

프로그래밍 역량 강화 전문기관, 민코딩

프로그래밍에서의 조건문



배우는 내용

프로그래밍에서의 조건문

1. 콘솔 입력
2. if와 if-else

시작 전 숙지해야 할 사항들

아래와 같은 기본코드에서 코딩을 시작할 것.

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

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

코딩 훈련을 할 때 소스코드는 항상 기본코드에서 시작할 것.

→ 이전에 코딩 했던 소스코드를 **하나씩 고치면서 코딩하지 말 것.**

항상 새롭게 짜는 습관을 들이도록 하자.
코딩실력이 **더 빨리 향상 된다.**

콘솔 입력 : cin

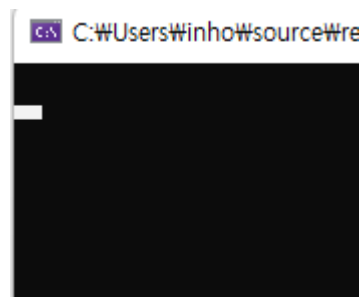
```
cin >> a;
```

```
// 수 하나를 키보드로 입력 받고,  
// 그 값을 변수 a에다가 넣는다.
```

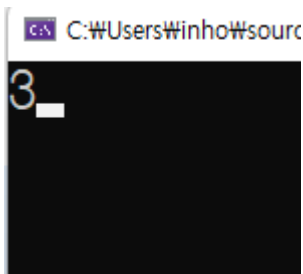
cin : Console Input을 뜻함

수(Number) / 문자 / 소수점을 키보드로 입력 받고,
입력 받은 값을 변수(박스) 안에 넣는 명령어이다.

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



커서가 깜박 거린다.



키보드로 3 입력 후
엔터를 누른다.

변수 a에
입력한 3 이
넣어진다.

콘솔 입력의 활용

수 2개를 입력 받는 방법

```
int a, b;
```

```
cin >> a >> b;
```

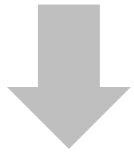
```
cout << a << " " << b;
```

숫자 “**3 5**” 를 입력하면
변수 a에는 3이 들어가고,
변수 b에는 5가 들어간다.

cin과 cout의 순서

✓ 문제 : 수를 입력 받고, 그 수를 출력하세요.

입력하세요 : _



3 입력하고 Enter

입력하세요 : 3

3을 눌렀음

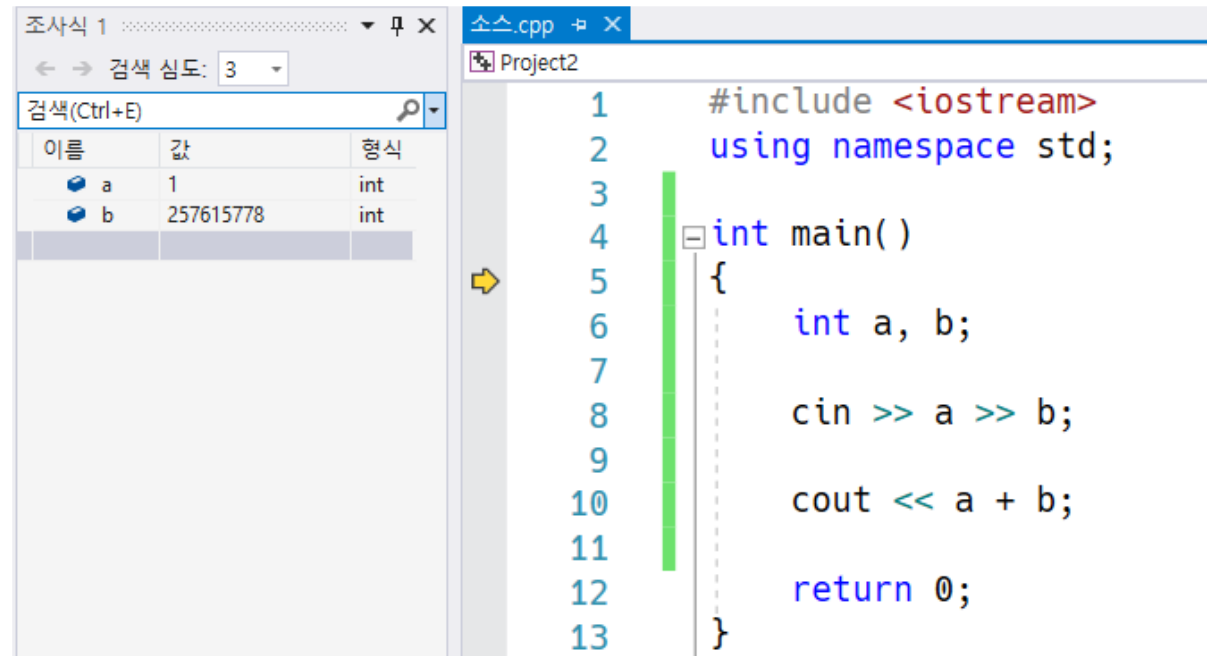
```
int a;  
cout << "입력하세요 : ";  
cin >> a;  
cout << a << "을 눌렀음";
```

cin / cout 순서 주의

“입력하세요” 글씨를 먼저 출력하고, 입력을 받아야 한다.
그런 후, 다음 내용을 출력 해야 한다.

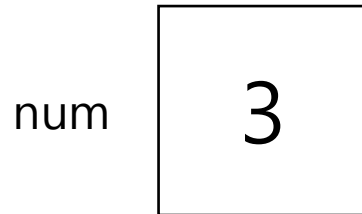
필수 : 소스코드 Trace 해보기

- ✓ 숫자 2개를 입력 받고 합을 출력하는 프로그램을 작성 후, F10 을 누른다.
- ✓ 조사식에 변수 a, b를 등록시키고 F10을 반복해서 눌러본다.



변수에 있는 값을 더하고 빼기

변수 num 안에 3이 들어있다.
아래 코드는 이 변수 값에 5를 더하거나 빼는 두가지 방법이다.



```
num = num + 5;
```

줄임표현

→ `num += 5;`

```
num = num - 5;
```

줄임표현

→ `num -= 5;`

덧셈 뺄셈 뿐만 아니라 곱셈 나눗셈도 줄임 표현이 가능하다. ex) `x *= 5;` `x /= 2;`

$x++$ / $x--$

변수 1을 더하는 것과 1을 빼는 코드는 자주 쓰인다.
따라서 다음과 같은 간결한 문법이 있다.

- ▶ `x++;` // 값 1 더하는 방법
- ▶ `x--;` // 값 1 빼는 방법

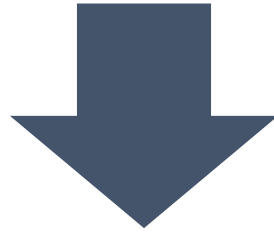
* 값 1 증가와 값 1 감소가 가능하지만, 곱셈 및 나눗셈에 대한 줄임 표현은 없다.

x++; 사용할 때 유의할 점

```
int x = 50;
```

```
int a = x++;
```

우선순위 규칙에 의해
a에는 값 50이 대입된다.
그 후, x는 51이 된다.



```
int x = 50;
```

```
a = x;
```

```
x++;
```

한 줄당 한 가지 명령어만 사용하면
우선순위가 혼동 될 일이 없다.

주석(메모) 사용하기

- ✓ 주석은 프로그래밍 소스코드에 메모를 남기는 기능이다.
- ✓ 한 줄 주석은 `//` 를 붙이고 메모를 남기면 된다.
- ✓ 여러 줄 주석은 `/*` `*/` 를 사용하면 된다.

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

int main()
{
    int c = 15;
    int d = -2;

    c *= d; // c에 -2 값을 곱한다.

    //출력결과 : 15 -30
    cout << c << " " << d;

    return 0;
}
```

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

int main()
{
    int x;

    x = 10;
    x++; //x에 1 더한다.

    //x를 출력한다
    cout << x;

    return 0;
}
```

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

int main()
{
    int a = 10;
    int b = 15;

    a += b;
    /*
    a 값에는 처음에 10 이 들어있었고
    a 에다가 a + b 값을 넣기 때문에
    a 는 25가 된다.
    */
    cout << a;

    return 0;
}
```

if문

✓ 만약 조건이 성립하는 경우에만, 괄호 안에 있는 소스코드를 수행한다.

```
if ( 조건이 참인 경우 )  
{  
    // 소스코드  
}
```

```
int a = 10;  
if ( a > 5 )  
{  
    cout << "#";  
}  
//a가 5보다 크기 때문에, 콘솔 화면에 #이 출력된다.
```

if-else문

- ▶ “그렇지 않다면”이라는 뜻으로 쓰인다.
- ▶ else 단독으로 쓸 수 없고, 항상 if 문 뒤에 붙여서 써야 한다.

```
int a;  
cin >> a;  
if (a >= 10)  
{  
    cout << "입력받은 a 값은 10보다 같거나 크다";  
}  
else  
{  
    cout << "입력받은 a 값은 10보다 작다!";  
}
```

정상적인 소스코드

```
int a;  
cin >> a;  
  
else  
{  
    cout << "이것은 에러";  
}
```

컴파일 에러발생!
else 단독으로 쓸 수 없다.

if에 들어가는 부등호

`>=` : 크거나 같다. (**순서 주의**, `=>` 가 아니라 `>=` 로 써야한다.)

`<=` : 작거나 같다.

`==` : 같다. (**개수 주의**, `=` 이 아니라 `==`)

`!=` : 같지 않다.

```
int b, c;

cin >> b >> c;

if (b == c)
{
    cout << "같은 수를 입력 함";
}
else
{
    cout << "다른 수를 입력 함";
}
```

```
int a;

cin >> a;

if (a != 3)
{
    cout << "a는 3이 아닙니다";
}
else
{
    cout << "a는 " << a << "입니
다.";
}
```

if에서 자주하는 실수

```
int a = 10;
if (a = 5) //버그발생
{
    cout << "a는 5 입니다";
}
```

→ "a가 5 와 같습니까?" 라고 묻는 것은
if (a == 5) 이렇게 작성해야 한다.

→ **if (a = 5)** 라고 코딩하면 안된다

위 경우 a에 숫자 5가 들어간 후, if 문은 항상 참이 된다.
문법 Error가 아니기에, 버그를 찾아내기 어렵다.

소스코드 해석 1

▶ 소스코드를 해석 해보세요.

문제) 만약 입력 받은 숫자가 3보다 같거나 크고 5보다 작을 때 "와우 " 출력

```
int a;
cout << "입력하세요 : ";
cin >> a;

if (3 >= a)
{
    if (5 < a)           //이렇게 if안에 if를 넣을 수 있습니다.
    {
        cout << "와우";
    }
}
```


소스코드 해석 2

▶ 소스코드를 해석 해 보세요

문제) 세 수의 곱이 음수라면 "곱창" 출력, 그리고 세 수의 합이 100보다 크면 "스테이크" 출력

```
int a, b, c;
```

```
if (a * b * c < 0)
{
    cout << "곱창";
}
```

```
if (a + b + c > 100)
{
    cout << "스테이크";
}
```

예습 : for문 체험하기

- ✓ for문은
특정 소스코드를 반복하고자 할 때 사용한다.

'#' 을
1500회
반복 출력

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

int main()
{
    int x;

    for (x = 0; x < 1500; x++)
    {
        cout << "#";
    }

    return 0;
}
```