

민선생코딩학원 훈련반

수업노트 LV-20

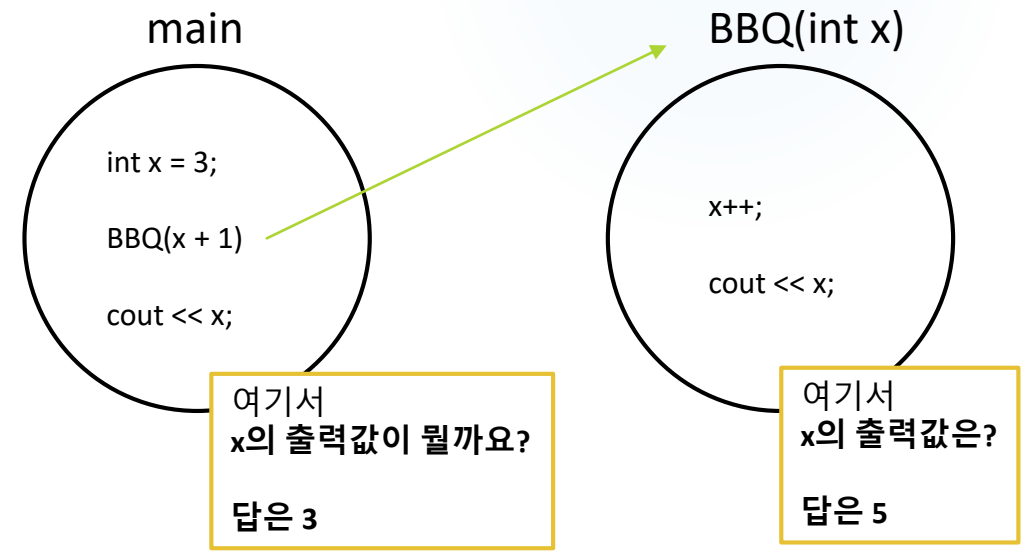
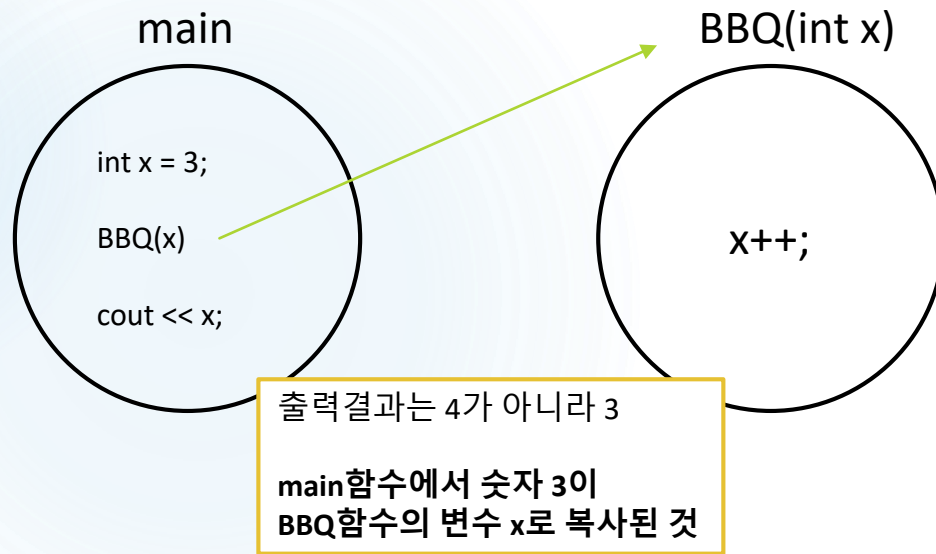


배우는 내용

1. 함수 복습하기
2. 재귀호출 1단계 : 무한 재귀호출 막는법
3. 재귀호출 1단계 : 재귀 문제 풀이

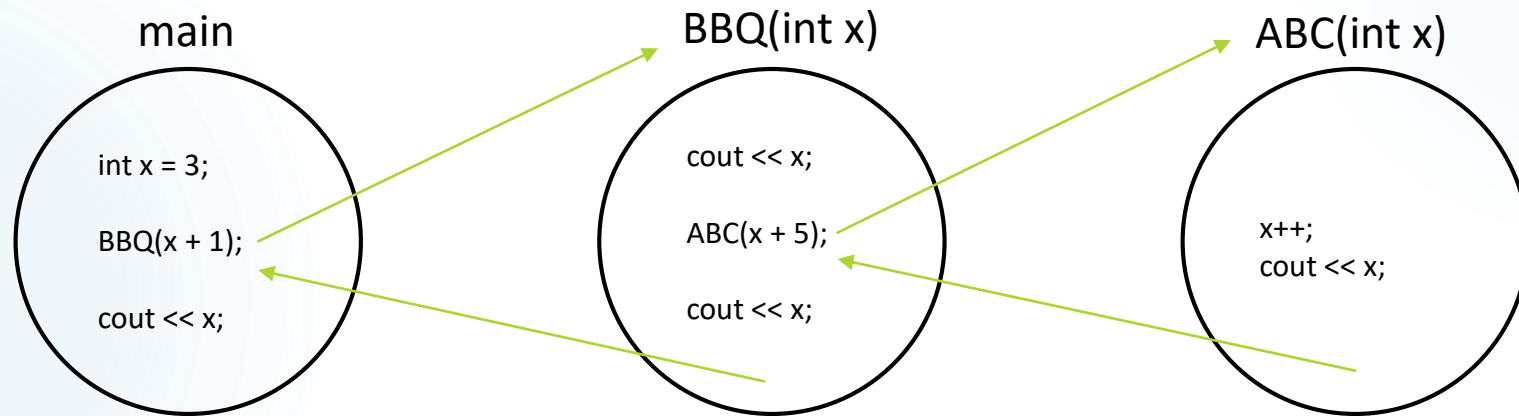
함수 복습 - 값 전달하기

- ▶ BBQ 함수를 호출할 때 값을 넘기는 코드
- ▶ main 함수안에 있는 변수 x와 BBQ 함수안에 있는 변수 x는 이름만 같고 서로 다른 변수



함수 복습 - return

- ▶ 함수가 종료되면, 호출한 곳으로 돌아감 (return 됨)
- ▶ 모든 프로그램은 main에서 시작하고 main에서 끝남



이 프로그램을 실행하면 출력 결과는?
→ 4 10 4 3

호출할 때 $x+1$ 과 $x++$ 의 차이

▶ $abc(x+1)$

- $x + 1$ 값을 abc 함수로 보냄
- x 변수값을 바꾸지 않는다.

▶ $abc(x++)$

- x 값 그대로 abc 로 보냄
- 보낸 뒤에, x 변수값에다가 1 더함

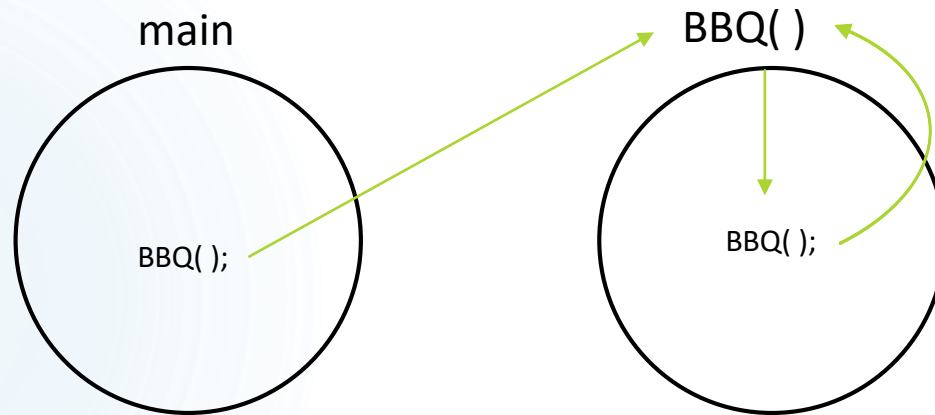
재귀함수란?

- ▶ 함수 안에서 자기 자신을 호출하는 함수
- ▶ 재귀함수를 이용한 코딩 기법을 배우게 됨

```
void abc()  
{  
    ...  
    abc();  
}
```

무한 재귀호출

- ▶ BBQ에서 BBQ로 호출하는 것 같이, 자기 자신을 호출하는 함수를 재귀함수라고 함
- ▶ 결과적으로, 무한 함수 호출이 발생 하면서 Runtime Error 발생



for문의 한계

숫자 1개를 입력받고, 입력받은 만큼 for문 돌리기

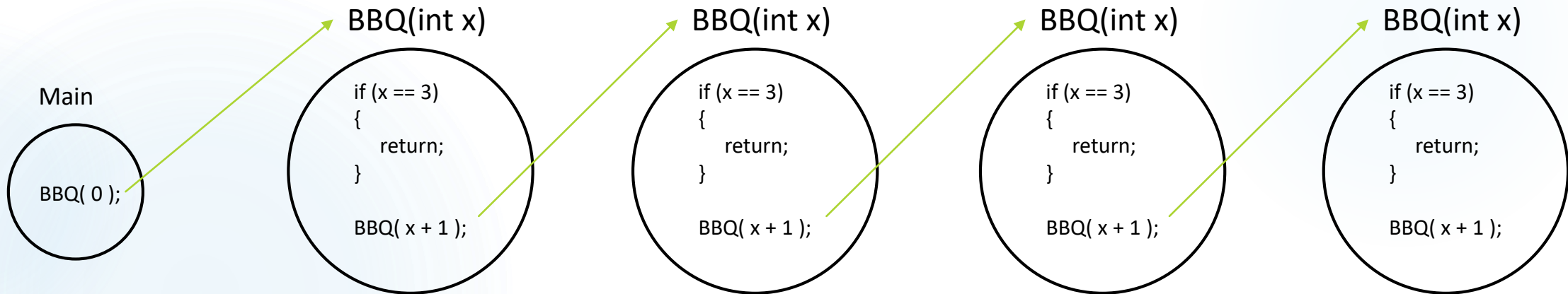
1을 입력받으면 1중 for문
2를 입력받으면 2중 for문
3을 입력받으면 3중 for문
...
30을 입력받으면 30중 for문

→ For문으로 풀기 힘들지만, 재귀호출로 풀 수 있음

→ For는 특정 소스코드를 반복할 때 사용하고,
모든 경우를 시도해보는 프로그램을 짤 때는 재귀호출로 구현 가능하다

일단 무한 재귀호출 막는 방법부터

- ▶ 무한 재귀 호출을 막는 가장 좋은 방법 소개 (**$x++$ 이 아니라 $x + 1$ 임을 주의**)

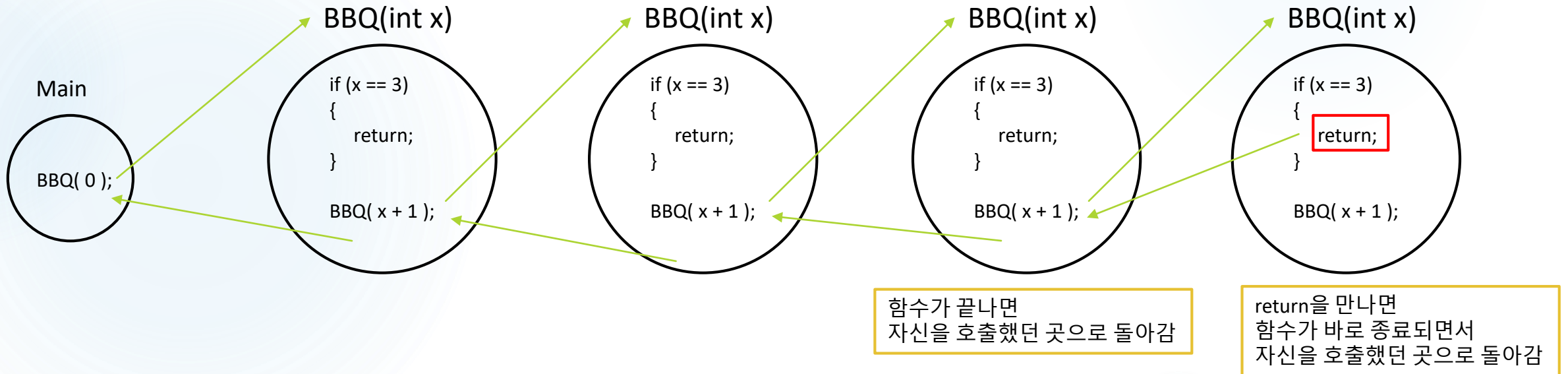


if (x == 3) return; 코드에 의해 무한 재귀를 막을 수 있음

이 지역변수 x 를 이용하여, 몇 번 재귀 호출할지 정할 수 있음

무한 재귀호출 막는 법

- ▶ x 변수만 보면 0 ~ 3번 시점부터 어떤 시점의 함수인지 알 수 있음
- ▶ 만약 x 가 2라면 세 번째 BBQ 함수라는 것을 바로 알 수 있음



n번 재귀호출 해보기

- ▶ 출력결과 : 0123
- ▶ level이라는 변수가 3이 될 때
숫자 3 출력 후 return 되기 시작한다.

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

void bbq(int level)
{
    cout << level;
    if (level == 3)
    {
        return;
    }

    bbq(level + 1);
}

int main()
{
    bbq(0);

    return 0;
}
```

출력결과 예상하기1

- ▶ 세 소스코드는 실행결과가 모두 다르다. 실행결과 맞춰보기

```
void bbq(int level)
{
    if (level == 3)
    {
        return;
    }

    cout << level;
    bbq(level + 1);
}

int main()
{
    bbq(0);

    return 0;
}
```

결과 : 012

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

void bbq(int level)
{
    if (level == 3)
    {
        return;
    }

    bbq(level + 1);
    cout << level;
}

int main()
{
    bbq(0);

    return 0;
}
```

결과 : 210

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

void bbq(int level)
{
    cout << level;

    if (level == 3)
    {
        return;
    }

    bbq(level + 1);
}

int main()
{
    bbq(0);

    return 0;
}
```

결과 : 0123

출력결과 예상하기2

▶ 실행결과 맞춰보기

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

void bbq(int level)
{
    bbq(level + 1);

    if (level == 3)
    {
        return;
    }

    cout << level;
}

int main()
{
    bbq(0);

    return 0;
}
```

결과 : Runtime Error, 무한재귀

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

void bbq(int level)
{
    cout << level;

    if (level == 3)
    {
        return;
    }

    bbq(level + 1);

    cout << level;
}

int main()
{
    bbq(0);

    return 0;
}
```

결과 : 0123210

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

void bbq(int level)
{
    if (level == 3)
    {
        return;
    }

    cout << level;
    bbq(level + 1);
    cout << level;
}

int main()
{
    bbq(0);

    return 0;
}
```

결과 : 012210

들어가며 출력, 리턴하면서 출력1

- ▶ 123454321 출력하는 예제

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

void abc(int level)
{
    cout << level;
    if (level == 5)
    {
        return;
    }

    abc(level + 1);
    cout << level;
}

int main()
{
    abc(1);

    return 0;
}
```

들어가며 출력, 리턴하면서 출력2

▶ 12344321 출력

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

void abc(int level)
{
    if (level == 5)
    {
        return;
    }

    cout << level;
    abc(level + 1);
    cout << level;
}

int main()
{
    abc(1);

    return 0;
}
```

배열 값 재귀 호출로 출력하기

5	7	1	2	3
---	---	---	---	---

- ▶ 57123 출력 후 역으로 32175 까지 출력
- ▶ 실행결과 : 57123**32175**

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

int vect[5] = { 5, 7, 1, 2, 3 };
void abc(int level)
{
    if (level == 5)
    {
        return;
    }

    cout << vect[level];
    abc(level + 1);
    cout << vect[level];
}

int main()
{
    abc(0);

    return 0;
}
```