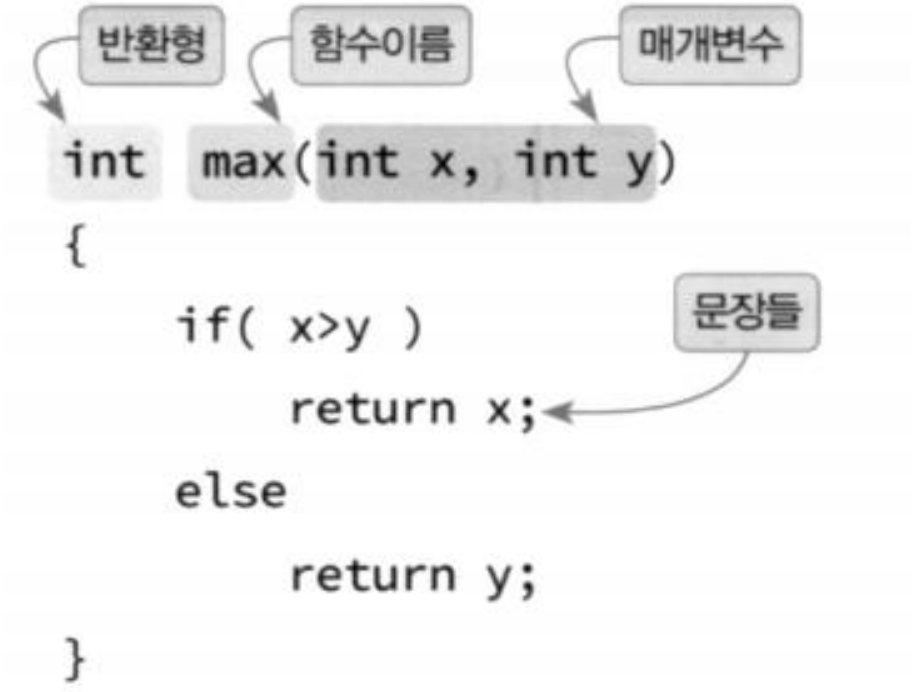
함수와 문자열

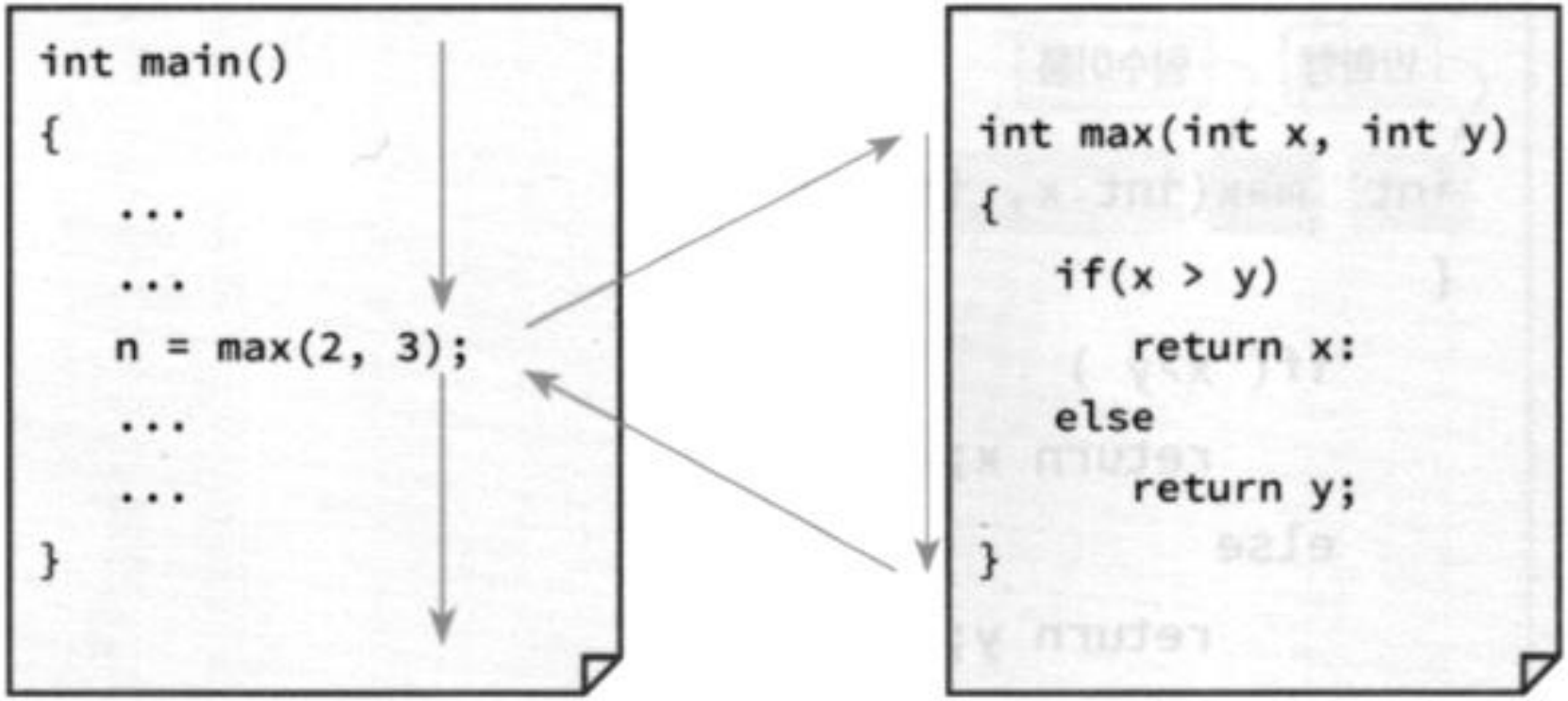
함수

■ 함수의 구조


반환형
함수이름
매개변수
int max(int x, int y)
{
if(x>y)
문장들
return x;
else
return y;
}

함수

■ 함수의 호출



함수

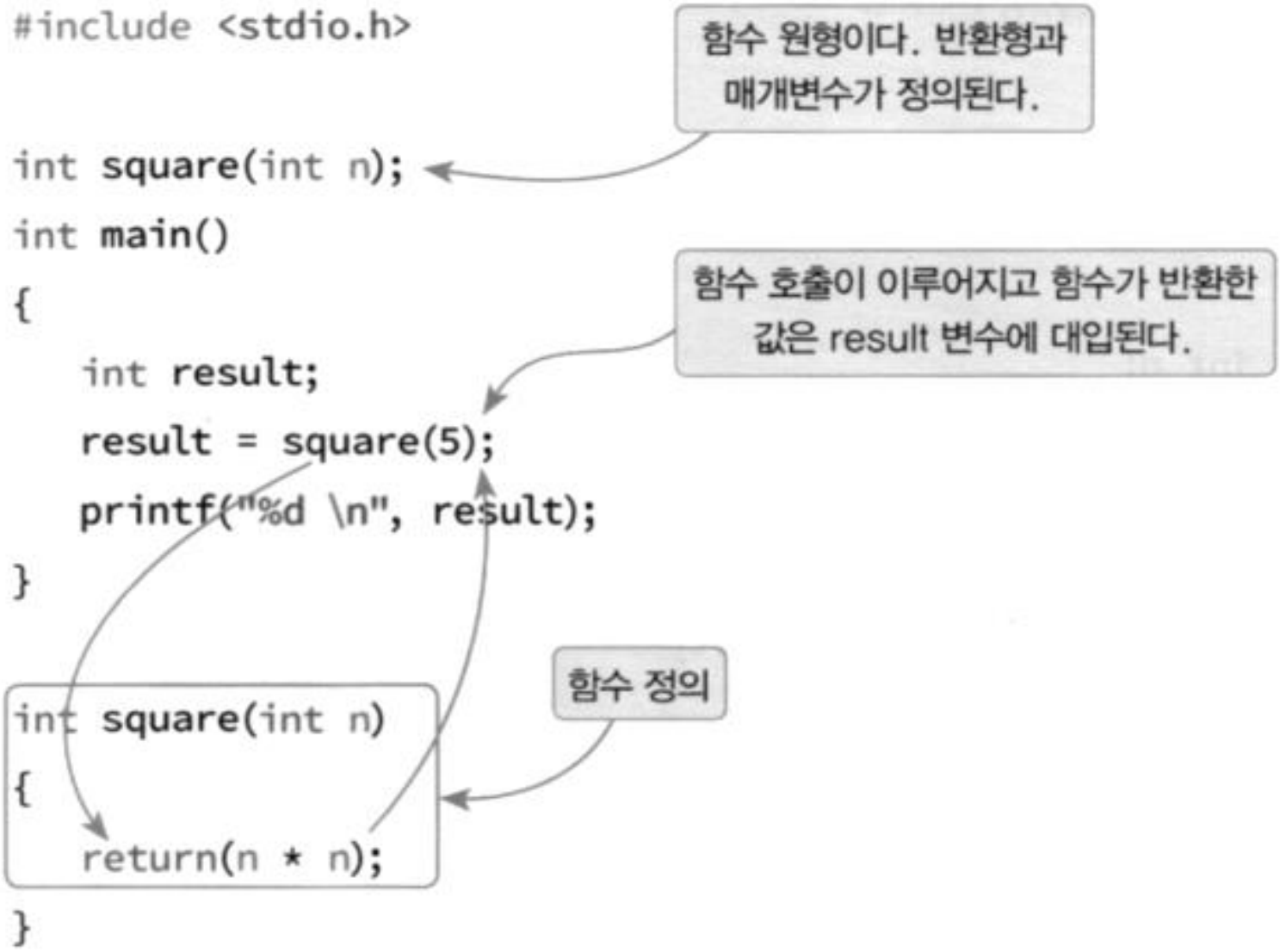
■ chapter03/ex01_func.cpp] 함수

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

int max(int x, int y) {
  if(x>y)
    return x;
  else
    return y;
}

int main(int argc, char const *argv[]) {
  int n;
  n = max(2, 3);
  cout << "함수 호출 결과 : " << n << endl;
  return 0;
}
```

```
함수 호출 결과 : 3
```

함수

■ 함수 원형


#include <stdio.h>
int square(int n);
int main()
{
int result;
result = square(5);
printf("%d \n", result);
}
int square(int n)
{
return(n * n);
}
함수 원형이다. 반환형과 매개변수가 정의된다.
함수 호출이 이루어지고 함수가 반환한 값은 result 변수에 대입된다.
함수 정의

함수

■ 함수 원형


매개 변수의 이름은 생략하여도 된다.
반드시 끝에 ;을 붙여야 한다.
int square(int);
int get_integer (void);

함수

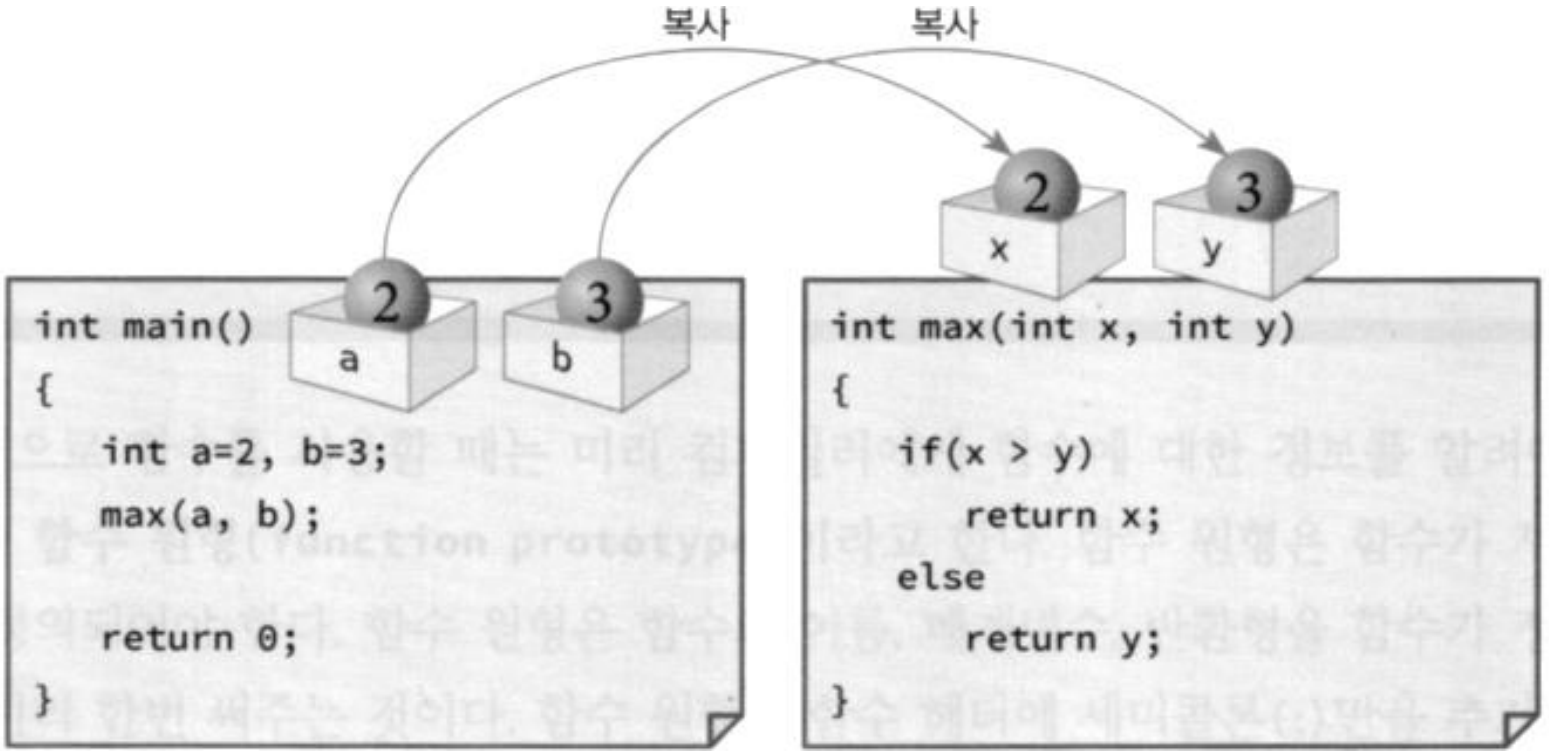
■ 함수 인자 전달방법

- call by value
- call by reference
- call by address(pointer)

함수

■ 함수 인자 전달 방법

○ call by value



함수

- chapter03/ex02_call_by_value.cpp] call by value

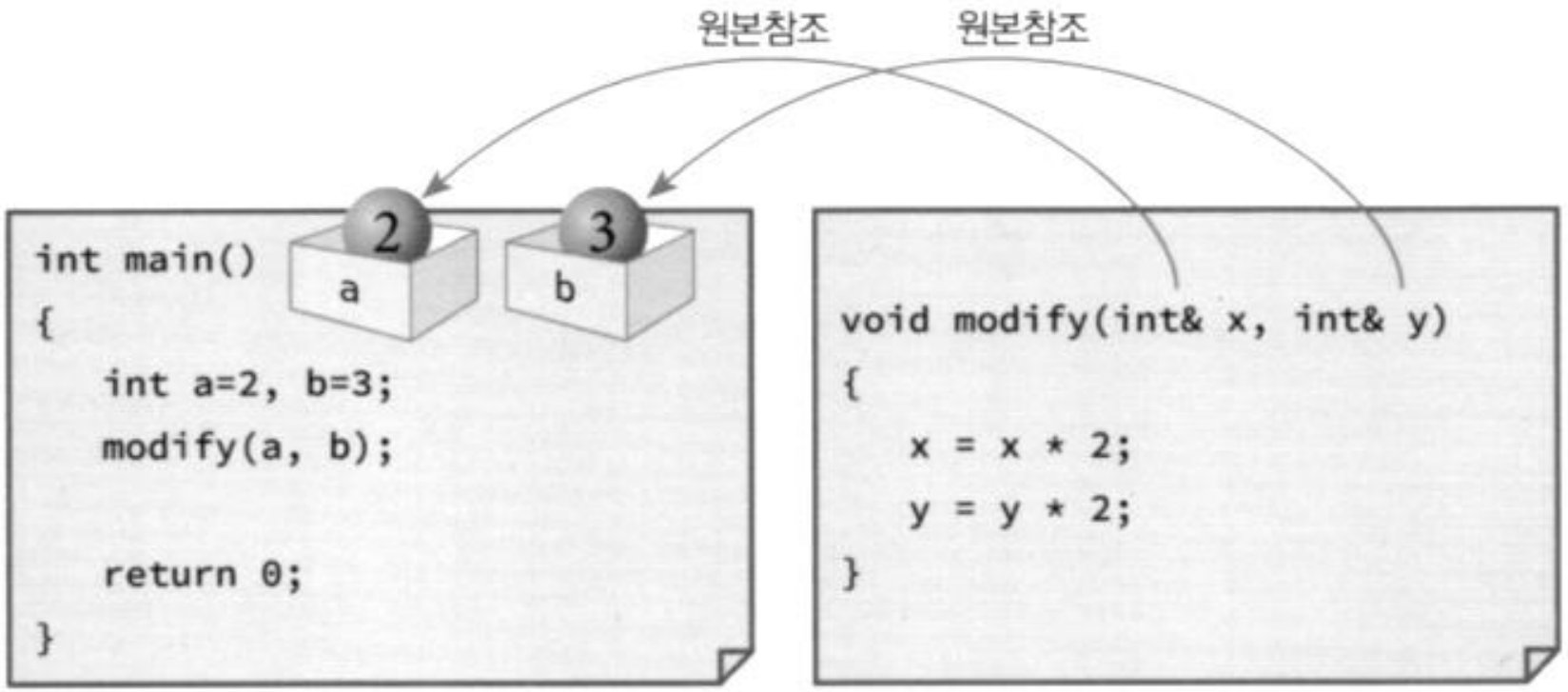
```
#include <iostream>
using namespace std;

void swap(int x, int y) {
  int t;
  t = x;
  x = y;
  y = t;
}

int main(int argc, char const *argv[]) {
  int a = 100, b = 200;

  printf("a=%d, b=%d\n", a, b);
  swap(a, b);
  printf("a=%d, b=%d\n", a, b);
  return 0;
}
```

```
a=100, b=200
a=100, b=200
```

함수

■ 함수 인자 전달 방법

- call by reference



함수

■ chapter03/ex03_call_by_reference.cpp] call by reference

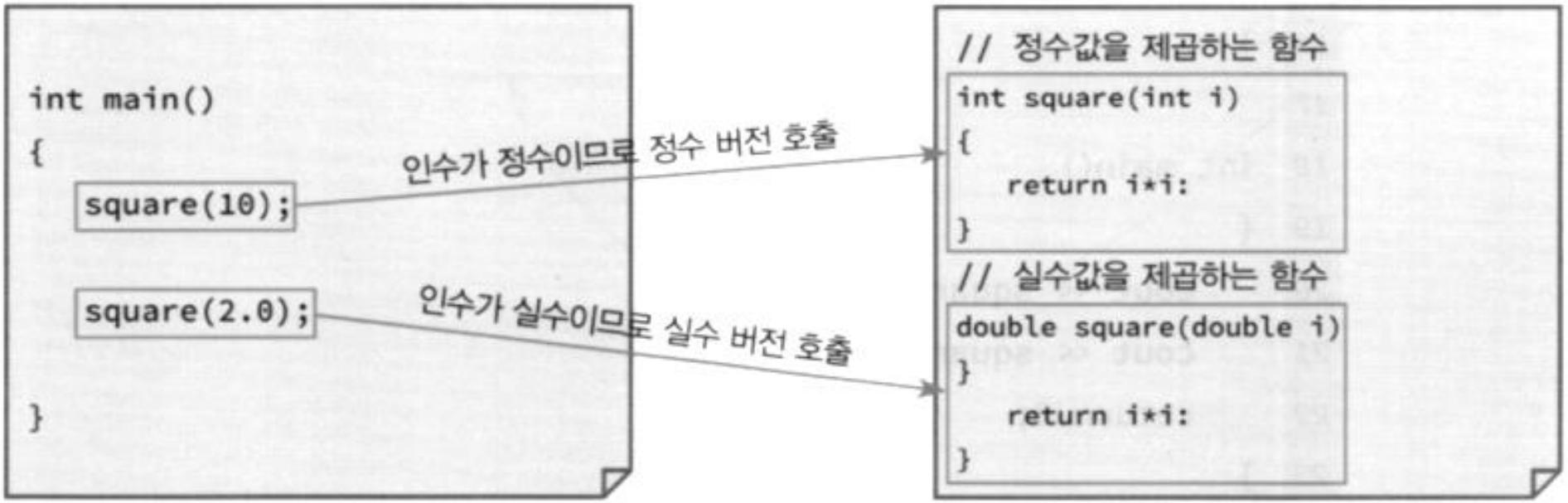
```
#include <iostream>
using namespace std;

void swap(int& x, int &y) {
  int t;
  t = x;
  x = y;
  y = t;
}

int main(int argc, char const *argv[]) {
  int a = 100, b = 200;

  printf("a=%d, b=%d\n", a, b);
  swap(a, b);
  printf("a=%d, b=%d\n", a, b);
  return 0;
}
```

```
a=100, b=200
a=200, b=100
```

함수

■ 중복 함수(overload)

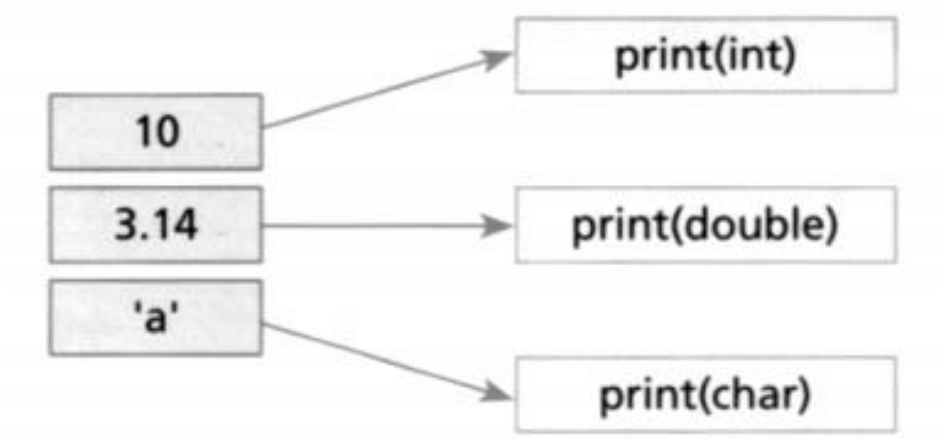
- ○ 함수의 이름은 동일하지만 함수의 인자가 다르면 다른 함수로 인식
- ○ 리턴 타입은 상관없음



함수

- 중복 함수(overload)

```
void print(int);
void print(double);
void print(char);
```



함수

■ chapter03/ex04_overload.cpp] 중복 함수(overload)

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

int square(int i) {
  cout << "square(int) 호출" << endl;
  return i*i;
}

double square(double i) {
  cout << "square(double) 호출" << endl;
  return i*i;
}

int main(int argc, char const *argv[]) {
  cout << square(10) << endl;
  cout << square(2.0) << endl;
  return 0;
}
```

```
square(int) 호출
100
square(double) 호출
4
```

함수

■ 인수의 디폴트 값

○ 함수 호출시 인수 값을 지정하지 않았을 때 가지는 값

■ chapter03/ex05_default_param.cpp] 중복 함수(overload)

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

void display(char c = '*', int n = 10) {
  for(int i=0; i<n; i++) {
    cout << c;
  }
  cout << endl;
}

int main(int argc, char const *argv[]) {
  display();
  display('#');
  display('#', 5);
  return 0;
}
```

```
**********
##########
#####
```

함수

■ 인수의 디폴트 값 지정시 주의 사항

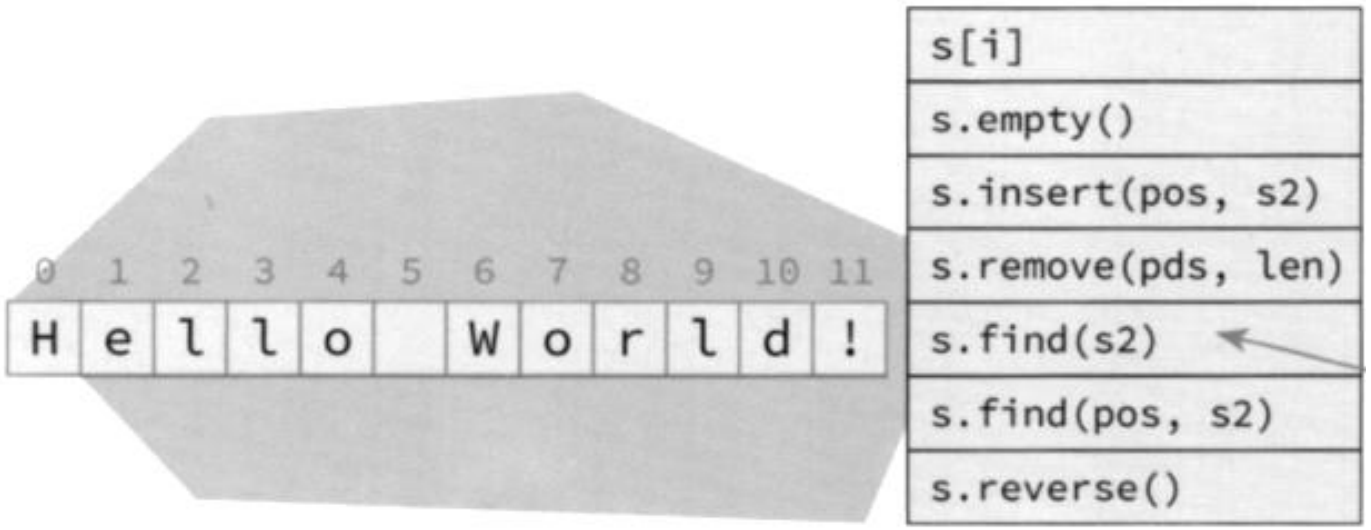
- ○ 뒤에서 부터 배정
- ○ 앞에서 부터 배정하는 경우 에러

```
int print( double dvalue = 0.0, int prec ); // 오류
```

문자열

■ string 클래스

- ○ 문자열 데이터 저장 및 문자열 처리 함수(메서드) 제공
- ○ `#include <string>`을 먼저 지정 후 사용



문자열

■ chapter03/ex06_string.cpp] string 클래스

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

int main(int argc, char const *argv[]) {
  string s1 = "Slow", s2 ="steady";
  string s3 = "the race.";
  string s4;

  s4 = s1 + " and " + s2 + " wins " + s3;

  cout << s4 << endl;

  return 0;
}
```

```
Slow and steady wins the race.
```

문자열

■ chapter03/ex07_string.cpp] string 클래스

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

int main(int argc, char const *argv[]) {
  string s1, addr;

  cout << "이름을 입력하세요: " ;
  cin >> s1;
  cin.ignore(); // 엔터키 제거

  cout << "주소를 입력하세요: ";
  getline(cin, addr);

  cout << addr << "의" << s1 << "씨 안녕하세요?" << endl;
  return 0;
}
```

```
이름을 입력하세요:  홍길동
주소를 입력하세요: 서울시 종로구
서울시 종로구의홍길동씨 안녕하세요?
```

문자열

■ string 클래스

멤버 함수	설명
s[i]	i번째 원소
s.empty()	s가 비어있으면 true 반환
s.insert(pos, s2)	s의 pos 위치에 s2를 삽입
s.remove(pos, len)	s의 pos 위치에 len만큼을 삭제
s.find(s2)	s에서 문자열 s2가 발견되는 첫번째 인덱스를 반환
s.find(pos, s2)	s의 pos 위치부터 문자열 s2가 발견되는 첫번째 인덱스를 반환

문자열

■ chapter03/ex08_string.cpp] string 클래스

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

int main(int argc, char const *argv[]) {
  string s = "When in Rome, do as the Romans.";

  int size = s.size();
  int index = s.find("Rome");

  cout << size << endl;
  cout << index << endl;

  return 0;
}
```

```
31
8
```

문자열

■ chapter03/ex09_string.cpp] string 클래스

```
#include <iostream>
#include <string>

using namespace std;

int main(int argc, char const *argv[])
{
  string s = "When in Rome, do as the Romans.";

  for(auto& ch: s) {
    cout << ch << ' ';
  }

  return 0;
}
```

```
W h e n   i n   R o m e ,   d o   a s   t h e   R o m a n s .
```

문자열

■ chapter03/ex10_string.cpp] string 클래스

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

int main(int argc, char const *argv[]) {
  string list[] = { "홍길동", "고길동", "둘리" } ;

  for(auto& name: list) {
    cout << name << endl;
  }

  return 0;
}
```

```
홍길동
고길동
둘리
```