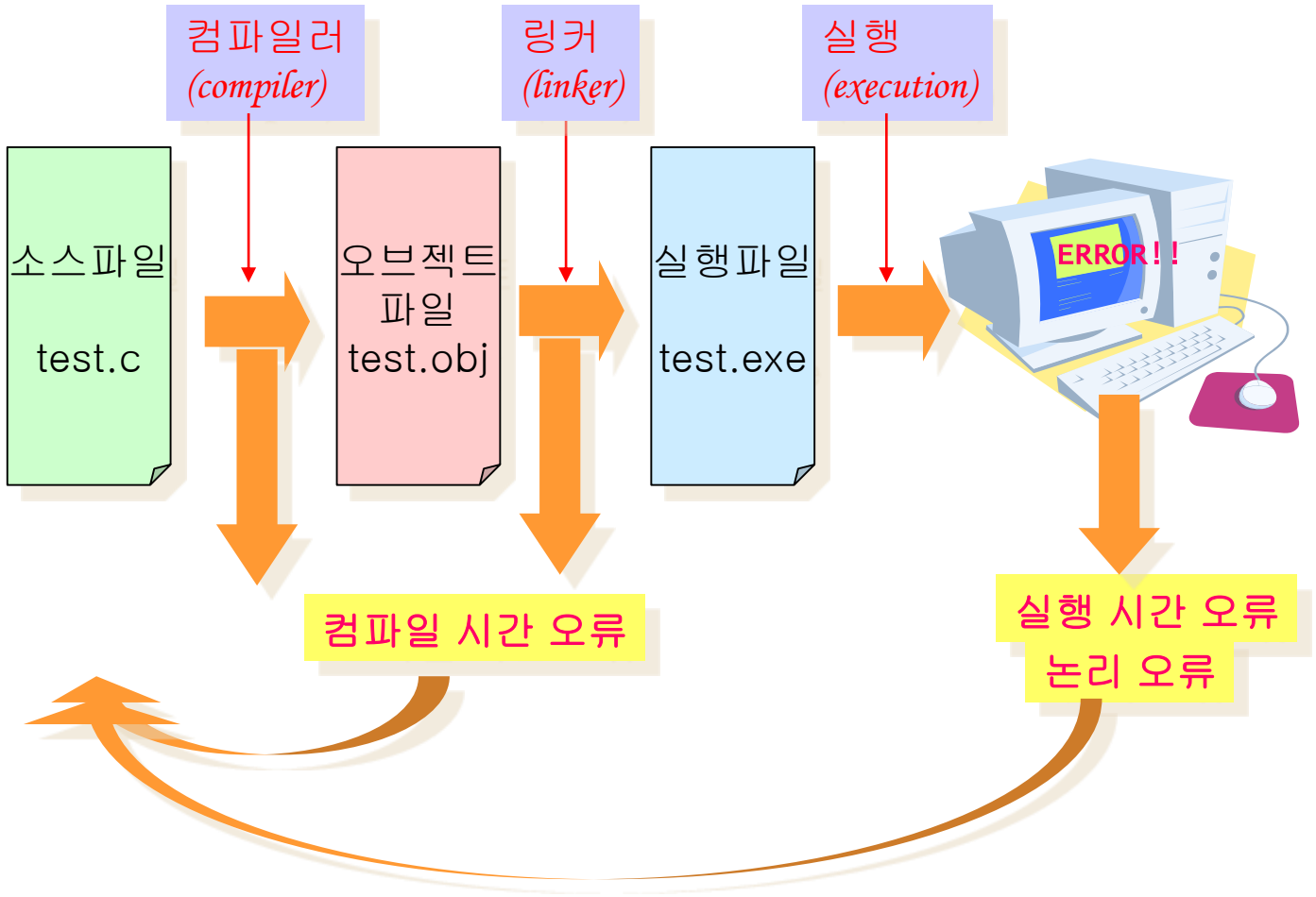
- 컴파일이나 실행 시에 오류가 발생할 수 있다.

- 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


컴파일러
(compiler)
링커
(linker)
실행
(execution)
소스파일
test.c
오브젝트
파일
test.obj
실행파일
test.exe
ERROR!!
컴파일 시간 오류
실행 시간 오류
논리 오류

오류 #1

add - Microsoft Visual Studio

error1.c*

```
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

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

오류가 발견된 소스 파일

오류가 발견된 줄번호

오류 #2

add - Microsoft Visual Studio

error3.c

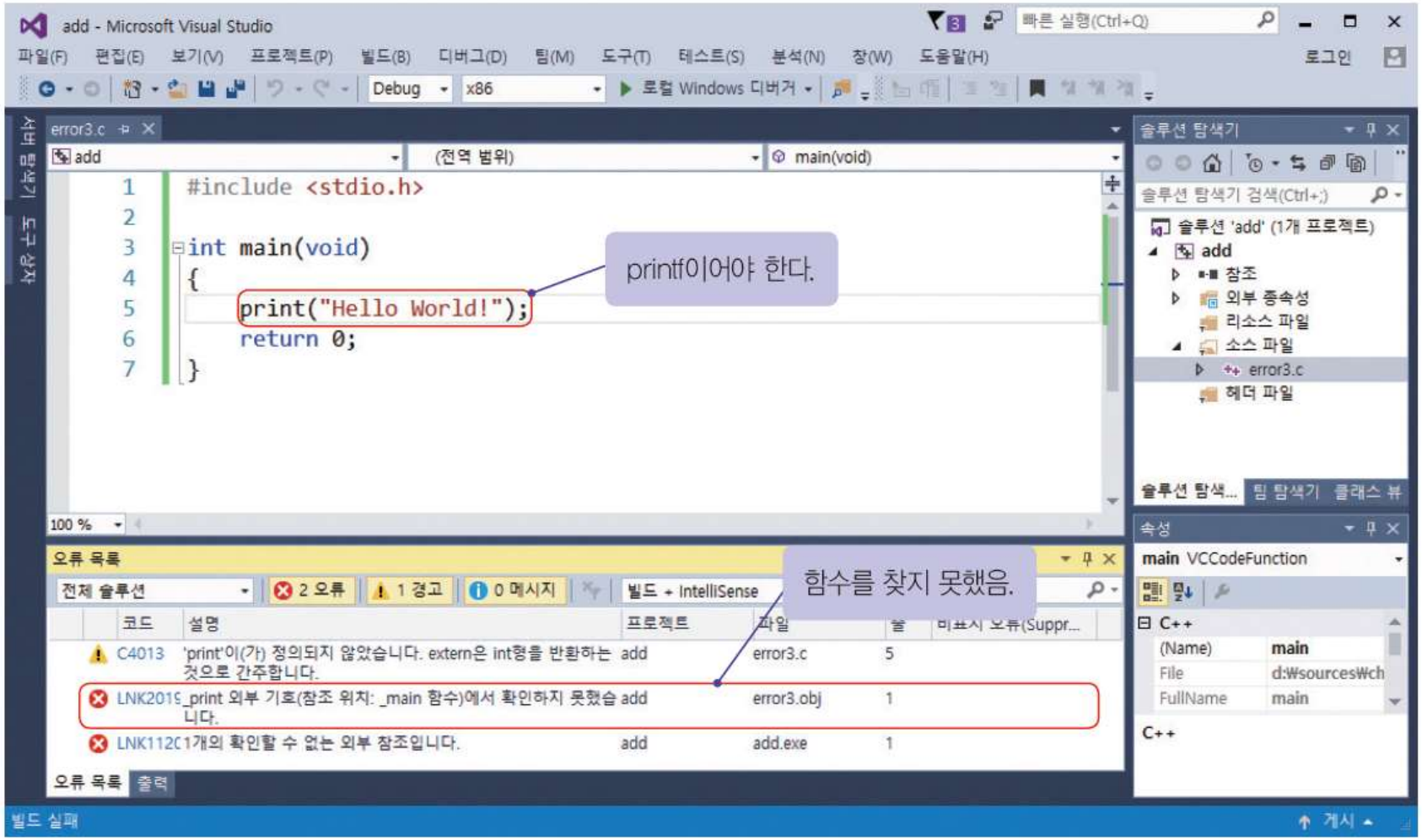
```
1  #include <stdio.h>
2
3  int main(void)
4  {
5      printf(Hello World!);
6      return 0;
7  }
```

문자열을 표시할 때, "와 "을 빠뜨렸다.

오류 목록

코드	설명	프로젝트	파일	줄
	식별자 "Hello"이(가) 정의되어 있지 않습니다.	add	error3.c	5
	')'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 #3


add - Microsoft Visual Studio
error3.c
#include <stdio.h>
int main(void)
{
print("Hello World!");
return 0;
}
printf이어야 한다.
함수를 찾지 못했음.
오류 목록
C4013 'print'이(가) 정의되지 않았습니다. extern은 int형을 반환하는 것으로 간주합니다.
LNK2019 _print 외부 기호(참조 위치: _main 함수)에서 확인하지 못했습니다.
LNK1120 1개의 확인할 수 없는 외부 참조입니다.
솔루션 탐색기
add
참조
외부 종속성
리소스 파일
소스 파일
error3.c
헤더 파일
속성

논리 오류

- 다음과 같은 출력을 가지는 프로그램을 작성하여 보자.

논리 오류가 존재하는 프로그램


```c
#include <stdio.h>

int main(void)
{
        printf("Hey!");
        printf("Good Morning");
        return 0;
}
```

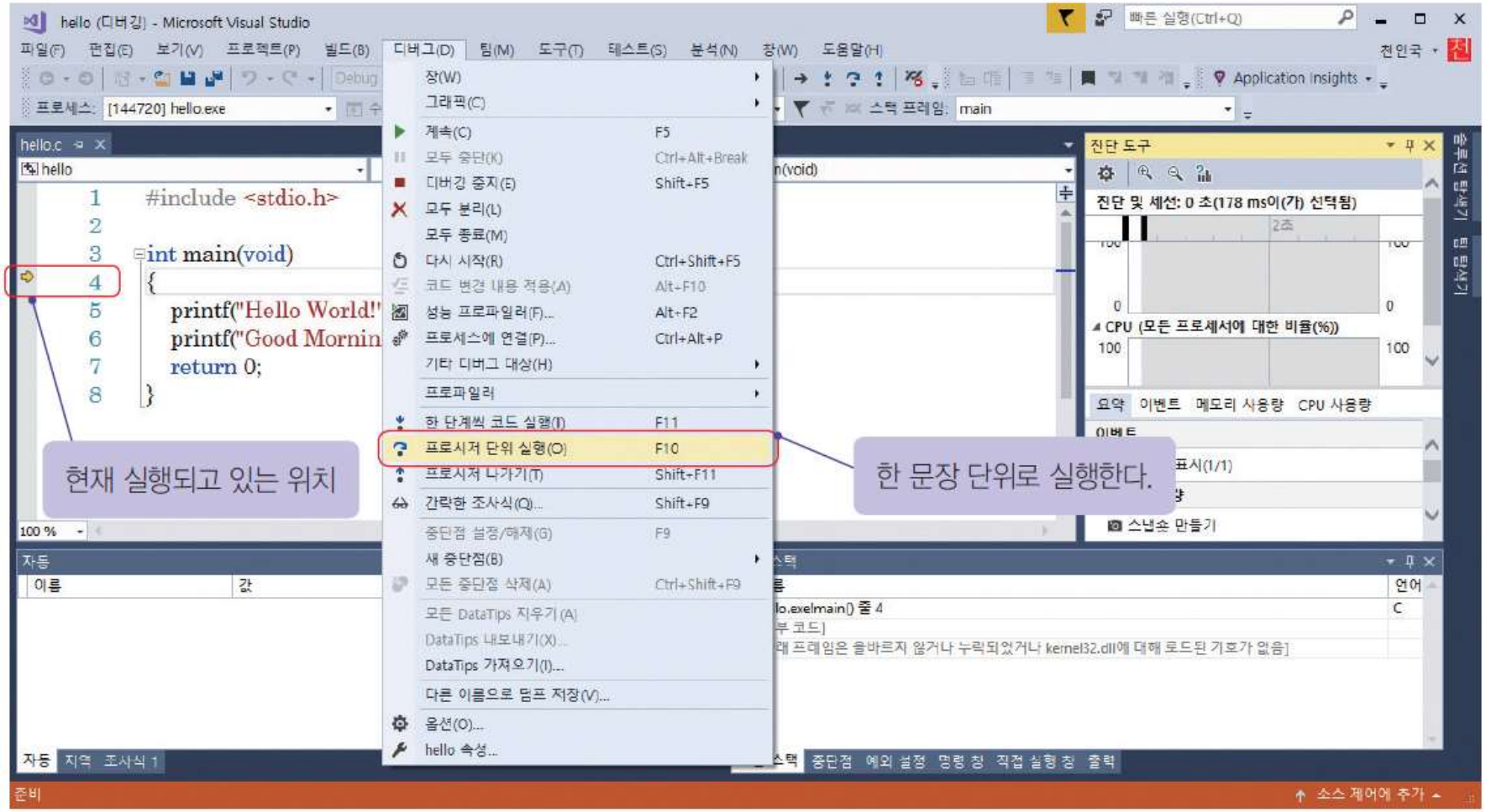
논리 오류가 수정된 프로그램


```c
#include <stdio.h>

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

디버깅

- 디버깅: 논리 오류를 찾는 과정

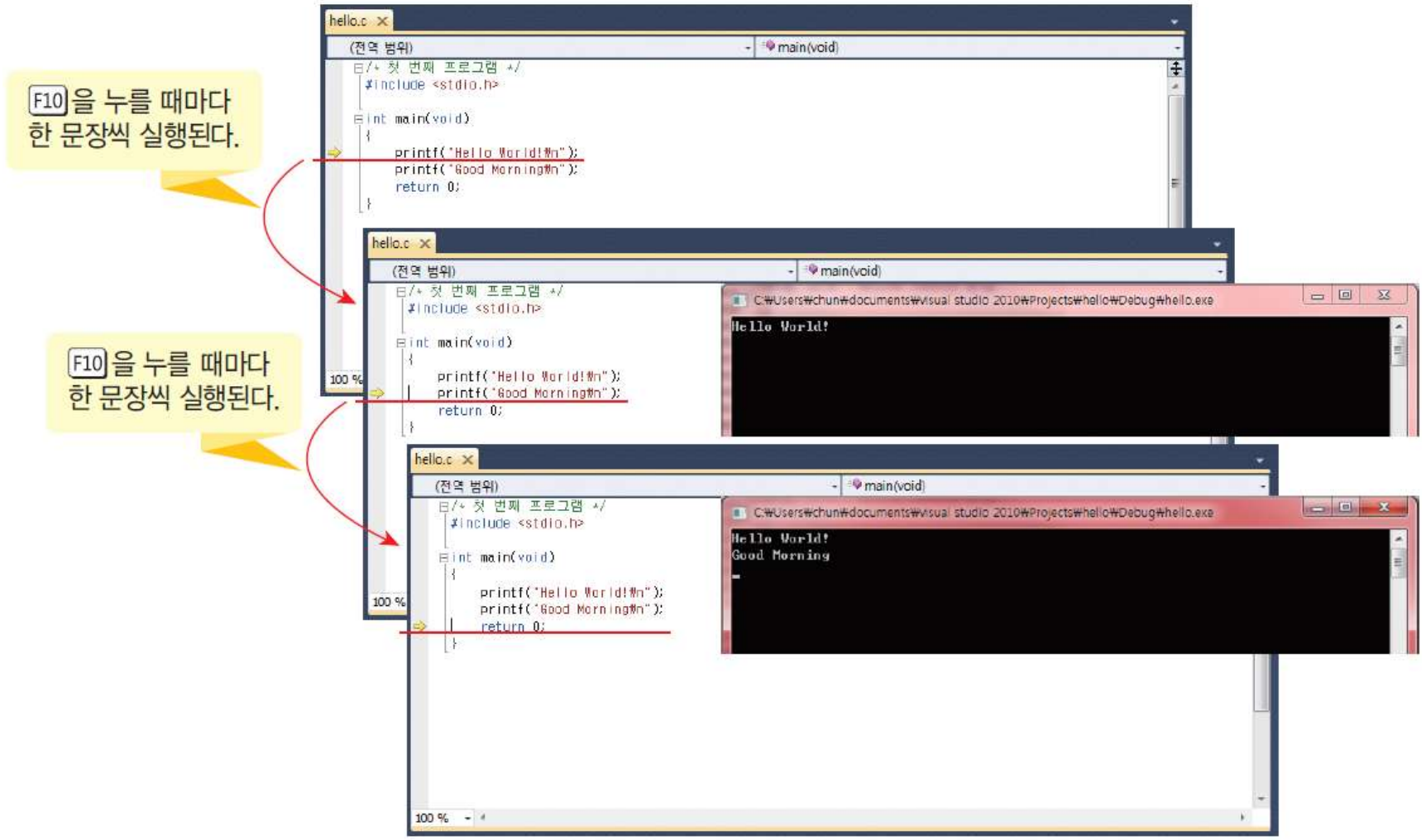
디버거(debugger)



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
(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!
Hello World!
Good Morning

중간 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!

```c
#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