C언어_ 기초

&

변수 이름 앞에 &를 붙이면 그 변수의 주소를 나타냄

상수

숫자, 문자, 논리와 같은 값 자체

변수

다른값을 저장할 수 있는 이름을 변수(variable)라고 함

변수의 선언 : 처음 변수를 지정하는 것

```
변수는 값을 저장하기 위한 메모리의 일부 공간
변수로 지정된 이름은 특정 메모리 공간을 가리키며, 이 공간을 효율적으로 사용할 수 있도록 자료형으로 선언

메모리 - 컴퓨터는 변수의 각종 데이터들을 컴퓨터의 메모리, 즉 RAM 이라는 특별한 기억공간에 이를 기록
```

자료형

- 저장되는 데이터의 종류에 따른 형태로, 저장되는 값의 종류와 범위에 따라 다르게 표현
    1. int : 정수형 변수
    2. float : 실수형 변수
    3. char : 문자형 변수
- 자료형에 따라 저장될 수 있는 값의 종류와 범위가 결정되어 있습니다.

```
자료형의 크기와 범위
1. 정수형 변수
   char(문자)     | 1byte | [ -128 ~ +128 ]
   int            | 4byte | [ -2147483648 ~ +2147483647]
   unsigned int   | 4byte | [0 ~ 4294967295 ]
   long long int | 8byte | [ - 매우큼 ~~ + 매우큼]
   short          | 2byte | [ -32768 ~ +32767 ]
2. 실수형 변수
   float          | 4byte | [ - 매우작음 ~~ + 매우큼]
   double         | 8byte | [ - 매우매우작음 ~~ +매우매우큼]
```

아스키코드

- 미국표준협회에서 개발한 것으로, 알파벳과 아라비아 숫자, 그리고 특수문자를 표현하는 2진수 코드 체계
- 숫자로 128개의 문자를 표현하여 메모리에 저장될때는 0~127 사이로 변환되어 저장

```
ex)  char a = 'A'; 를 하면 a의 메모리에는 'A'의 아스키코드인 65가 실제로 변환되어 저장됨

그럼 char b = a + 1; 를 했을때 어떤게 출력이 될까요 ??
```

문자열형 변수

문자형 변수와 유사하지만 저장과 계산의 효율성을 위해서 쓰임

```
#include <stdio.h>
main ()
{
    char a[5] = { 'K', 'O', 'R', 'E', 'A',};
    printf ( "%c\n", a[1]);
    printf ( "%s\n", a);
    char b[6] = { 'K', 'O', 'R', 'E', 'A', '\0'};
    printf ( "%s\n", b);
    char c[6] = "KOREA";
    printf ( "%s\n", c);
}
```

이게 실행되면 어떤 결과가 나올까요 ?

```
O                               -> a[1]을 뽑아서
KOREA"                          -> 끝에 왜 " 이 나올까요 ?? 끝을 알수 없기 때문에 다른문
자도 출력
KOREA                           -> 마지막에 \0로 끝을 알려주는게 있어 정확하게 출력
KOREA                           -> 입력의 불편함을 막기위해 그냥 저렇게 써도 됨
```

문자열 변수는 여러 문자를 저장하기 위해서 여러 개의 메모리 공간을 연속적으로 할당받아 한 문자씩 저장

여기서 java랑 다른게 존재

```
KOREA를 저장할때
  java는 String str = "KOREA";
  c언어는 char str[6] = "KOREA";
크기가 c언어가 1개 더 많음 그이유는 무엇일까?

문자열 상수길이보다 하나 큰 공간을 할당해야한다. 그 이유는 종료문자가 포함될 수 있도록
```

논리변수

'True', 'False'의 논리 상수값을 저장하는 변수를 사용

java에서는 boolean 타입을 지원하지만 c언어에서는 boolean을 사용하기 위해

```
#include <stdbool.h>
```

를 선언 해줘야 사용이 가능하다.

변수명 지정

변수의 이름을 만들 때에는, 예약어(keyword), 식별자(identifier), 상수 등과 구별하여 인식할 수 있도록 지정

1. 변수의 이름은 알파벳, 아라비아 숫자, 특수기호 '_'의 조합으로 가능
2. 변수 이름은 알파벳 대소문자를 구분하며, 첫글자에는 **숫자 불가능**
3. 시스템 예약어(if, for...)는 불가능

제어문자

1. \a 경고음
2. \b 백스페이스
3. \f 폼피드 : 위치가 다음 페이지의 시작 부분으로 넘겨짐
4. \n 줄바꿈
5. \r 캐리지 리턴 : 현재 위치를 나타내는 커서를 맨 앞으로 이동시킨다.
6. \t 수평 탭
7. \v 수직 탭
8. \\ 백슬래시(\)
9. \' 작은 따음표
10. \" 큰 따음표

서식 지정자의 종류

서식 지정자	출력 형태	서식 지정자	출력 형태
%c	단일 문자	%x	부호없는 16진정수(소문자)
%d	부호 있는 10진 정수	%X	부호없는 16진정수(대문자)
%i	%d와 같음	%e	e표기법에 의한 실수
%f	부호 있는 10진 실수	%E	E표기법에 의한 실수
%s	문자열	%g	값에 따라서%f,%e중 하나를 선택
%o	부호 없는 8진 정수	%G	값에 따라서%f,%e중 하나를 선택
%u	부호 없는 10진 정수	%%	%기호 출력

서식 지정자의 쓰임

서식 지정자	출력대상	출력형태
%d	char,short,int	부호 있는 10진 정수
%ld	long	부호 있는 10진 정수
%lld	long long	부호 있는 10진 정수
%u	unsigned int	부호 없는 10진 정수
%o	unsigned int	부호 없는 8진 정수
%x, %X	float,double	부호 없는 16진 정수
%f	float	10진수 방식의 부동소수점 실수
%lf	long double,double	10진수 방식의 부동소수점 실수
%c	char,short,int	값에 대응하는 문자
%s	char*(문자열)	문자열
%p	void(주소값)	포인터 주소 값

```
#include <stdio.h>
main ()
{
    int a = 365;
    float b = 3.14;
    printf ( "%5d \t", a);
    printf ( "%3.1f \n", b);
}
```

```
  365     3.1
```

"%d" 랑 "%5d"의 차이는?

- 정수형 서식 문자 앞에 붙는 숫자 (5)는 오른쪽 정렬을 위한 숫자
- 즉 5개의 칸을 만들어 놓고 오른쪽으로 정렬하여 출력

"%f" 랑 "%3.1f"의 차이는?

- 정수 자리의 3은 오른쪽 정렬 숫자
- 소수 자리의 1은 소수 둘째 자리에서 반올림하여 첫째 짜리까지만 출력

---

scanf ()

```
#include <stdio.h>
main ()
{
    int a = 0;
    scanf ( "%d", &a);
    printf ( "%d", a);
}
```

입력될 변수의 이름 앞에 &를 붙여야한다. &은 뭘까요 ?