

프로그래밍 역량 강화 전문기관, 민코딩

---

# 프로그래밍 첫 걸음

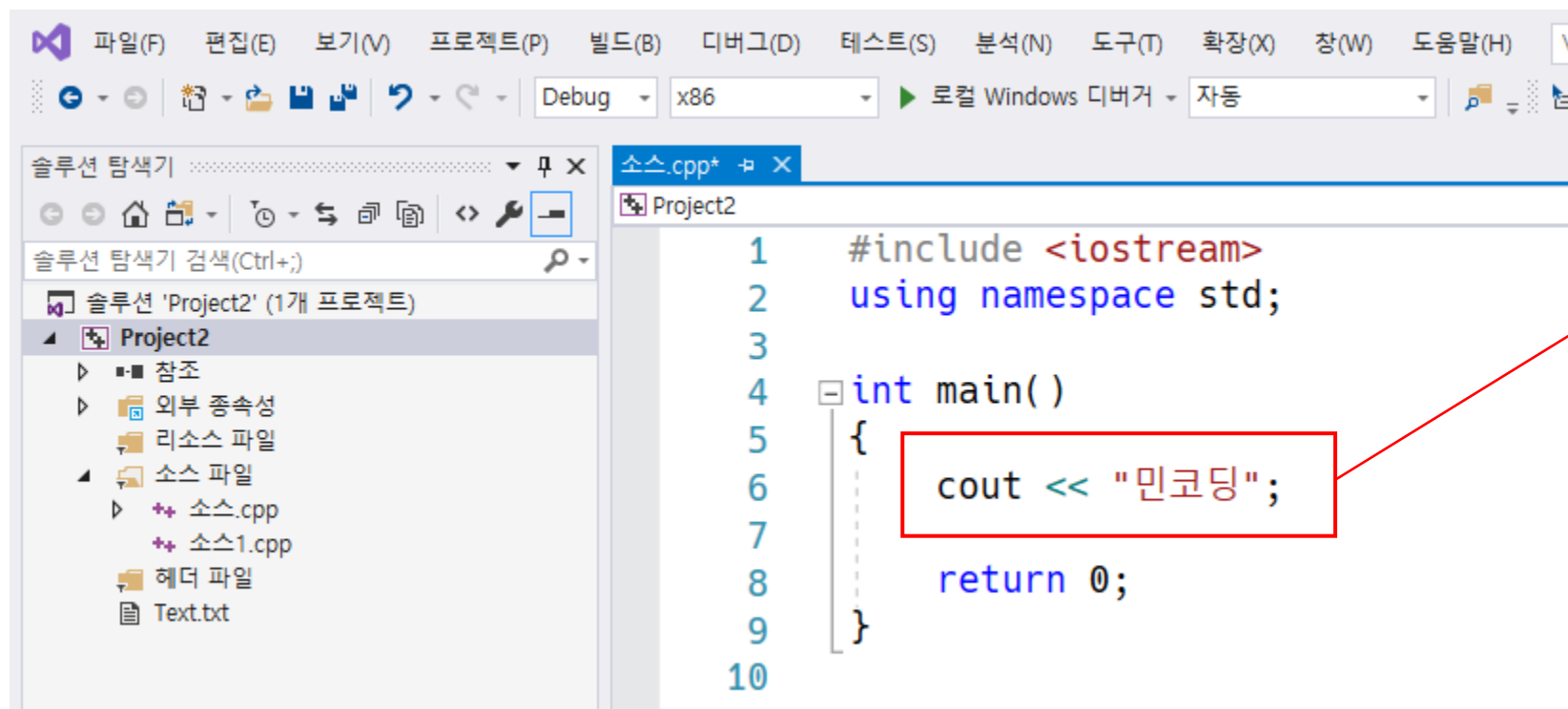


# 배우는 내용

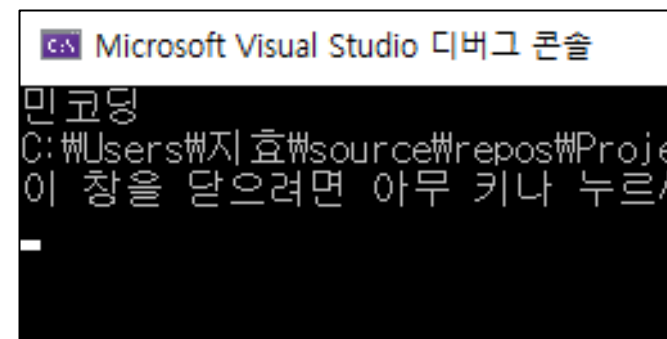
## 프로그래밍 첫 걸음

1. 콘솔 출력
2. 변수

# 글자 출력하는 소스코드



민코딩이라는 글씨를 출력함  
Ctrl + F5를 눌러서 빌드 (결과확인 가능)



# 빌드 결과 확인 (Ctrl + F5)

\*cout : 씨아웃 이라고 발음 (Console Output의 약자)

The screenshot displays the Visual Studio IDE with a C++ project named 'Project2'. The source file '소스.cpp' contains the following code:

```
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     cout << "하하하";
6
7     return 0;
8 }
9
10
11
```

The 'Output' window at the bottom left shows the build process:

```
출력
출력 보기 선택(S): 빌드
1>----- 빌드 시작: 프로젝트: Project2, 구성: Debug Win32 -----
1>소스.cpp
1>Project2.vcxproj -> C:\Users\지효\source\repos\Project2\Debug\Project2.exe
----- 빌드: 성공 1, 실패 0, 최신 0, 생략 0 -----
|
```

A red box highlights the output window with the text: **빌드에 성공했음을 나타내는 메세지** (Message indicating successful build).

The 'Debug Console' window on the right shows the execution output:

```
Microsoft Visual Studio 디버그 콘솔
하하하
C:\Users\지효\source\repos\Project2\Debug\Project2.exe(14516 프로세스)이(가) 0 코드로 인해 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.
```

A red box highlights the debug console with the text: **결과화면을 콘솔창이라고 한다.** (The result screen is called the console window).

# 기본 소스코드의 해석

```
#include <iostream>

int main()
{
    std::cout << "민코딩";

    return 0;
}
```

**#include <iostream>**은  
iostream.h 파일을 불러오는 코드  
std::cout 명령어를 사용할 수 있다.

```
#include <iostream>

int main()
{
    std::cout << "민코딩";

    return 0;
}
```

**std::cout**은 민코딩 글자를 출력한다.

```
#include <iostream>
using namespace std;


int main()
{
    cout << "민코딩";

    return 0;
}
```

std::cout에서  
std:: 글자를 생략하기 위해  
**using namespace std;**를 적어준다.

\* 차후에 #include와 using namespace에 대한 내용을 자세히 알 수 있다.

# 코딩 공부를 위한 기본 소스코드

```
#include <iostream>
using namespace std;
int main()
{
    
    return 0;
}
```

{ } 괄호 사이에 소스코드를 넣으면 된다.

공백이 많이 들어가도 상관없음

프로그래밍 공부의 가장 기본이 되는 소스코드  
이 형틀을 잊어버리지 않도록 반복 연습을 통해 기억하자.

# cout으로 글자 출력하기

두 소스코드는 같은 결과가 나온다.  
마음에 드는 방식으로 코딩한다.

```
#include <iostream>
using namespace std;
int main()
{
    cout << "민코딩";
    cout << "개발자";
    cout << "BBQ";

    return 0;
}
```

```
#include <iostream>
using namespace std;
int main()
{
    cout << "민코딩개발자BBQ";

    return 0;
}
```

빌드

Microsoft Visual Studio 디버그 콘솔

```
민코딩개발자BBQ
C:\Users\최지효\source\repos\Project2\Det
이 창을 닫으려면 아무 키나 누르세요.
```

# 줄바꿈 출력

다음과 같이 출력하고 싶다면?

민코딩  
개발자  
BBQ



```
#include <iostream>
using namespace std;
int main()
{
    cout << "민코딩" << endl;
    cout << "개발자" << endl;
    cout << "BBQ" << endl;

    return 0;
}
```

줄바꿈을 뜻함  
(End Line의 약자)

Microsoft Visual Studio 디버그 콘솔

```
민코딩
개발자
BBQ

C:\Users\zzzz\source\repos\Project
이 창을 닫으려면 아무 키나 누르세:
```



# 변수란?

- ✓ 컴퓨터 안에 **값을 저장하는 박스** - 숫자(정수)나 문자, 소수점을 저장할 수 있다.
- ✓ 박스를 만들 때는 고유 이름을 지어줘야 한다.
- ✓ 변수(박스)의 종류 예시



숫자(정수)가 들어  
가는 int 타입 박스

=인트형 변수



글자가 들어가는  
char 타입 변수

=캐릭터형 변수

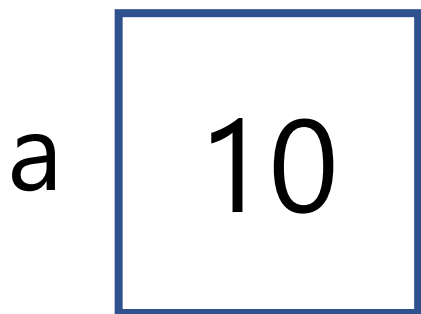


소수점 숫자가 들어가  
는 double 타입 박스

=더블형 변수

# 변수 만드는 법

아래 예시는 a라는 박스를 만든 후, 박스 안에 값 10을 넣었다.  
둘 중 마음에 드는 방식으로 하면 된다.



```
#include <iostream>
using namespace std;
int main()
{
    int a;
    a = 10;

    return 0;
}
```

```
#include <iostream>
using namespace std;
int main()
{
    int a = 10;

    return 0;
}
```

위 소스코드를 작성 후, 빌드하면 콘솔창에 아무것도 뜨지 나오지 않는다.  
박스를 만들었을 뿐, cout명령어로 글자를 출력하지 않았기 때문이다.

# 개발자가 사용하는 용어 (암기)

a라는 이름의 변수(박스)를 만든다.

→ 변수 a를 선언한다.

변수(박스) a에 값 10을 넣는다.

→ 변수 a를 10으로 초기화한다.

# 변수 선언 및 초기화 예제

변수 a, b, c를 선언하고 각각 1, 2, 3으로 초기화하기

```
#include <iostream>
using namespace std;
int main()
{
    int a;
    int b;
    int c;

    a = 1;
    b = 2;
    c = 3;

    return 0;
}
```

=

```
#include <iostream>
using namespace std;
int main()
{
    int a;
    a = 1;

    int b;
    b = 2;

    int c;
    c = 3;

    return 0;
}
```

# 버그 발생 코드 1

버그란? : 소스코드 안의 **벌레**, 프로그래머 실수로 인한 오류를 뜻함

→ 변수 선언을 먼저 하지 않고, 안에 값을 넣어야 하기 때문에 버그 발생

```
#include <iostream>
using namespace std;
int main()
{
    int a;
    int b;
    b = 2;
    c = 3;
    int c;

    return 0;
}
```

# 버그 발생 코드 2

오른쪽 소스코드는 왜 버그가 발생할까?

→ 같은 이름의 변수를 2개 만들었기 때문이다.

```
#include <iostream>
using namespace std;
int main()
{
    int a = 40;
    int b = 20;
    int c = -5;

    a = 15;
    b = 22;
    int c = 15;

    return 0;
}
```

# 변수 만드는 법

```
int a;           // a라는 변수(박스)를 만들었다.  
int bbq123;      // bbq123이라는 변수를 만들었다.  
int a, b, c;      // a, b, c라는 변수를 만들었다.  
int d1 = 5, d2 = 10; //d1과 d2 변수를 만들자마자 값을 5와 10을 넣었다.
```

변수 선언하면 초기값으로 쓰레기 값이 저장되어 있다.

(**쓰레기 값** : 의도하지 않은 값. 아무 값도 넣지 않았을 때 자동으로 넣어지는 값)

# 변수의 활용 1

```
int a = 10;  
a = 100;  
// 변수 a에 10을 넣고, 다시 a값을 100으로 바꾸었다.  
// a 안에 있던 10은 없어지고, 값 100이 들어가 있다.
```

```
int g = 1, k = 2;  
int c = g + k;  
// c라는 변수에는 숫자 3이 들어간다
```



# 변수의 활용 2

```
int g;  
g = 10 / 3;  
// int 변수는 소수점을 저장하는 박스가 아니라, 정수를 저장하는 박스이다.  
// 변수 g에는 소수점 뒷자리를 버리고, 숫자 3이 들어간다.
```

```
int a = 3 * 10;  
// a에는 숫자 30이 들어간다.
```

# 변수 용어 정리

`int a;` // 변수 `a`를 선언한다. (쓰레기 값이 들어있음)

`int a = 10;` // 변수 `a`를 선언 후 10으로 초기화한다.

`int a = 1, b = 10;` // 변수 `a, b` 선언 후 각각 1과 10으로 초기화 한다.

# 변수 값 출력하기

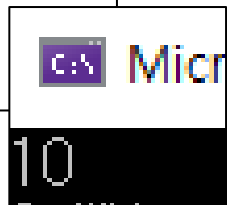
cout을 이용하여 변수의 값을 화면에 출력할 수 있다.

변수 값을 출력하는 것과 글자를 출력하는 것을 헷갈리지 않도록 주의하자.

```
#include <iostream>
using namespace std;
int main()
{
    int a = 10;
    int b = 20;
    int c = a + b;

    cout << a;

    return 0;
}
```

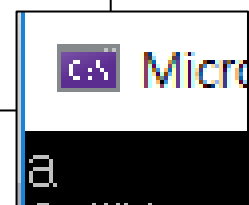


10

```
#include <iostream>
using namespace std;
int main()
{
    int a = 10;
    int b = 20;
    int c = a + b;

    cout << "a";

    return 0;
}
```



a

# 글자와 변수를 함께 출력하기 1

<< 기호를 여러 번 사용하여, 글자와 변수를 동시에 출력이 가능하다.

```
#include <iostream>
using namespace std;
int main()
{
    int a = 10;
    int b = 20;
    int c = a + b;

    cout << "a의 값은 : " << a << "입니다" << endl;

    return 0;
}
```

Microsoft Visual Studio 디버그 콘솔

a의 값은 : 10입니다

# 글자와 변수를 함께 출력하기 2

<< 기호를 여러 번 사용하여, 글자와 변수를 동시에 출력이 가능하다.

```
#include <iostream>
using namespace std;
int main()
{
    int a = 10;
    int b = 20;
    int c = a + b;

    cout << a << "와 " << b << "를 더하면";
    cout << c << "입니다";

    return 0;
}
```

Microsoft Visual Studio 디버그 콘솔

10와 20를 더하면30입니다

# 자주 발생하는 버그 (세미콜론 생략)

소스코드 끝에 세미콜론(;)을 붙이지 않는 경우, 코드 주변에 빨간 밑줄이 그어집니다.

```
#include <iostream>
using namespace std;
int main()
{
    int a = 10;
    int b = 20;
    int c = a + b

    return 0;
}
```

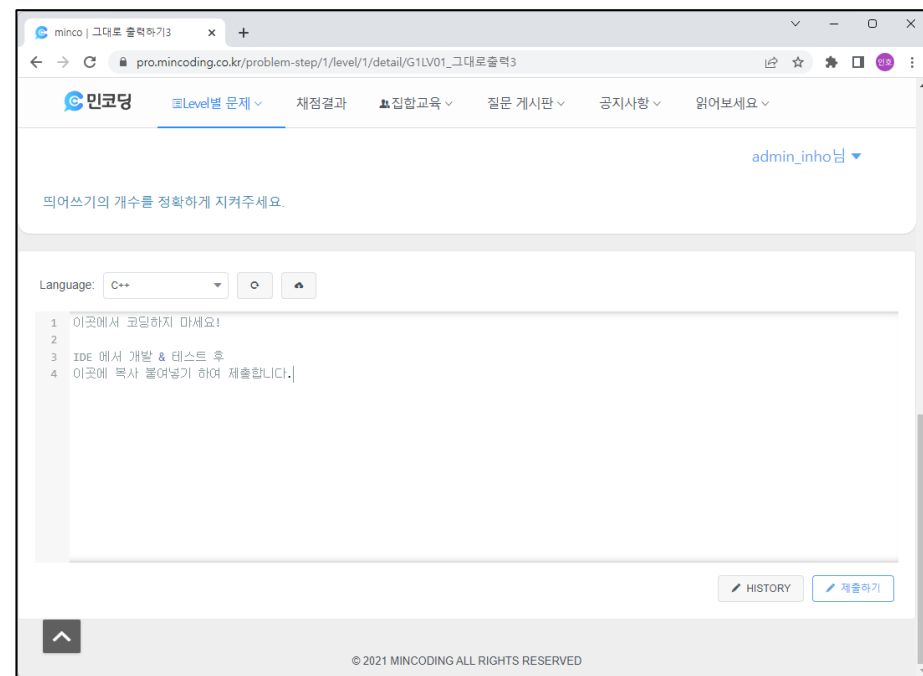
버그, 세미콜론을 붙이지 않음

버그 문장 주변에 빨간 밑줄이 그어짐  
(이곳은 버그가 아님)

# 숙제 할 때 주의할 점 1

✓ Visual Studio에서 코딩한 후, 훈련사이트로 채점하기. (훈련사이트에서 코딩하지 말 것)

```
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     int a = 10;
6     int b = 20;
7     int c = a + b;
8
9     cout << a;
10
11     return 0;
12 }
13
14
15
```



코딩은 Visual Studio에서 하고, 빌드 까지 해서 오류가 없는지 확인해야 한다.  
완료가 되었으면, 소스코드를 복사 붙여넣기를 해서 제출하시면 된다.

# 숙제 할 때 주의할 점 2

✓ 문제를 풀기 시작할 때, **기존 소스코드를 지우고**, 기본 코드부터 새로 작성할 것.

```
#include <iostream>
using namespace std;

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

항상 처음부터 소스코드를 작성한다는 생각으로  
코딩을 해야 실력이 빠르게 늘 수 있다.

소스코드 의존성 제거를 위한 학습 훈련

**(번거롭지만 실력 향상을 위해 꼭 지킬 것)**