



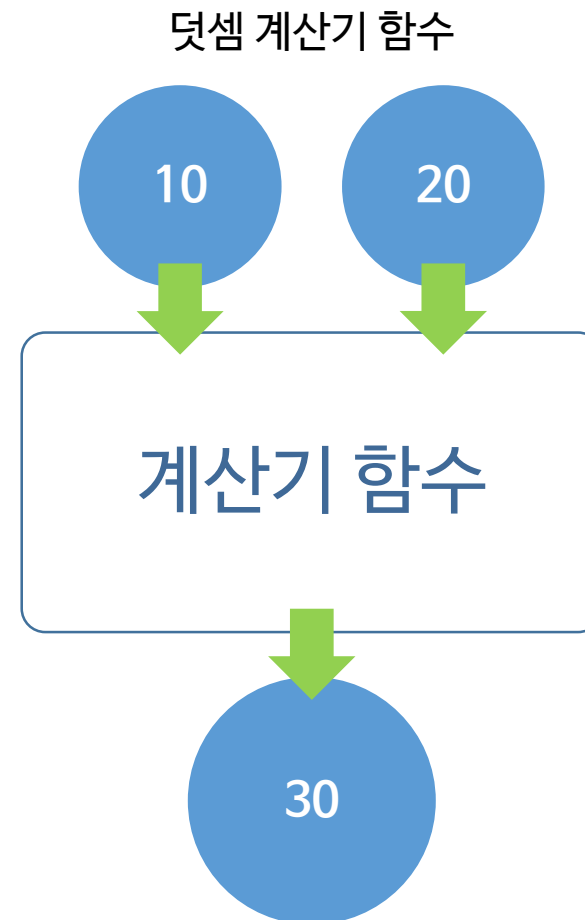
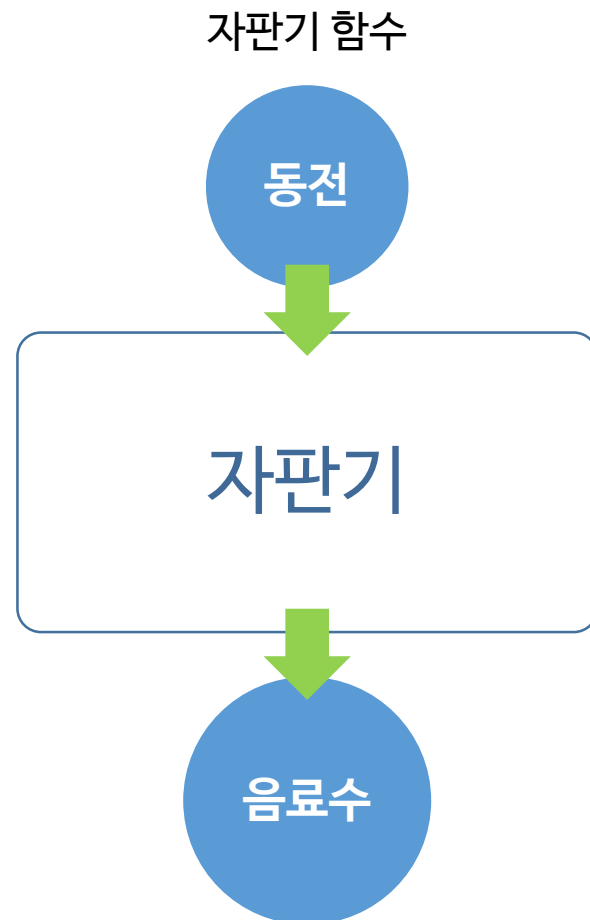
11강_함수

특정 기능을 구현해 놓고 재사용할 수 있는 방법에 대해서 학습합니다.

- 11-1 함수란?
- 11-2 함수 만들기
- 11-3 함수의 다양한 형태
- 11-4 매개변수
- 11-5 디폴트 매개변수
- 11-6 중복함수

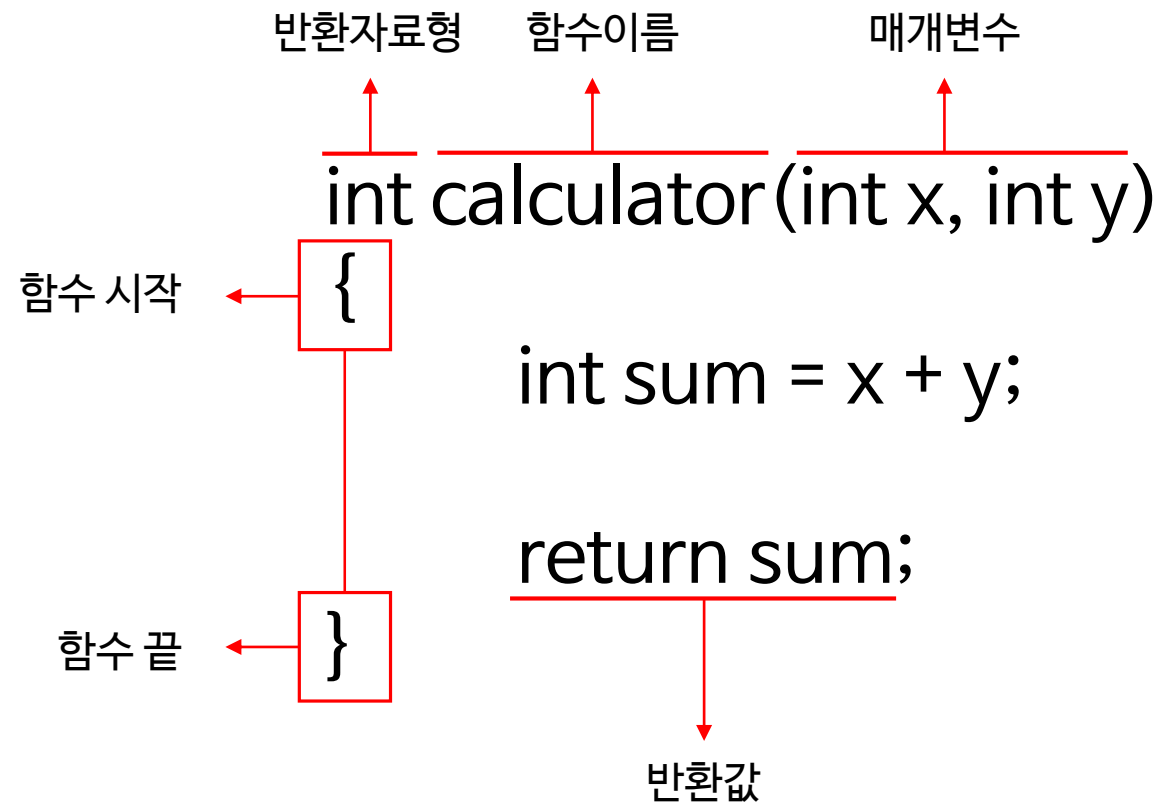
11-1 : 함수란?

특정기능을 수행하는 코드 덩어리



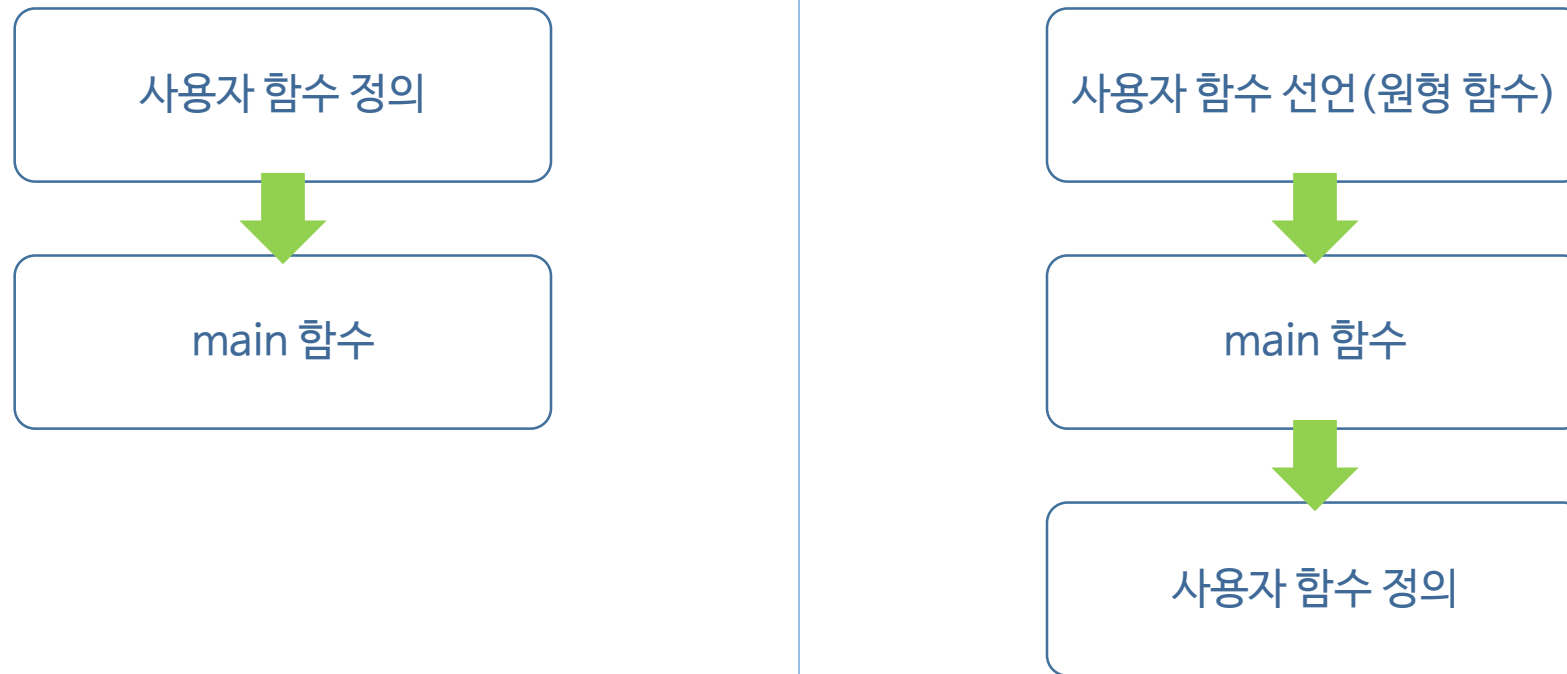
11-2 : 함수 만들기

함수이름, 매개변수, 반환값, { ... } 를 이용해서 함수를 만들 수 있다.



11-3 : 함수의 다양한 형태

사용자 함수는 main함수 위에 정의하는 것과 원형함수를 이용하는 방법이 있다.




11-4 : 매개변수


매개변수는 값을 이용하는 방법과 참조를 이용하는 방법이 있다.

```
int cbv(int);    // call-by-value  
int cbr(int&);   // call-by-reference
```

```
int a = 10;  
  
cout << "call-by-value" << endl;  
cout << cbv(a) << endl;  
cout << "a : " << a << endl;
```




```
int cbv(int x)  
{  
    x = x + 1;  
  
    return x;  
}
```




```
call-by-value  
11  
a : 10
```

```
cout << "call-by-reference" << endl;  
cout << cbr(a) << endl;  
cout << "a : " << a << endl;
```



```
int cbr(int& x)  
{  
    x = x + 1;  
  
    return x;  
}
```



```
call-by-reference  
11  
a : 11
```

11-5 : 디폴트 매개변수

매개변수를 디폴트로 정의해서 호출 시 매개변수가 없어도 함수가 정상적으로 실행할 수 있다.

```
void gugudan(int x = 9)
{
    for (int i = 1; i <= 9; i++)
    {
        cout << x << " * " << i << " = " << (x * i) << endl;
    }
}
```



```
gugudan();  
cout << endl;
```



```
9 * 1 = 9  
9 * 2 = 18  
9 * 3 = 27  
9 * 4 = 36  
9 * 5 = 45  
9 * 6 = 54  
9 * 7 = 63  
9 * 8 = 72  
9 * 9 = 81
```



```
gugudan(5);  
cout << endl;
```



```
5 * 1 = 5  
5 * 2 = 10  
5 * 3 = 15  
5 * 4 = 20  
5 * 5 = 25  
5 * 6 = 30  
5 * 7 = 35  
5 * 8 = 40  
5 * 9 = 45
```

11-6: 중복함수

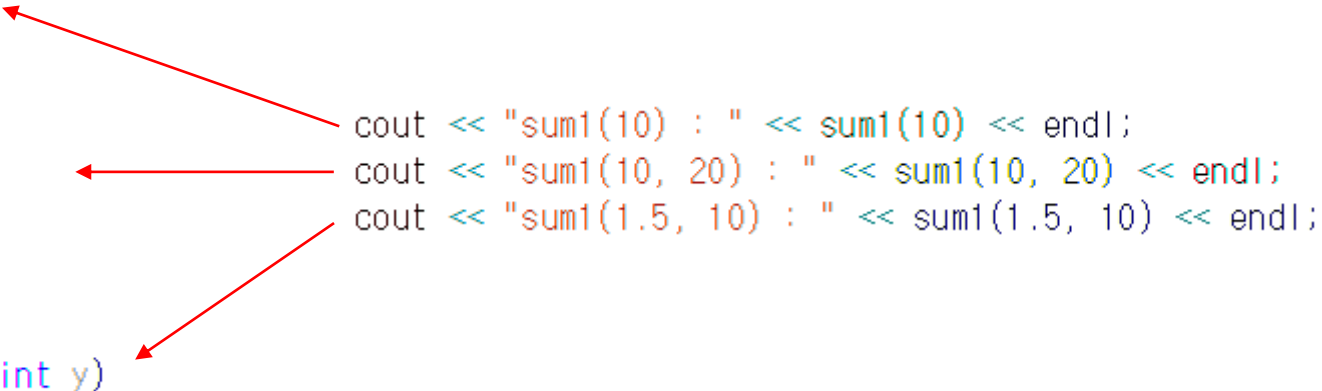
이름이 같은 함수를 매개변수에 다양하게 정의할 수 있다.

```
int sum1(int x)
{
    return x + x;
}

int sum1(int x, int y)
{
    return x + y;
}

double sum1(double x, int y)
{
    return x + y;
}
```

cout << "sum1(10) : " << sum1(10) << endl;
cout << "sum1(10, 20) : " << sum1(10, 20) << endl;
cout << "sum1(1.5, 10) : " << sum1(1.5, 10) << endl;



```
sum1(10) : 20
sum1(10, 20) : 30
sum1(1.5, 10) : 11.5
```