



C++ 스터디

3. 조건문, 반복문, switch문

C++의 boolean (True or False)

- C++의 논리형은 boolean 타입으로 정의 되며 비교 연산('>', '==', '<') 등은 boolean 타입 값을 반환한다.
- 꼭 true나 false 같은 boolean 타입이 아니더라도 (int, double 등) 조건문에 들어가면 알아서 boolean 타입으로 **타입 캐스팅** (타입 변화)가 일어난다.
 - 예시)
 - 0 -> false (int)
 - 1.3 -> true (double)
 - 5 -> true (int)
 - 0.0 -> false (double)
 - 'h' -> true (char)
 - '\0' (null) -> false (char)
 - 0 == 0 -> true

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     if (5 > -6) {
6         cout << "hi" << endl;
7     }
8 }
9
```

Microsoft Visual Studio 디버깅 콘솔: hi

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     if ('\0') {
6         cout << "hi" << endl;
7     }
8 }
9
```

Microsoft Visual Studio 디버깅 콘솔: 종료되었습니다. 이 창을 닫으려면 아무 키를 누르십시오.

if-else문 (문법)

- 조건부 실행
- if문 안의 내용은 if문 조건 충족 시 실행
- else문 안의 내용은 if문 조건 충족 실패 시 실행

```
if ( 조건(bool) )  
    실행할 내용  
else  
    실행할 내용
```

실행할 내용이 한 줄일 경우
괄호를 생략 가능하다.

```
if ( 조건(bool) )  
{  
    실행할 내용 #1  
    ~  
    실행할 내용 #n  
}  
else  
{  
    실행할 내용 #1  
    ~  
    실행할 내용 #n  
}
```

if-else문 (예시)

- C++의 컴파일러는 공백과 줄 바꿈을 무시하기에 아래의 10번과 11번 문장처럼 한 문장이 길다면 쪼개서 쓸 수도 있다. 그래도 한 문장이기에 괄호를 생략 가능하다.

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int dividend, divisor; // 나뉘지는 수, 나눌 수
6
7      cout << "나눌 두 정수를 입력해주세요 : ";
8      cin >> dividend >> divisor;
9      if (divisor != 0) // 나누는 수가 0이 아니면
10         cout << dividend << '/' << divisor << " = "
11         << dividend / divisor << endl;
12     else
13         cout << "0으로 나누는 것은 안됩니다.\n";
14 }
15
```

Microsoft Visual Studio 디버그 콘솔

나눌 두 정수를 입력해주세요 : 5 2
5/2 = 2

C:\Users\bluej\Desktop\방학프로그램
종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.

if-else문 (예시)

- 공백과 줄 바꿈을 무시한다는 것은 세 문장이 동일하다는 것이다.
- 보통은 첫번째 경우처럼 작성한다.

```
if (x < 10)  
    y = x;
```

```
if (x < 10) y = x;
```

```
if (x < 10)  
y = x;
```

if-else문 (else if)

- 실행 내용 1 : if문 조건을 충족 시 실행
- 실행 내용 2 : if문 조건 충족 안하고 else if 조건 만족 시 실행
- 실행 내용 3 : if문과 상위의 else if문 조건 만족 안하고 else if 조건 만족 시 실행
- 실행 내용 4 : if, else if문 조건 모두를 충족 못했을 시 실행

```
if ( 조건(bool) ) {  
    실행 내용 1  
}  
else if( 조건 (bool) ) {  
    실행 내용 2  
}  
else if( 조건 (bool) ) {  
    실행 내용 3  
}  
else {  
    실행 내용 4  
}
```

if-else문 (else if 예시)

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int i;
6      cout << "숫자를 입력해 주세요 : ";
7      cin >> i;
8      if (i < 5) {
9          cout << "5보다 작습니다!" << endl;
10     }
11     else if (i < 10) {
12         cout << "5와 같거나 크고 10보다 작습니다!\n";
13     }
14     else {
15         cout << "10과 같거나 큼니다.";
16     }
17 }
```

Microsoft Visual Studio 디버그 콘솔

```
숫자를 입력해 주세요 : 5
5와 같거나 크고 10보다 작습니다!
```

while문 (문법)

- while문의 조건이 true인 동안 반복한다.

```
while ( 조건(bool) )  
    실행할 내용
```

```
while ( 조건(bool) )  
{  
    실행할 내용 #1  
    ~  
    실행할 내용 #n  
}
```


while문 (예시)

- 반복문을 사용하여 일을 줄일 수 있다. 두 프로그램의 결과는 동일하다.

Microsoft Visual Studio 디버그 콘솔

```
1  
2  
3  
4  
5
```

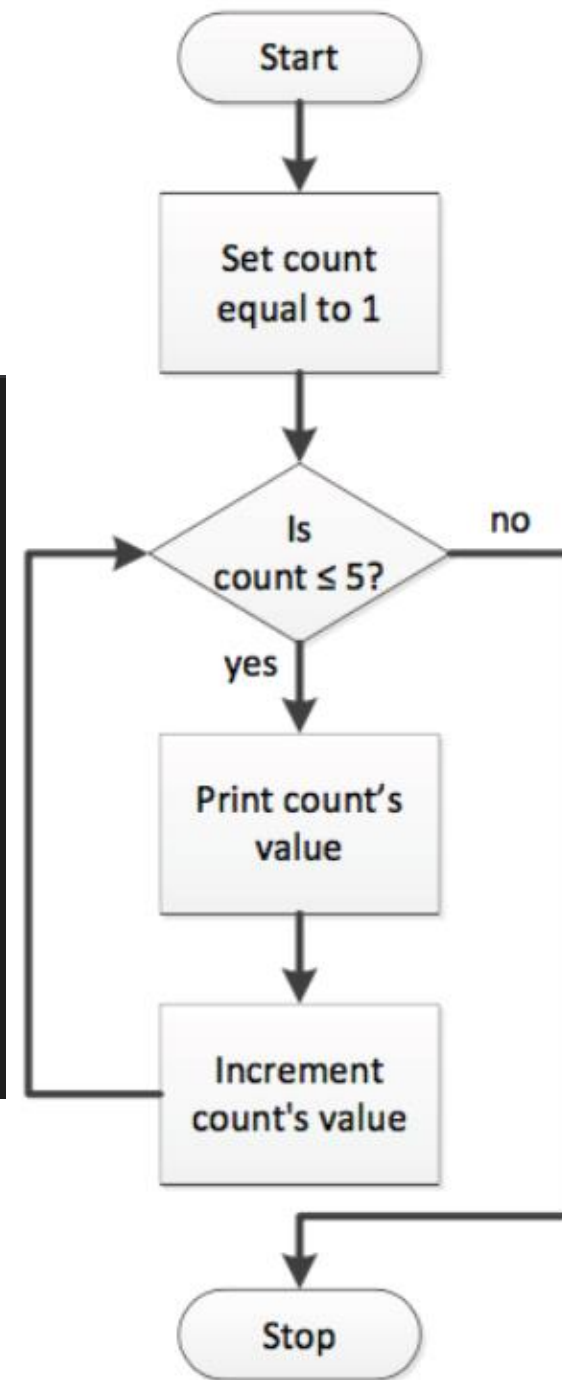
```
1  #include <iostream>  
2  using namespace std;  
3  
4  int main() {  
5      cout << "1\n" << endl;  
6      cout << "2\n" << endl;  
7      cout << "3\n" << endl;  
8      cout << "4\n" << endl;  
9      cout << "5\n" << endl;  
10 }
```

```
1  #include <iostream>  
2  using namespace std;  
3  
4  int main() {  
5      int count = 1;  
6      while (count <= 5) {  
7          cout << count << endl;  
8          count++; // count = count + 1  
9      }  
10 }
```

while문 (예시)

- 반복문을 사용하여 일을 줄일 수 있다.

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int count = 1;
6      while (count <= 5) {
7          cout << count << endl;
8          count++; // count = count + 1
9      }
10 }
```



while문 (do/while)

```
do{  
    실행할 내용  
}while ( 조건(bool) );
```

- while문과 다른 점은 조건을 검사하기 전에 do문을 실행한다는 점이다.
- 이런 것도 있구나 하고 넘어가면 된다.

while문 (do/while 예시)

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int in_value;
6      cout << "0~10의 정수를 입력해주세요 : ";
7      do
8          cin >> in_value;
9      while (in_value < 0 || in_value > 10); // and
10     cout << "범위의 숫자 " << in_value << "가 입력됨\n";
11 }
```

Microsoft Visual Studio 디버그 콘솔

0~10의 정수를 입력해주세요 : -1

-5

11

5

범위의 숫자 5가 입력됨

C:\Users\bluej\Desktop\방학프로그래밍
종료되었습니다.

이 창을 닫으려면 아무 키나 누르세요.

break, continue문은 Python과 사용법이 같다.

- break는 반복문을 빠져나오게 하는 명령어이고
- continue는 반복문이 끝나지 않아도, 다시 반복하게 해주는 문구이다.

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int i = 0;
6      while (1) { // int -> bool 타입캐스트
7          if (i == 5)
8              break;
9          cout << ++i << endl; // 덧셈 먼저하고 출력
10     }
11 }
```

Microsoft Visual Studio 디버그 콘솔

1
2
3
4
5

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int i = 0;
6      while (i < 5) {
7          i++;
8          if (i % 2 == 0)
9              continue;
10         cout << i << endl;
11     }
12 }
```

Microsoft Visual Studio 디버그 콘솔

1
3
5

for문 (while문과 비교)

- while문

```
값 초기화  
while ( 조건 ) {  
    수행할 행동  
    값 변경  
}
```

- for문

```
for( 값 초기화; 조건; 값 변경) {  
    수행할 행동  
}
```

for문 (문법)

- for 문은 초기식, 조건식, 증감식을 모두 포함하고 있는 반복문이다.
- 초기식 -> (조건식 -> 반복문 -> 증감식) 순으로 진행 한다. 괄호는 반복하며 조건식을 불만족시 저 괄호를 빠져나온다.

```
for (초기식; 조건식; 증감식) {  
    수행할 행동  
}
```

for문 (예시)

```
1 #include <iostream>
2 #include <iomanip> // setw가 있는 라이브러리
3 using namespace std;
4
5 int main() {
6     int size;
7     cout << "표의 크기를 정해주세요 : ";
8     cin >> size;
9
10    cout << "  ";
11    for (int col = 1; col <= size; col++)
12        cout << setw(4) << col; // 왼쪽 여백 4칸 설정
13    cout << '\n';
14
15    cout << "  +";
16    for (int col = 1; col <= size; col++)
17        cout << "----";
18    cout << '\n';
19    for (int row = 1; row <= size; row++) {
20        cout << setw(4) << row << " |";
21        for (int col = 1; col <= size; col++) {
22            cout << setw(4) << row * col;
23        }
24        cout << '\n';
25    }
26 }
```

Microsoft Visual Studio 디버그 콘솔

표의 크기를 정해주세요 : 5
1 2 3 4 5

	+	-----				
1		1	2	3	4	5
2		2	4	6	8	10
3		3	6	9	12	15
4		4	8	12	16	20
5		5	10	15	20	25

C:\Users\bluej\Desktop\방학프로그
종료되었습니다.
이 창을 닫으려면 아무 키나 누르세

switch문 (문법)

- switch문은 if / else 문과 비슷한 조건문이다.
- break를 각 조건마다 달아주지 않는다면 처음 실행된 case를 수행하고 밑에 있는 모든 코드도 실행한다.
- 보통은 if else문으로 다 적지만 퀴즈나 시험에 자주 나온다.

```
switch (변수) {  
    case 변수조건1:  
        행동1  
        break;  
    case 변수조건2:  
        행동2  
        break;  
    default:  
        행동3  
        break;  
}
```

switch문 (예제)

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int value;
6     cout << "0~3의 정수를 입력하세요";
7     cin >> value;
8     switch (value)
9     {
10     case 0:
11         cout << "영";
12         break;
13     case 1:
14         cout << "일";
15         break;
16     case 2:
17         cout << "이";
18         break;
19     case 3:
20         cout << "삼";
21         break;
22     default:
23         if (value < 0)
24             cout << "넘 작음";
25         else
26             cout << "넘 크다";
27         break;
28     }
29     cout << "\n";
30 }
```

Microsoft Visual Studio 디버그 콘솔

0~3의 정수를 입력하세요 : 1
일

C:\Users\bluej\Desktop\방학프로그램
종료되었습니다.
이 창을 닫으려면 아무 키나 누르세

과제 3번

- 랩 3번을 푸세용~

