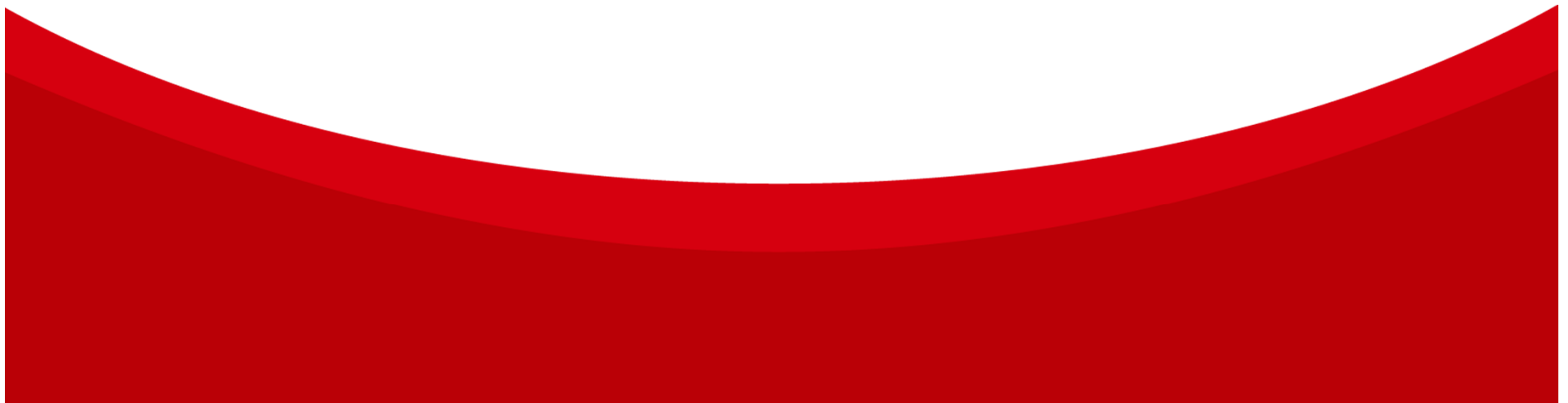


Chapter 2

입력 - 실습교안



목표

- 강의 교안에서 핵심적인 내용 재정리
- 추가의 실습이 필요한 내용 정리

정수 자료형의 크기 및 범위

자료형	크기	범위	비고
char signed char	1 바이트, 8비트	-128~127	
unsigned char	1 바이트, 8비트	0~255	
short short int	2바이트, 16비트	-32,768~32,767	int 생략 가능
unsigned short unsigned short int	2바이트, 16비트	0~65,535	int 생략 가능
int signed int	4바이트, 32비트	-2,147,483,648~ 2,147,483,647	
unsigned unsigned int	4바이트, 32비트	0~4,294,967,295	int 생략 가능
long long int signed long signed long int	4바이트, 32비트	-2,147,483,648~ 2,147,483,647	int 생략 가능
unsigned long unsigned long int	4바이트, 32비트	0~4,294,967,295	int 생략 가능
long long long long int signed long long signed long long int	8바이트, 64비트	-9,223,372,036,854,775,808~ 9,223,372,036,854,775,807	int 생략 가능
unsigned long long unsigned long long int	8바이트, 64비트	0~18,446,744,073,709,551,615	int 생략 가능

- 정수 자료형의 크기 및 범위
 - 자료형에 여러가지가 있는 경우 아무거나 써도 똑같은 효과
 - 예) char, signed char 선언해도 같은 효과
- int, long 은 달라보이지만 같은 자료형 임
 - long 형이 시스템에 따라 64bit가 될 수도 있지만 대부분 32bit임
 - long int를 줄여서 long 또는 int 로 부른다고 생각해도 됨
- unsigned 는 뒤에 아무 type이 없지만 int가 생략된 것임
- long long 은 1989년에 재정된 C89라 불리는 표준에는 없었으나, 1999년에 재정된 C99 부터 표준에 포함됨
- 출력방법
 - int 는 %d, long은 %ld, long long은 %lld 사용
 - printf("%d, %ld, %lld", intv, longv, longlongv);

- 실습 과제 1
 - 정수 자료형의 크기 및 범위 표의 모든 자료형 선언 방법을 사용해서 변수 선언 하기 (23개)
 - 최저 또는 최대값 할당한 후, 출력
- 출력 방법
 - max char c1 : 127
 - max signed char c2 : 127
 -
 - max unsigned long long int l9 :18446744073709551615

- 변수이름은 어떤 용도로 사용할 것인지에 따라 알아보기 쉽도록 의미 있게 정하지 않으면 나중에 소스코드를 보고 이해하기 어려움
- “반드시” 의미에 맞는 변수 사용 해야함
- i,j,k,l 등은 반복문 등에서 주로 사용하므로 의미를 담는 변수로 사용하면 안됨
- 의미를 담는 변수 예.
- count => 카운팅 하는 변수
- number => 숫자를 담고 있는 변수
- sum => 합을 담고 있는 변수
- num_student => 학생 숫자를 담고 있는 변수
- ...

소스

```
#include <stdio.h>

int main()
{
    int age;

    printf("당신의 나이는 몇 살입니까? ");
    scanf("%d", &age);

    printf("당신의 나이는 %d살 이군요. \n", age);

    return 0;
}
```

- `scanf("%d", &age);`
- `&` 는 앰퍼샌드라고 읽는다
- 연산자로도 사용되지만, 변수 앞에 단독으로 사용될 경우 해당 변수의 메모리 주소를 의미함
- 정수 하나를 읽어서 그 값을 `age`의 주소에 넣으라는 뜻
- `scanf` 에서 `&`를 빠뜨려서 많은 오류가 발생하니 특히 주의 필요!

Chapter 2

입력

소스

```
#include <stdio.h>

int main()
{
    int height;
    double weight;
    char name;

    printf("키를 입력하세요. ");
    scanf("%d", &height);

    printf("몸무게를 입력하세요. ");
    scanf("%lf", &weight);

    printf("이름을 입력하세요. ");
    scanf(" %c", &name);

    printf("키 = %d \n", height);
    printf("몸무게 = %.1f \n", weight);
    printf("이름 = %c \n", name);

    return 0;
}
```

- `scanf(" %c", &name);`

한 개의 문자를 입력받아 `name`에 저장한다.

“%c” 앞에 공백을 넣어 준 것에 유의해야 한다. 일반적인 자료형의 경우에는 이전에 입력된 `Enter`나 공백문자, 탭문자(`Tab`) 등을 입력구분자로 처리하여 무시하지만 문자형 자료(`char`)의 경우에는 모든 문자를 입력으로 처리하게 된다. 만약 공백을 생략하게 되면 위에서 `weight`를 입력받은 후 누른 `Enter`가 버퍼에 저장되어 있다가 위의 입력명령을 만나면 `name`에 저장되어 엉뚱한 결과가 출력된다.

문자형 자료에서 위와 같은 특수문자를 입력구분자로 처리하기 위하여 “%c” 앞에 공백을 넣어준 것이며 이렇게 하면 버퍼에 저장된 특수문자를 무시하고 새로운 문자를 입력으로 받게 되는 것이다.

- 실습 과제 2
 - 이전 소스코드를 활용하여 사람 소개 프로그램을 개발하기
 - 영문이름 세글자 입력받아 출력
 - 정수형 정보 입력 받기: 나이, 학번, 전화번호 뒷자리
 - 실수형 정보 입력 받기: 몸무게, 시력, 키
 - 정수형 및 실수형 정보 5개 더 받기 (스스로 정의함)
 - 전체 정보를 이용해서 입력 받은 사람 정보 출력하기

Thank You!!!

