

민선생코딩학원 시작반

수업노트 LV-10



배우는 내용

1. return 키워드
2. 2차배열의 연습 – 숫자 채우기
3. 글자 바뀌서 출력하기

return 키워드

- ▶ 함수 호출했던 곳으로 돌아간다는 키워드
→ 함수를 즉시 종료 하는 것

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

void abc()
{
    int t = 10;

    if (t == 10)
    {
        return;
    }

    cout << "2";
}

int main()
{
    abc();

    return 0;
}
```

숫자 2는
출력 안됨

값을 return하기

```
int abc()  
{  
    return 5;  
}
```

- ▶ 호출했던 곳으로 돌아가면서
값 5를 가지고 돌아간다.

→ **abc 함수는 숫자 5를 return하는 함수**

- ▶ [C언어 규칙]

숫자를 return하면 함수 이름 앞에 int를 써주고
문자를 return하면 함수 이름 앞에 char를 써준다

→ return 하는 값의 타입을
함수 이름 옆에 반드시 적어줘야 한다

값을 return받는 소스코드

- ▶ 만약 abc() 라는 함수가 숫자 5를 갖고 return한다면?
main에서 받아주면 된다.

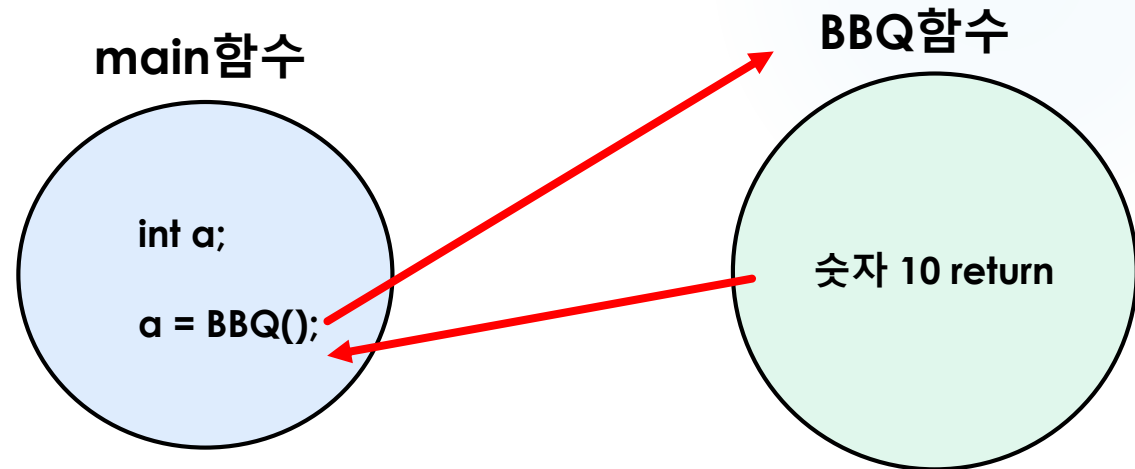
```
int main()  
{  
    int ret;  
  
    ret = abc();  
  
    return 0;  
}
```

이 부분이 숫자 5로 바뀌어
ret = 5가 된다.

함수에서 값을 return 받는 전체 소스코드

이곳에 어떤 Type을 return할지 적어 주어야 함
이 예제는 숫자를 return 해 줄 것이기 때문에 int를 적음

```
int BBQ( )  
{  
    return 10;  
}  
  
int main()  
{  
    int a;  
    a = BBQ( );  
    cout << a;  
    return 0;  
}
```



값을 return하는 코드 예제

- ▶ KFC에서 숫자 2개 입력받고 합을 return하는 소스코드

```
int KFC()  
{  
    int a, b;  
    cin >> a >> b;  
    return a + b;  
}  
  
int main()  
{  
    cout << "SUM = " << KFC();  
    return 0;  
}
```

여기서 KFC함수가 호출되고
return되는 숫자로 바뀌어서
결국 입력받은 두 수의 합이 출력 됨

함수호출 유의사항1

- ▶ main함수는 프로그램이 시작할 때 자동으로 호출되는 함수
- ▶ **main함수 호출 할 필요없이 ABC함수가 끝나면 자동으로 main으로 돌아감**

```
void ABC( )  
{  
    main( ); //main함수를 호출하지 말자  
}  
  
int main()  
{  
    ABC( );  
    return 0;  
}
```

출력결과 : 무한 함수 호출 발생

- ▶ return 명령어를 만나면 함수가 즉시 종료 됨

```
void ABC( );  
{  
    cout << "#";  
    return;  
    cout << "@"; //수행안됨, 필요없는 코드  
}  
  
int main()  
{  
    ABC( );  
    return 0;  
}
```

출력결과 : #

함수호출 유의사항2

- ▶ return할 값이 없으면
함수 앞에 void를 써 주어야 함

```
void ABC(int x)
{
    if (x > 7)
    {
        return;
    }

    cout << "###";
}

int main()
{
    ABC(5);
    return 0;
}
```

출력결과 : ###

- ▶ return값이 있으면
함수 앞에 return type을 써 주어야 함

```
char ABC(char a, char b)
{
    if (a > b)
    {
        return a;
    }

    return b;
}

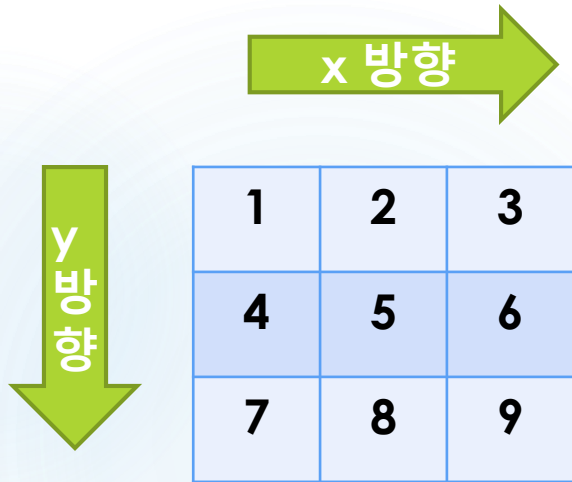
int main()
{
    char max = ABC('D', 'Q');
    cout << max;
    return 0;
}
```

D, Q 값을 보내고
MAX를 return받음

출력결과 : Q

2차 배열의 연습 - 숫자 채우기1

번호 순서대로 숫자를 배열에 채우기

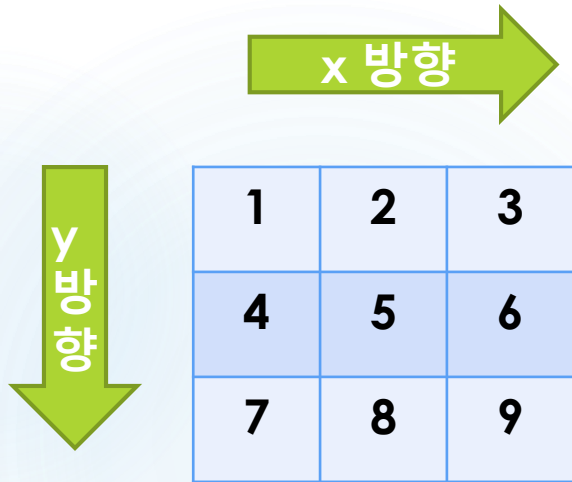


- ▶ x방향이 바쁘게 움직이고, y방향은 x방향이 3번 움직일때마다 한번씩 움직인다.
- ▶ 따라서 아래 구조대로 for를 돌려야 한다.

```
for (y...) //덜 바쁘게 움직이는 y
{
    for (x...) //바쁘게 움직이는 x
    {
        //...
    }
}
```

2차 배열의 연습 - 숫자 채우기2

번호 순서대로 숫자를 배열에 채우기



- ▶ x는 0 ~ 2까지
y는 0 ~ 2까지 반복한다.

```
int vect[3][3] = { 0 };

int x, y;
int t = 1;
for (y = 0; y < 3; y++)
{
    for (x = 0; x < 3; x++)
    {
        vect[y][x] = t;
        t++;
    }
}
```

2차 배열의 연습 - 숫자 채우기3

7	4	1
8	5	2
9	6	3

- ▶ y방향이 바쁘게 움직이고,
x방향이 천천히 움직인다.
- ▶ x는 2 부터 0까지 움직이고,
y는 0 부터 2까지 움직인다.

2차 배열의 연습 - 숫자 채우기4

7	4	1
8	5	2
9	6	3

- ▶ y방향이 바쁘게 움직이고,
x방향이 천천히 움직인다.
- ▶ x는 2 부터 0까지 움직이고,
y는 0 부터 2까지 움직인다.

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

int main()
{
    int vect[3][3] = { 0 };

    int x, y;
    int t = 1;

    for (x = 2; x >= 0; x--)
    {
        for (y = 0; y < 3; y++)
        {
            vect[y][x] = t;
            t++;
        }
    }

    return 0;
}
```

글자 바꿔서 출력하기

1 일때 #으로,
0 일때 @으로 바꾸어서
출력하는 코드를 작성하시오

1	0	0
0	1	1
1	0	1

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

int main()
{
    int vect[3][3] = {
        1, 0, 0,
        0, 1, 1,
        1, 0, 1,
    };

    int x, y;
    for (y = 0; y < 3; y++)
    {
        for (x = 0; x < 3; x++)
        {
            if (vect[y][x] == 1)
            {
                cout << "#";
            }
            else
            {
                cout << "@";
            }
        }
        cout << endl;
    }

    return 0;
}
```