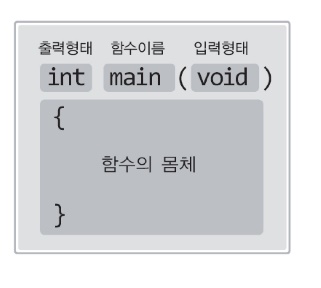
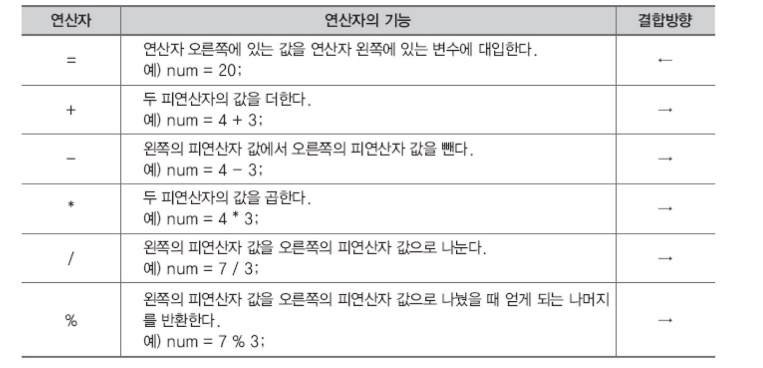
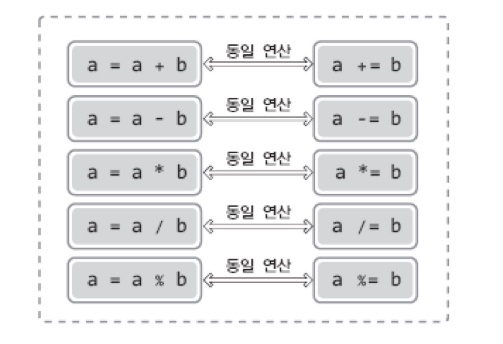
**main함수의 형태**

****

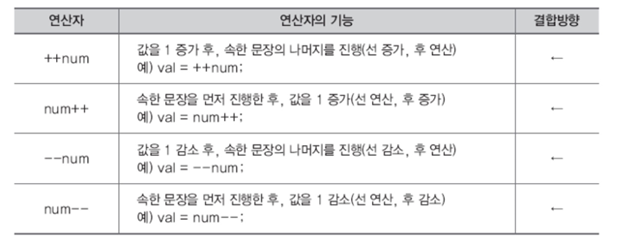
**대입 연산자, 산술 연산자**

****

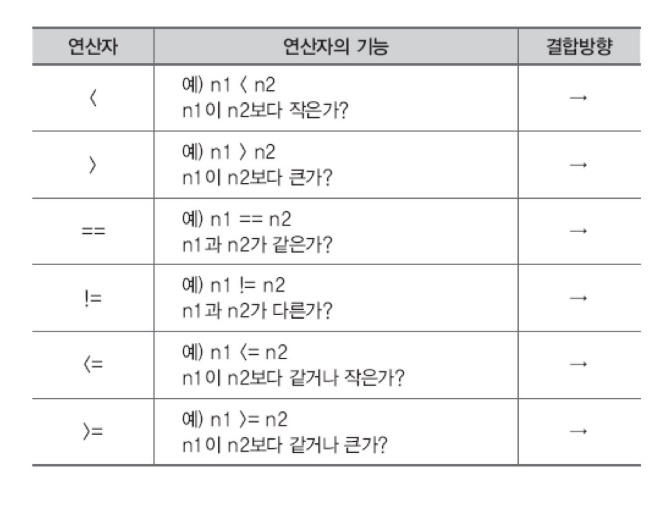
**복합 대입 연산자**

****

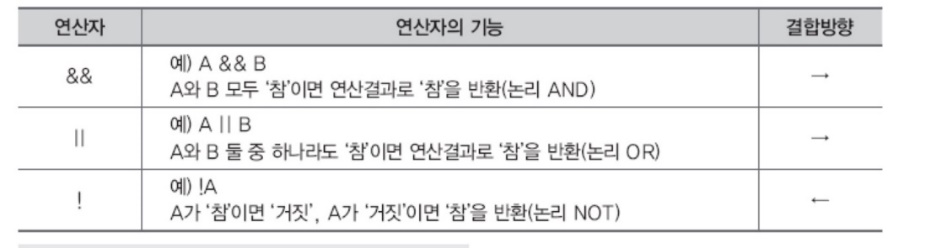
**증가, 감소 연산자**

****

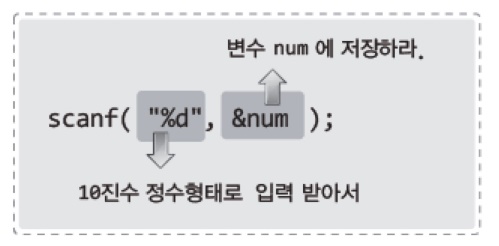
**관계 연산자**

****

**논리 연산자**

****

**scanf함수의 형태**

****

**for 문의 형태**

**텍스트, 폰트, 라인, 스크린샷이(가) 표시된 사진

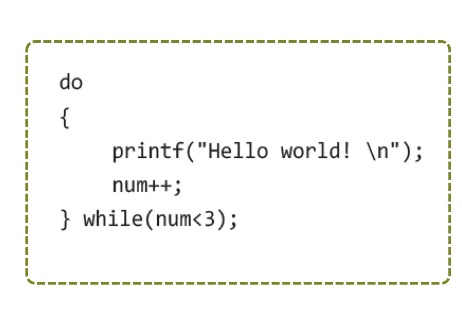
자동 생성된 설명**

**while문의 형태**

**텍스트, 폰트, 스크린샷, 직사각형이(가) 표시된 사진

자동 생성된 설명**

**do~while문의 형태**

****

**if/if-else/if-elseif-else문의 형태**

**텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

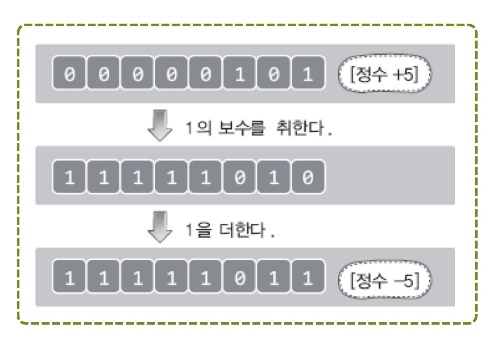
자동 생성된 설명텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 폰트, 도표이(가) 표시된 사진

자동 생성된 설명**

**음의 정수 표현방식**

: 1의 보수를 취한다 -> 1을 더한다.



**double형 변수의 출력 서식문자 %f -printf**

**double형 변수의 입력 서식문자 %lf -scanf**

**unsigned : 0과 양의 정수만 표현**

**명시적 형 변환: 강제로 일으키는 형 변환**

* divResult = (double)num1/num2; : (type)은 type형으로의 형 변환을 의미한다.

**특수문자의 종류**

**텍스트, 스크린샷, 번호, 폰트이(가) 표시된 사진

자동 생성된 설명**

**printf함수의 서식지정과 서식문자들**

**텍스트, 스크린샷, 번호, 폰트이(가) 표시된 사진

자동 생성된 설명**

**필드 폭을 지정하여, 정돈된 출력 보이기**

**%8d** : 필드 폭을 8칸 확보하고, 오른쪽 정렬해서 출력을 진행한다.

**%-8d** : 필드 폭을 8칸 확보하고, 왼쪽 정렬해서 출력을 진행한다.

**정수 기반의 입력형태 정의하기**

**%d** : 10진수 정수의 형태로 데이터를 입력 받는다.

**%o** : 8진수 양의 정수의 형태로 데이터를 입력 받는다.

**%x** : 16진수 양의 정수의 형태로 데이터를 입력 받는다.

**실수 기반의 입력형태 정의하기**

float, double, long double의 데이터 출력 : %f, %f, %Lf

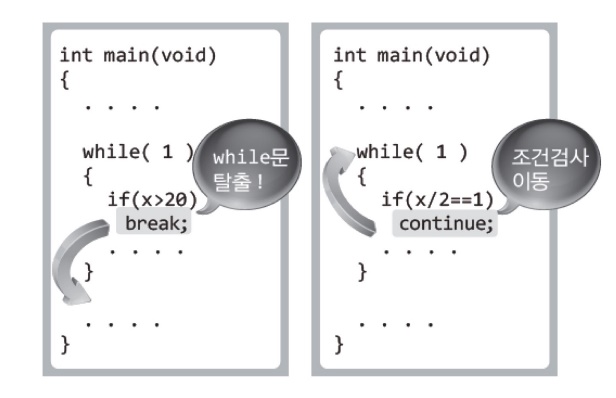
float, double, long double의 데이터 입력 : %f, %lf, %Lf

**Break**

: break문은 자신을 감싸는 반복문 하나를 빠져나가는 역할을 함.

**Continue**

: 나머지를 생략하고 반복조건을 확인함. 반복조건이 참이면 반복영역을 처음부터 실행함.



**switch문**

**텍스트, 스크린샷, 폰트, 도표이(가) 표시된 사진

자동 생성된 설명**

**goto문**

**텍스트, 폰트, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명**

**함수의 구분(전달인자/반환 값)**

반환 값 x -> 반환형태 : void -> return x/printf

인자전달 x -> 입력형태 : void -> scanf(&num), return(num)

**return;**

: '값의 반환'과 '함수의 탈출'이라는 기능이 있는데, return;은 **'함수의 탈출'** 기능을 한다.

**printf함수의 반환**

: printf 함수는 출력된 문자열의 길이를 반환하는데, 반환 값이 필요 없어서, 반환되는 값을 저장하지 않았을 뿐임.

**전역변수**

: 지역변수의 이름이 전역변수의 이름을 가림.

**static 변수**

: 지역변수와 전역변수의 특성을 동시에 가지는 변수

-> 선언된 함수내에서만 접근이 가능함.(지역변수 특성)

-> 딱 1회 초기화되고 프로그램 종료 시까지 메모리 공간에 존재한다.(전역변수 특성)

-> static 지역변수를 사용하여 전역변수의 선언 최소화

**register 변수**

: 빈번히 사용하는 변수이므로, 접근이 가장 빠른 레지스터에 저장하여 성능향상에 도움을 줌.

-> register : 힌트를 제공하는 키워드, 컴파일러는 이를 무시하기도 함.

-> register int num = 3;

**재귀함수**

: 자기자신을 재호출하는 형태로 정의된 함수, **탈출조건**이 꼭 필요함!