



ALOHA

#1주차(1)

Visual Studio 소개

C 문법을 통한 기초 I/O

#CH.1

Visual Studio 소개 및 설치!





Visual Studio?

알로하에서는 프로그래밍 언어들 중, C/C++을 주로 사용할 건데, **Visual Studio**라는 프로그램에서 저희가 하고 싶은 명령을 언어로 적으면 이 언어를 아주 잘 번역해줘요.

Visual Studio 외에도 CodeBlocks나 Dev C++처럼 C++을 번역해주는 다른 프로그램들이 많지만, **VS**가 가장 편하기 때문에 애를 주로 이용할 거예요.

우리는 이것을 VS 홈페이지에서 최신버전으로 편하게 설치할 수 있습니다

우선 말씀드립니다!

C++만을 위한 환경을 만든다면, Visual Studio는 대략 7GB의 공간을 요구합니다 (비워두세용)

또한 소요되는 시간이 매우 길어요. WIFI 환경을 이용합시당.

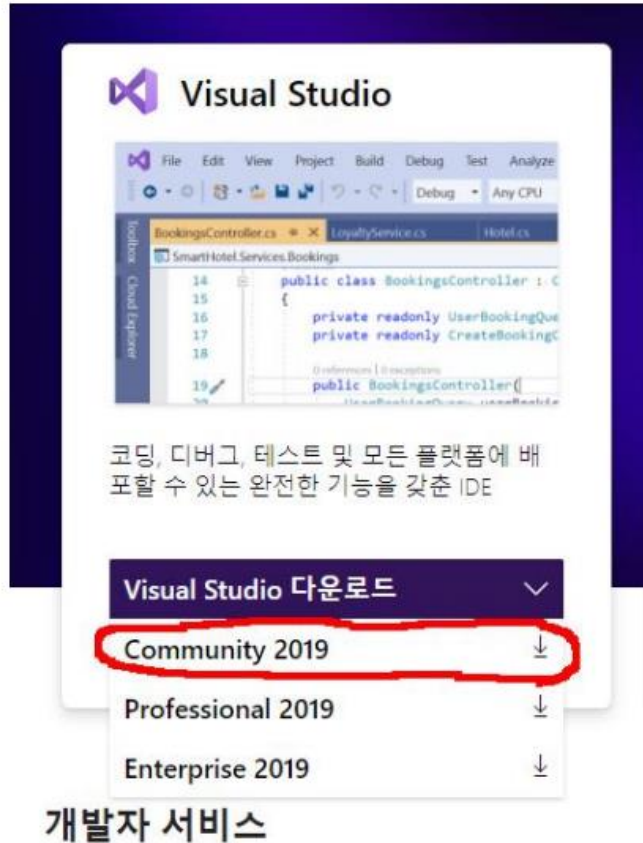
우선 말씀드립니다!

Visual Studio Community 2019(혹은 C++ 기반 컴파일러)는
두 번째 회합 이전까지는 반드시!!! 설치해 주셔야 합니다.

만약 설치 중에 화면이 넘어가지 않거나 **에러가 뜬다면?**
→ 곧바로 **멘토**에게 물어보세요!!

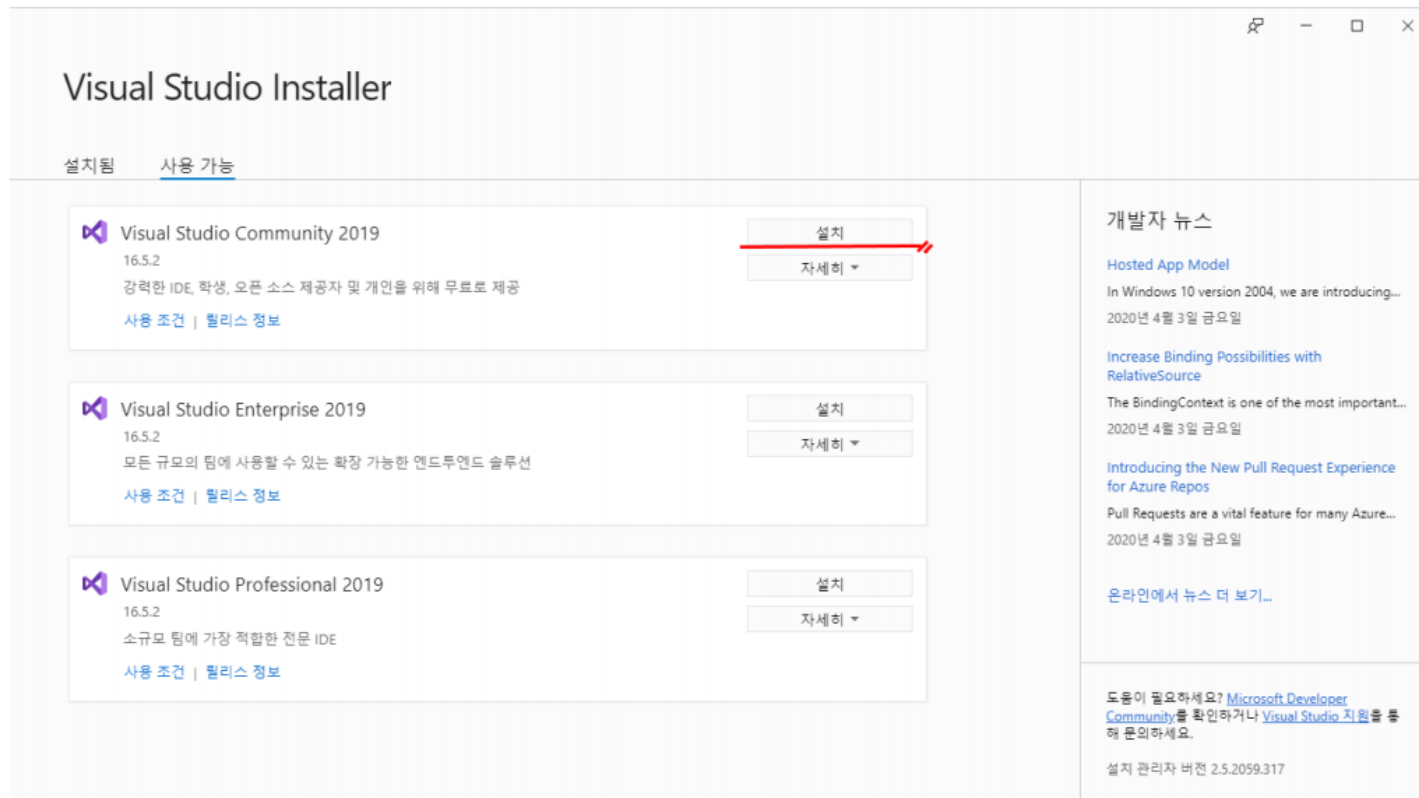
멘토도 모른다면 잡담방에 물어보세요!

Visual Studio를 설치해 봅시다



먼저, <https://visualstudio.microsoft.com/ko/> 으로 들어가신 후 왼쪽 사진에 빨간 동그라미 처리 되어있는 부분을 클릭하여, VS 인스톨러를 받아주세요.

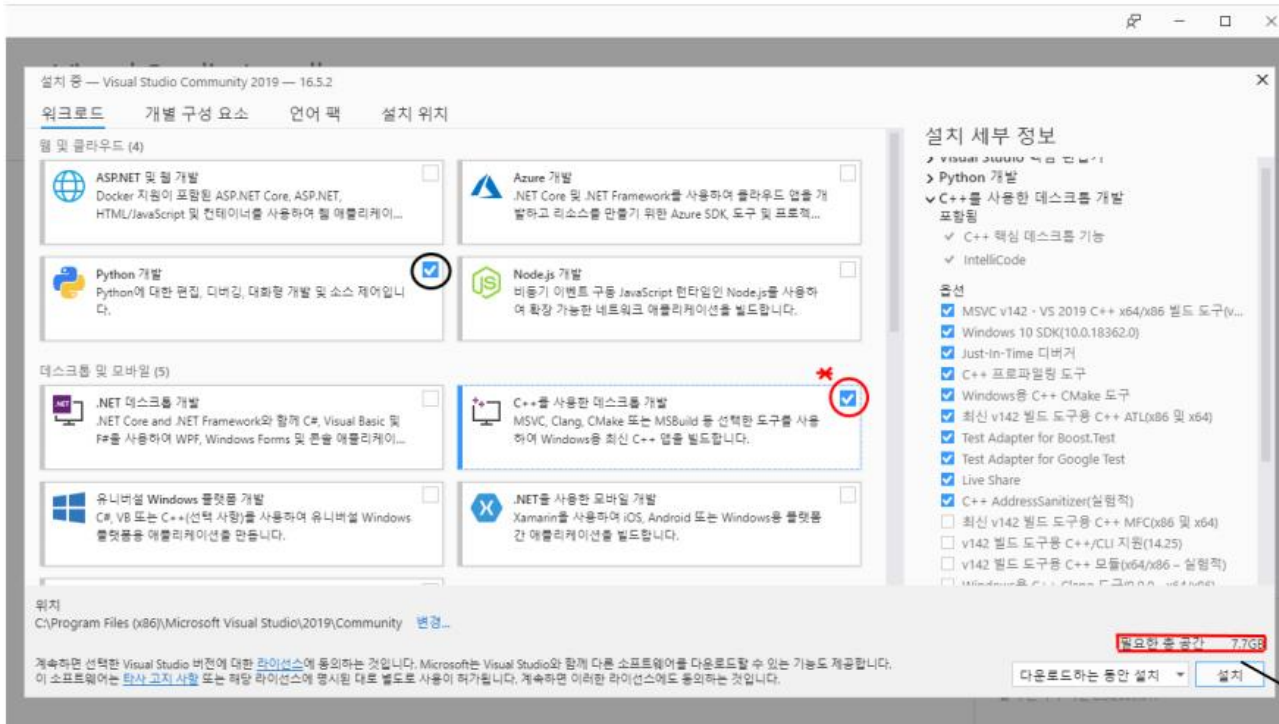
Visual Studio를 설치해 봅시다



그러면 왼쪽과 같은 화면이 뜰 거예요.

여기서, 빨간 부분을 눌러주세요.

Visual Studio를 설치해 봅시다



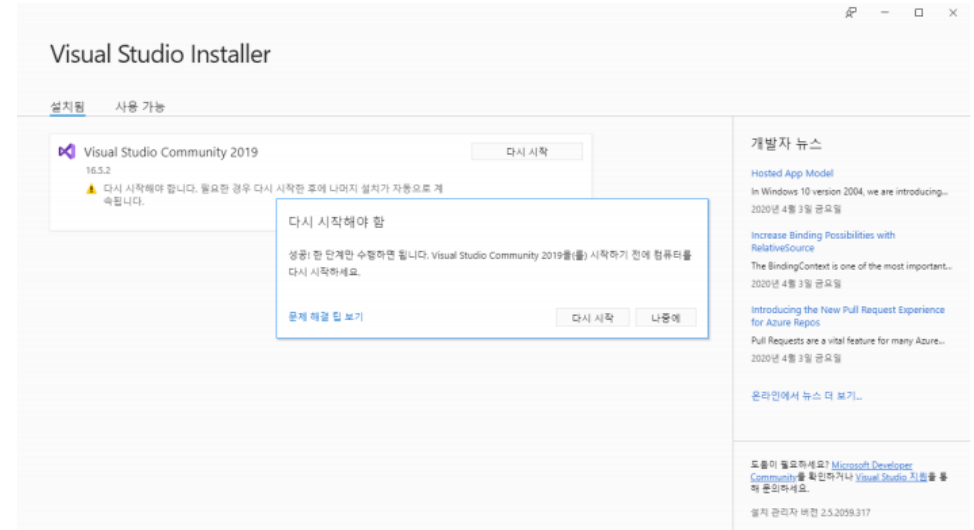
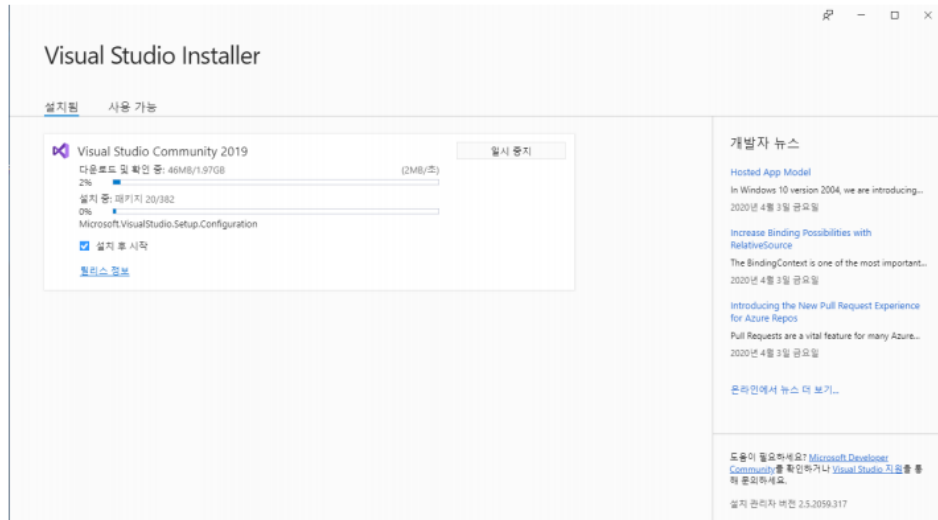
이 부분에서 우리가 사용할 기능을
고를 수가 있는데,

필수적으로 Visual C++은 하셔야 하고,
만약 Python도 Visual Studio에서 쓰고
싶다면 체크하셔서 설치하시면 되겠습니다.

(대신 시간과 용량이 더 필요합니다.)

이거 7.7GB입니다!

Visual Studio를 설치해 봅시다



후루룩 넘기시고
다시 시작 안 하면 진행이 안되므로 재부팅 해주세요

Visual Studio를 설치해 봅시다

Visual Studio

시작합니다.

모든 개발자 서비스에 연결하세요.

Azure 크레딧 사용을 시작하고, 프라이빗 Git 리포지토리에 코드를 게시하고, 설정을 동기화하고, IDE 잠금을 해제하려면 로그인합니다.

[자세히](#)

로그인()

계정이 없는 경우 [새로 만드세요!](#)

[나중에 로그인](#)

설치가 완료되면 왼쪽과 같은 창이 나옵니다.

우선 지금은 **나중에 로그인**을 누르도록 해요.

Visual Studio를 설치해 봅시다

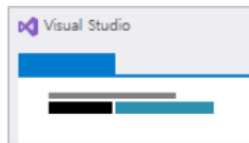
Visual Studio

친숙한 환경에서 시작

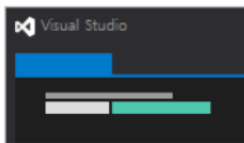
개발 설정(V): Visual C++ *

색 테마 선택

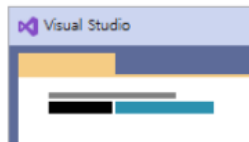
☐ 광원



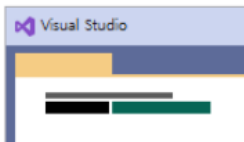
☒ 어둡게



☐ 파랑



☐ 파랑(추가 대비)



나중에 언제든지 이 설정을 변경할 수 있습니다.

Visual Studio 시작(S)

또한 우리는 C/C++ 환경을 원하기 때문에 빨간 네모부분을 반드시 사진과 같게 해주세요.

아래 배경 색상은 개인 취향대로 골라 넘어가 주시면 됩니다. 저는 코드가 눈에 잘 보이도록 어두운 배경을 골랐습니다.

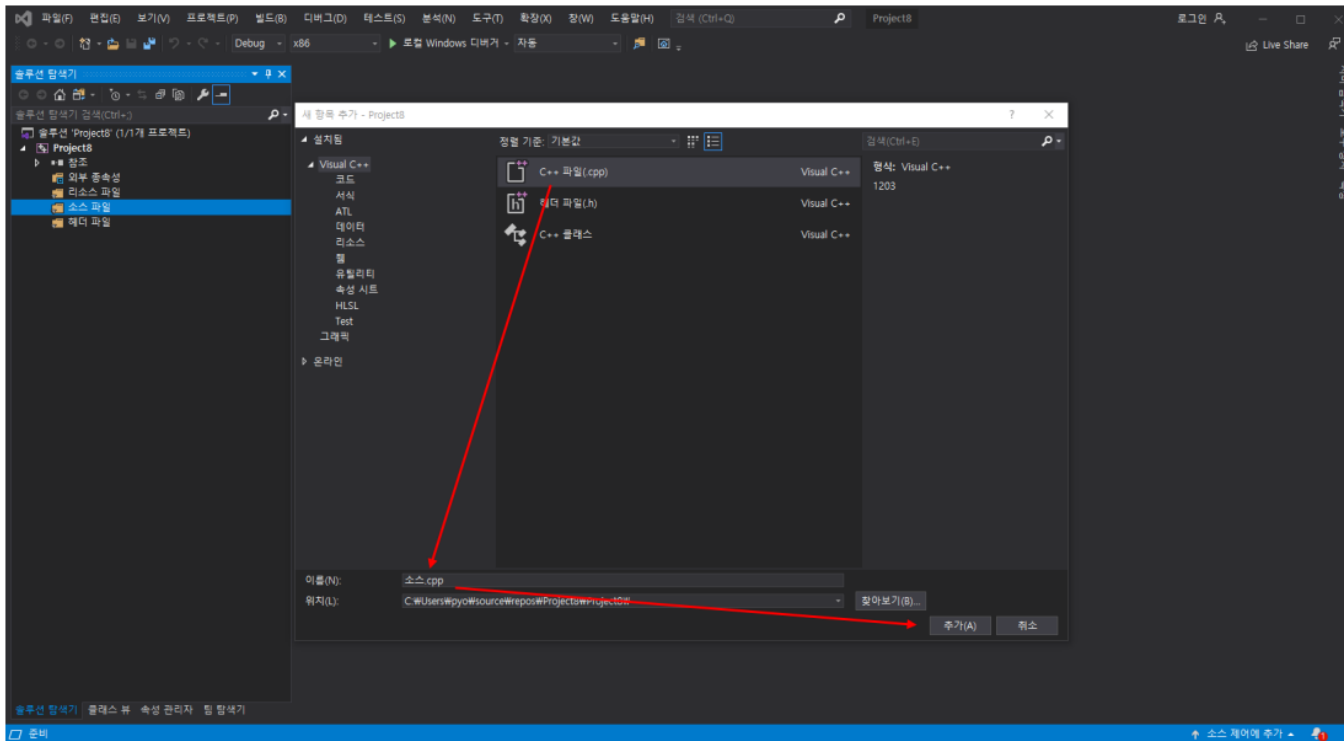
Visual Studio를 설치해 봅시다



자! 이제 프로젝트 생성입니다.

새 프로젝트 만들기를 누르시고,
데스크톱 마법사로 열어주세요.

Visual Studio를 설치해 봅시다



이제 소스파일을 생성해봅시다.

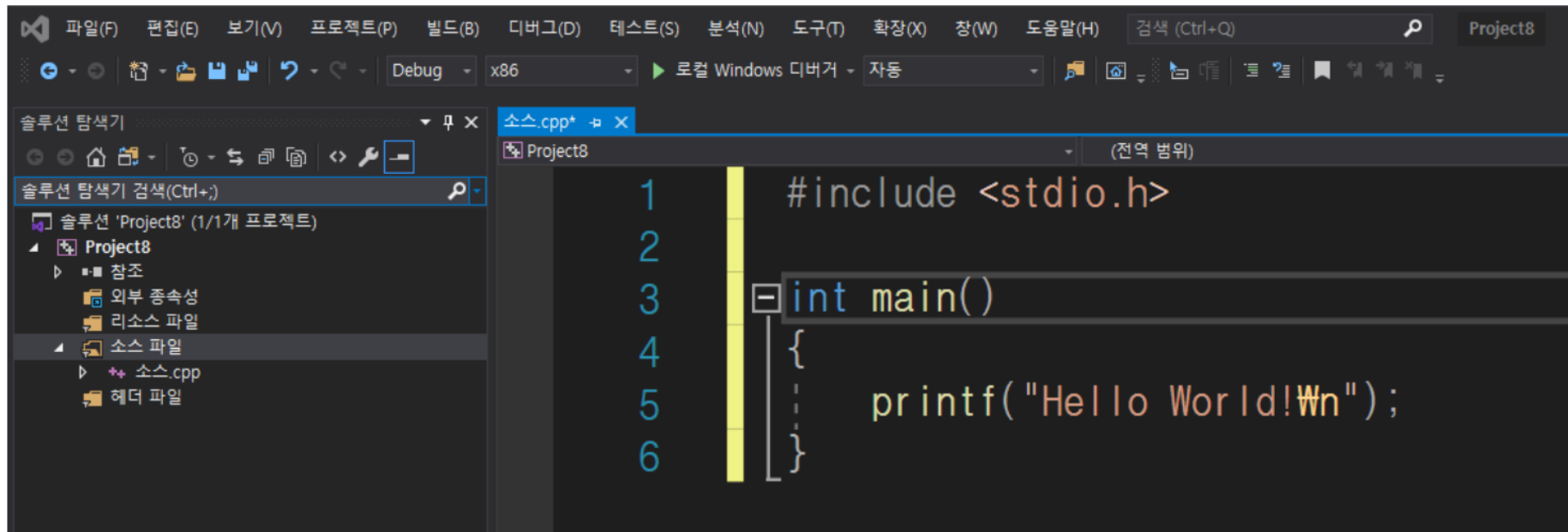
메인에서 Ctrl + Shift + A

혹은

소스파일(우클릭) – 추가 – 새 항목

으로 생성할 수 있습니다.

Visual Studio를 설치해 봅시다



그러면 소스창이 열리고, 이제 배우기 위한 모든 준비는 끝입니다!

#CH.2

C 문법을 통한 기초 I/O



C 문법을 통한 기초 I/O

이번 시간이 첫 회합인 만큼, 오늘은 아주 가벼운 얘기를 해볼까 해요!

바로 **I/O**, 입력 및 출력에 대한 내용입니다. (Input/Output)

C 문법을 통한 기초 I/O

```
1  #include <stdio.h>
2  // 헤더파일
3
4  int main()
5  {
6      int x;
7      // 입력을 받을 변수를, 정수형으로 정해서 선언했다.
8
9      scanf("%d", &x);
10     // 입력이다!
11     printf("%d", x);
12     // 출력이다!
13
14     return 0;
15 }
```

이 수업을 듣는 여러분 시점에서는
소입설 시간에 **Python**을 배워서 입출력을 하고,
다양한 것들을 표현해 보았을 거예요.
자료형에 대한 내용도 어느 정도 알고는 있겠죠.

물론 C를 이전에 접해봤다면 더욱 쉽겠지만!
Python의 내용만 알고 있어도 괜찮습니다.

지금부터 x라는 **정수형 변수**에, 값을 **받고**,
출력을 해보고자 합니다.

C 문법을 통한 기초 I/O

```
1  #include <stdio.h>
2  // 헤더파일
3
4  int main()
5  {
6      int x;
7      // 입력을 받을 변수를, 정수형으로 정해서 선언했다.
8
9      scanf("%d", &x);
10     // 입력이다!
11     printf("%d", x);
12     // 출력이다!
13
14     return 0;
15 }
```

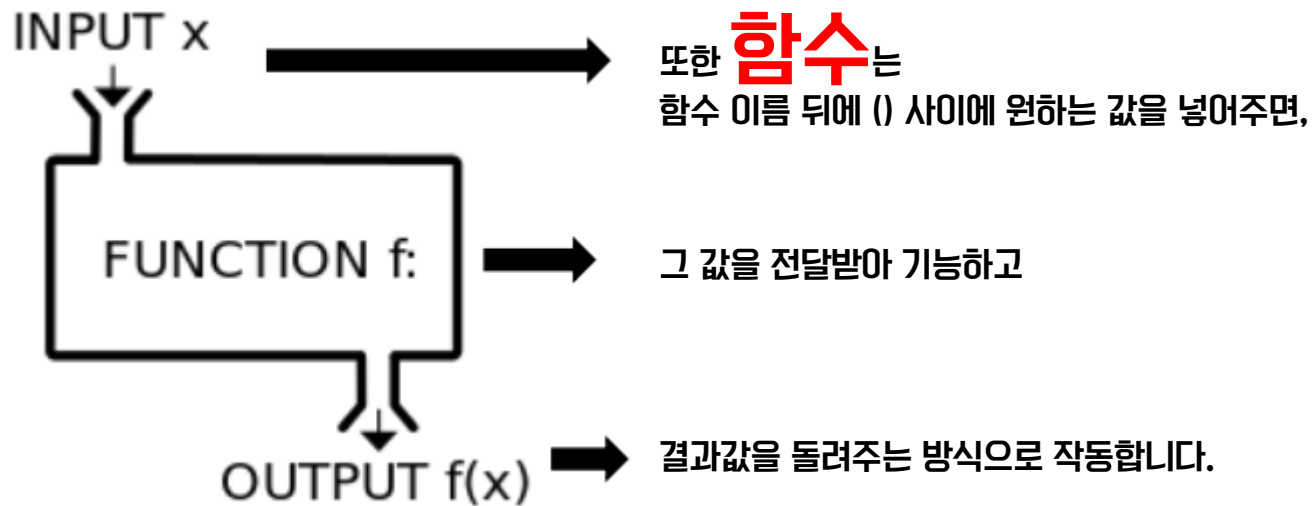
우선 첫 줄 `#include <stdio.h>`

C언어를 쓴다고 하면, 대부분의 명령이
어떠한 내용을 담는 함수를 통해 표현될 거예요.

그 함수는 **헤더파일**이라는 곳에 모여져 있는데,
우리는 항상 시작할 때 저렇게 적어주어야 합니다!

(헤더파일도 다양한 종류가 있습니다!)

C 문법을 통한 기초 I/O



그런데, 대부분의 함수는 미리 정의되어 있어요!

따라서 우리는 그 함수가
1. 어떤 기능을 하는지 알고
2. 쓰는 법만 알면 되는 것이죠.

(우리가 함수를 직접 만들 수도 있어요)

C 문법을 통한 기초 I/O

```
1  #include <stdio.h>
2  // 헤더파일
3
4  int main()
5  {
6      int x;
7      // 입력을 받을 변수를, 정수형으로 정해서 선언했다.
8
9      scanf("%d", &x);
10     // 입력이다!
11     printf("%d", x);
12     // 출력이다!
13
14     return 0;
15 }
```

int main()

프로그램의 모든 동작은 **main 함수를 실행함**으로써 시작합니다. 따라서, main함수 안에 원하는 명령을 넣으면 돼요.

그리고, Python과 조금 다르게

반드시 명령 하나를 적으면 ; 을 반드시 적어주어야 합니다.
이게 “하나의 명령이 끝났다.”라는 표시이기 때문이에요.

변수를 선언할 때는,

TYPENAME_ VAR_NAME = VAL; 순으로 적어요.

여기서는 정수형(**int**) 변수인 **x**를 초기화하지 않고 선언했네요.
(값을 처음에 넣지 않았다는 의미예요.)

C 문법을 통한 기초 I/O

```
1  #include <stdio.h>
2  // 헤더파일
3
4  int main()
5  {
6      int x;
7      // 입력을 받을 변수를, 정수형으로 정해서 선언했다.
8
9      scanf("%d", &x);
10     // 입력이다!
11     printf("%d", x);
12     // 출력이다!
13
14     return 0;
15 }
```

scanf("%d", &x); 부분입니다.
이 함수는 우리가 stdio 헤더에서 불러온 **표준 입력 함수**예요!

“와 ” 사이에 %를 이용하여 **값을 받는 형식**을 정해주고,
뒤에 순서대로 값이 어디에 들어가야 하는지를 열거해주면 되는데,
숫자 대입과 느낌이 매우 비슷합니다.

대신, 뒤에 **열거하는 부분**에는 변수명이 아닌,
&를 변수명과 붙여서 적어주어야 합니다!

& 연산자에 대한 자세한 내용은 포인터 시간에 배우도록 합시다!

C 문법을 통한 기초 I/O


```
1  #include <stdio.h>
2  // 헤더파일
3
4  int main()
5  {
6      int x;
7      // 입력을 받을 변수를, 정수형으로 정해서 선언했다.
8
9      scanf("%d", &x);
10     // 입력이다!
11     printf("%d", x);
12     // 출력이다!
13
14     return 0;
15 }
```

이때! 형식 지정자는

%d	정수형 (int)
%lld	정수형 (long long int)
%lf	실수형 (double)
%s	문자열 (문자열은 문자의 배열이에요. 나중에 다뤄봅시다!)
%c	문자형 (char)

형식이든 입력이든 출력이든 항상 꼭 맞춰주어야 합니다!

C 문법을 통한 기초 I/O



```
scanf("%d%s%f", &x, s, &y);
```

또한 여러 개를 입력 받을 수도 있습니다.

입력을 구분하는 기준은 **white space**(공백)입니다.

이때 **white space**는 스페이스바(' '), 엔터 키('\n'), 탭 키('\t') 등을 모두 포함합니다.

※참고로 문자열은 &를 붙이지 않습니다! (배열이거든요.)

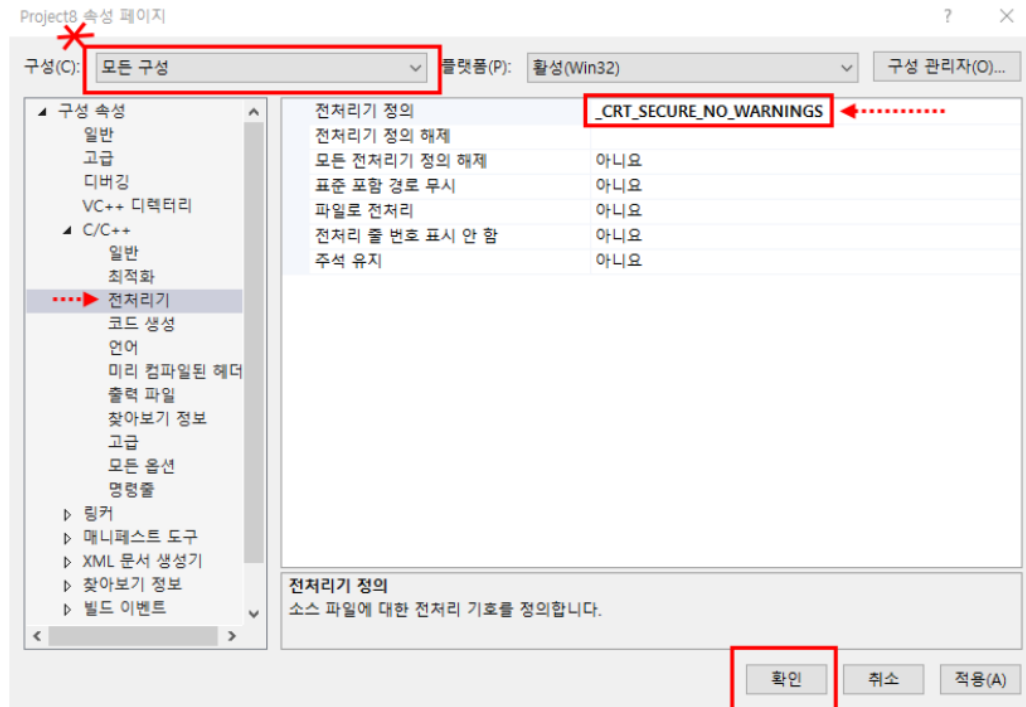
C 문법을 통한 기초 I/O



Visual Studio에서는 scanf로 적고 실행하면 오류가 날 거예요!
이유는 안전성 때문인데요, 우리는 이에 크게 신경 쓸 것 없이
설정 하나만 하고 넘어갑시다!

VS 상단 메뉴 부분에서 [프로젝트]-[속성] 들어가고,
나오는 창에서 왼쪽 부분에, C/C++ - 전처리기 부분으로 들어가주세요.

C 문법을 통한 기초 I/O



그리고, 왼쪽 사진처럼 모든 구성으로 한 뒤
전처리 정의에

`_CRT_SECURE_NO_WARNINGS`

를 추가해 주시고 확인 눌러주세요!

C 문법을 통한 기초 I/O

```
1  #include <stdio.h>
2  // 헤더파일
3
4  int main()
5  {
6      int x;
7      // 입력을 받을 변수를, 정수형으로 정해서 선언했다.
8
9      scanf("%d", &x);
10     // 입력이다!
11     printf("%d", x);
12     // 출력이다!
13
14     return 0;
15 }
```

`printf("%d", x);`
이 함수는 **표준 출력 함수**입니다.

Python의 `print` 함수와 매우 유사해요. 대신 개행을 자동으로 하지는 않습니다. 왼쪽을 보시면 형식지정자를 통해 `x`라는 변수의 내용을 출력해주고 있습니다.

아까 `scanf` 함수와는 다르게 **`printf` 함수는 그냥 변수명으로 나열** 합니다.

C 문법을 통한 기초 I/O

```
1  #include <stdio.h>
2  // 헤더파일
3
4  int main()
5  {
6      int x = 6;
7      double d = 3.14;
8
9      printf("%d %f", x, d);
10     // 출력이다!
11
12     return 0;
13 }
```

Microsoft Visual Studio 디버그 콘솔

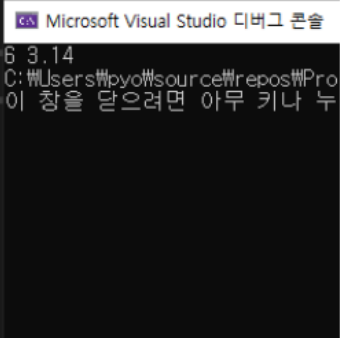
```
6 3.140000
C:\Users\pyo\source\repos\Pro
이 창을 닫으려면 아무 키나 누
```

또한 printf 함수는 **출력 형식을 여러분이 직접 지정**할 수 있어요

기본적으로 실수형 변수는 %f를 통해서 출력하면,
소수점 아래 7번째 자리에서 반올림합니다.
대신 약간의 처리를 통해 범위를 조정할 수도 있죠.

C 문법을 통한 기초 I/O

```
1  #include <stdio.h>
2  // 헤더파일
3
4  int main()
5  {
6      int x = 6;
7      double d = 3.14;
8
9      printf("%d %.2f", x, d);
10     // 출력이다!
11
12     return 0;
13 }
```



The screenshot shows the Microsoft Visual Studio debugger console. The title bar reads "Microsoft Visual Studio 디버그 콘솔". The output text is "6 3.14". Below the output, there is a file path "C:\Users\wpyo\source\repos\Pro" and a line of text "이 창을 닫으려면 아무 키나 누" (Press any key to close this window).

이렇게요.

`%.2f`의 `.2`는
소수점 아래 2자리까지만 출력하도록 지정할 수 있게 해줍니다.

C 문법을 통한 기초 I/O

```
1  #include <stdio.h>
2  // 헤더임동
3
4
5  int main()
6  {
7      int x;
8      // 변수라예
9
10     printf("Hi, everyone!\nMy name is PYO");
11     // 출력문이라능
12
13     return 0;
14 }
```

Microsoft Visual Stu
Hi, everyone!
My name is PYO
C:\Users\pyo\source
이 창을 닫으려면

하나 더!

우리는 **다음줄로 넘어가는 내용**을 적기 위해
개행(‘\n’) 문자를 사용할 수 있습니다!

다음줄로 넘겨야만 하는 부분이 있다면 꼭 써야겠죠?

C 문법을 통한 기초 I/O

```
1  #include <stdio.h>
2  // 헤더파일
3
4  int main()
5  {
6      int x;
7      // 입력을 받을 변수를, 정수형으로 정해서 선언했다.
8
9      scanf("%d", &x);
10     // 입력이다!
11     printf("%d", x);
12     // 출력이다!
13
14     return 0;
15 }
```

전체 코드를 한번 실행해 볼까요?

Visual Studio에서는

저장: Ctrl + S

되돌리기: Ctrl + Z

디버깅 없이 실행: Ctrl + F5

한줄씩 디버깅: F11

입니다!

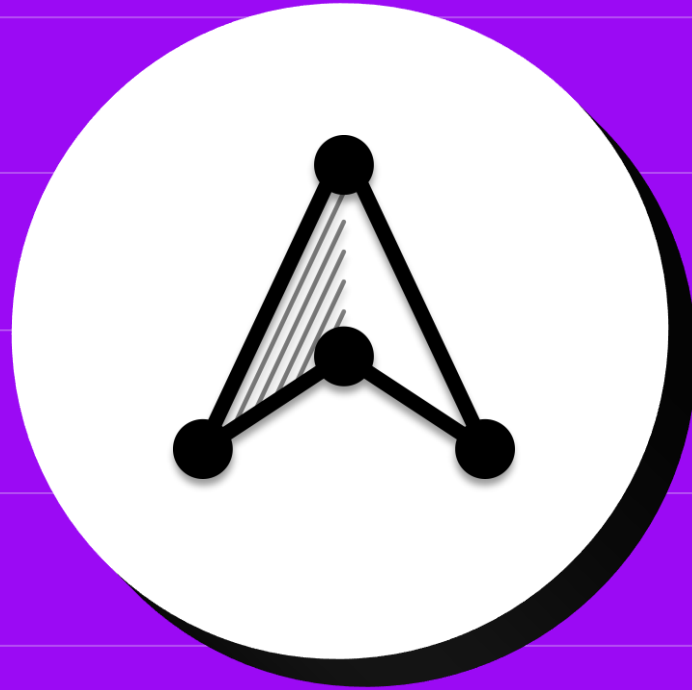
먼저 저장을 해주고, Ctrl + F5로 결과를 보도록 해요. 잘 되죠?

C 문법을 통한 기초 I/O

이렇게 **I/O**는 기본적인 것들만 다 다루어 보았어요.

물론 이게 전부는 아니지만 변수, 형식 및 함수와 연산자들을 더더욱 알아가다 보면,
다양한 입출력 방식이 존재한다는 것을 알게 돼요!

더 재밌는 내용이 이제 **다음 주차**에 나올 테니, 기대해보도록 합시다.



다음 시간에 만나요~