

# P1.S4

⌚ 작성일시	@2024년 5월 26일 오전 1:47
📄 강의 번호	C++ 언리얼
📄 유형	강의
☑ 복습	<input type="checkbox"/>
📅 날짜	@2024년 5월 26일

## 분기문

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

// 오늘의 주제 : 분기문
// 데이터를 메모리에 할당하고 가공하고 방법에 대해 알아봄
// 가공한 데이터를 이용해서 무엇인가를 하고 싶다면?

int main()
{
    int hp = 100; // 몬스터 HP
    int damage = 90; // 플레이어 데미지

    hp -= damage; // 피격 판정
    bool isDead = (hp <= 0); // 처치 판정

    // 몬스터가 죽었으면 경험치 추가
    // 어셈블리에서 CMP JMP

    // if-else if-else
    if (isDead)
        cout << "몬스터를 처치했습니다" << endl;
    else if (hp <= 20)
        cout << "몬스터가 도망가고 있습니다" << endl;
    else
        cout << "몬스터가 반격했습니다" << endl;
```

```

const int ROCK = 0;
const int PAPER = 1;
const int SCISSORS = 2;

int input = ROCK;

if (input == ROCK)
    cout << "바위를 냈습니다" << endl;
else if (input == PAPER)
    cout << "보를 냈습니다" << endl;
else if (input == SCISSORS)
    cout << "가위를 냈습니다" << endl;
else
    cout << "뭘 냈습니까?" << endl;

// switch-case-break

switch (input)
{
case ROCK:
    cout << "바위를 냈습니다" << endl;
    break;
case PAPER:
    cout << "보를 냈습니다" << endl;
    break;
case SCISSORS:
    cout << "가위를 냈습니다" << endl;
    break;
default:
    cout << "뭘 냈습니까?" << endl;
    break;
}
}

```

반복문

```

#include <iostream>
using namespace std;

// 오늘의 주제 : 반복문
// 데이터를 메모리에 할당하고 가공하고 방법에 대해 알아봄
// 가공한 데이터를 이용해서 무엇인가를 하고 싶다면?

int main()
{
    // while ~동안에
    // if-else 굉장히 유용하다
    // 하지만 한 번만 실행하는게 아니라, 특정 조건까지 계속 반복해야 하는
    // ex) 게임을 끝때까지 계속 게임을 실행해라
    // ex) 목적지에 도달할때까지~ 계속 이동하라

    int count = 0;

    while (count < 5)
    {
        cout << "Hello World" << endl;
        count++;
    }

    //do
    //{
    //  cout << "Hello World" << endl;
    //    count++;
    //} while (count < 5);

    for (int c = 0; c < 5; c++)
    {
        cout << "Hello World" << endl;
    }

    // break; continue;
    // 루프문의 흐름제어

```

```

int round = 1;
int hp = 100;
int damage = 27;

// 무한 루프 : 전투시작
while (true)
{
    hp -= damage;
    if (hp < 0)
        hp = 0;

    // 시스템 메시지
    cout << "Round " << round << " 몬스터 체력 " << hp << endl;

    if (hp == 0)
    {
        cout << "몬스터 처치!" << endl;
        break;
    }

    if (round == 5)
    {
        cout << "제한 라운드 종료" << endl;
        break;
    }

    round++;
}

for (int i = 1; i <= 10; i++)
{
    bool isEven = (i % 2) == 0;

    if (isEven)
        continue;

    cout << i << endl;
}

```

```
}
```

## 별찍기와 구구단

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

int main() {
    // 별찍기)
    // 유저들이 어떤 정수를 입력하면 N*N개의 별을 찍었으면 좋겠어요!

    int input;
    cin >> input;

    for (int i = 0; i < input; i++) {
        for (int j = 0; j < input; j++) {
            cout << "*";
        }
        cout << endl;
    }

    // 별찍기2)
    // 음.. 그냥 N*N별찍기는 너무 심심하다
    // 1개부터 시작해서 줄마다 순차적으로 증가하게 해주세요

    int input2;
    cin >> input2;

    for (int i = 0; i < input2; i++) {
        for (int j = 0; j < i + 1; j++) {
            cout << "*";
        }
        cout << endl;
    }

    // 별찍기3)
    // 모양이 많이 개선된
```

```

// N개부터 시작해서 줄마다 1개씩 줄인다

int input3;
cin >> input3;

for (int i = 0; i < input3; i++) {
    for (int j = 0; j < input - i; j++) {
        cout << "*";
    }
    cout << endl;
}

// 구구단)
// 2단부터 9단까지 구구단을 출력해주세요

for (int i = 1; i < 9; i++) {
    for (int j = 0; j < 9; j++) {
        cout << i + 1 << "*" << j + 1 << "=" << (i + 1) *
    }
}
}

```