

2장 자료형과 입출력

한림대학교 소프트웨어융합대학 양은샘.



2장 자료형과 입출력

- 안녕하세요? 여러분!
- 이번 장에서는 C의 자료형과 입출력 방법을 학습하도록 하겠습니다.
- C 프로그램에서는 주어진 자료들을 이용해 원하는 결과를 얻는 작업을 하게 됩니다.
- 그렇기 때문에 C 프로그램에서 사용하는 자료의 형태에는 어떤것들이 있는지,
- 처리는 어떻게 하는지에 대해 알아 둘 필요가 있습니다.
- 지난 시간에 학습한 내용을 리뷰한 후 학습을 시작하도록 하겠습니다.

학습 목차

2.1 자료형

2.2 변수(variable)

2.3 상수(constant)

2.4 출력 함수 printf()

2.5 입력 함수 scanf(), scanf_s()

2.6 식에서의 형변환

- ☐ 개념 확인 학습
- ☐ 적용 확인 학습
- ☐ 응용 프로그래밍

학습 목표

- C 프로그램에서 사용하는 자료형의 종류를 안다.
 - 변수(variable) 및 상수(constant)의 사용법을 안다.
 - printf()를 이용하여 화면에 출력할 수 있다.
 - scanf()를 사용하여 사용자 입력을 받을 수 있다.
 - 식에서의 형변환 과정을 안다.
-
- 개념 확인 학습으로 배운 내용을 정리한다.
 - 적용 확인 학습으로 개념 습득 여부를 확인한다.
 - 응용 프로그래밍으로 문제해결력을 키운다.

C 언어에서 제공하는 자료형

구 분	의미	표현 방법	메모리 크기 / 데이터 표현 범위
기본형	문자형	char	1byte = sizeof(char) / $-2^7 \sim 2^7 - 1$
		unsigned char	1byte = sizeof(unsigned char) / $0 \sim 2^8$
	정수형	short	2bytes = sizeof(short) / $-2^{15} \sim 2^{15} - 1$
		int	4bytes = sizeof(int) / $-2^{31} \sim 2^{31} - 1$
		long	4bytes = sizeof(long) / $-2^{31} \sim 2^{31} - 1$
		unsigned short	2bytes = sizeof(unsigned short) / $0 \sim 2^{16} - 1$
		unsigned int	4bytes = sizeof(unsigned int) / $0 \sim 2^{32} - 1$
		unsigned long	4bytes = sizeof(unsigned long) / $0 \sim 2^{32} - 1$
	실수형	float	4bytes = sizeof(float)
		double	8bytes = sizeof(double)
		long double	8bytes = sizeof(long double)
		소수점 이하 몇 자리를 사용하느냐에 따라 표현 범위가 다름.	
	형이 없음	void	

구 분	의미	표현 방법	메모리 크기 / 데이터 표현 범위
열거형		enum	내부적으로 정수형
집합형	배열	[]	
유도된 자료형	구조체	struct	
	공용체	union	
포인터형	포인터	*	

char의 데이터 표현 범위

⇒ char의 데이터 표현 범위

b7	b6	b5	b4	b3	b2	b1	b0
부호 및 크기 (1bit)	크기 (7bits)						

- $1000\ 0000_{(2)} = -2^7 = -128$
- $1000\ 0001_{(2)} = -2^7 + 2^0 = -128 + 1 = -127$
- $1111\ 1111_{(2)} = -2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 = -1$
- $0000\ 0000_{(2)} = 0$
- $0111\ 1111_{(2)} = +2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 = 127$

⇒ unsigned char의 데이터 표현 범위

b7	b6	b5	b4	b3	b2	b1	b0
크기 (8bits)							

- $0000\ 0000_{(2)} = 0$
- $1111\ 1111_{(2)} = +2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 = 255$

변수 이름 규칙

- 변수 이름은 알파벳 문자, 숫자 그리고 언더스코어 문자('_')로 이루어진다.
- 변수 이름은 반드시 영문자나 언더스코어 문자('_')로 시작되어야 한다.
- 대소문자는 서로 다른 이름으로 취급되므로 구별해야 한다.
- C에서 사용되는 키워드(Keyword)는 변수 이름으로 사용할 수 없다.
 - 제어문 관련 : for, do, while, if, else, switch, case, default, break, continue, goto, return.
 - 데이터형 관련 : char, short, int, float, double, long, unsigned, void, struct, union, enum, typedef.
 - 기억장소관련 : auto, extern, register, static.
 - 기타 : sizeof() 등 함수명들.
- 변수 이름을 정할 때는 위의 규칙에 따라 정하되, 변수가 사용되는 목적을 알 수 있는 단어로 선택하는 것이 바람직하다.

변수 선언의 예

- 변수 선언의 예

char c, ch;	//변수 c, ch를 문자형으로 선언
unsigned char _ch;	//변수 _ch를 양의 부호 문자형으로 선언
int m_m;	//변수 m_m를 정수형으로 선언
float f_data;	//변수 f_data를 4bytes 실수형으로 선언
double x, y, z;	//변수 x, y, z를 8bytes 실수형으로 선언

- 잘못된 변수 선언의 예

```
int 2number;  
float f&data;  
char case;
```


상수의 표현

구분		표현 예
숫자 상수	정수형 상수	<pre>int angel = 1004; //10진수 int octal = 034; //8진수, 0으로 시작하는 8진 숫자열 int hexa = 0xFF; //16진수, 0x로 시작하는 16진 숫자열 long lo = 22L; //long형, 10, 8, 16진 숫자 뒤에 'L' 표기</pre>
	실수형 상수	<pre>float f1 = 3.14; //실수 형태 double e1 = 0.14E-3; //지수 형태 double e2 = 1.5E+2; //지수 형태</pre>
문자 상수	문자 상수	<pre>//문자 상수는 내부적으로 ASCII 코드 값으로 변환 됨 char alpha = 'A'; //알파벳 문자 char alpha_d = 65; //'A'의 10진수 ASCII 코드 값=65 char alpha_h = 0x41; //'A'의 16진수 ASCII 코드 값=0x41 char one = '1' //숫자 문자, '1'의 10진수 ASCII 코드 값=49 char enter = '\n'; //제어 문자</pre>
	문자열 상수	<pre>char fruit[] = "apple"; //문자열 char job[] = "computer programmer"</pre>
기호 상수		<pre>#define ESC 27 #define MAX 100</pre>

출력 함수 printf()

➡ `printf("출력 형식 지정 문자열", 인수1, 인수2, ...);`

- “출력 형식 지정 문자열” : 여러 가지 제어문자 포함 가능
- 인수1, 인수2, ... : 출력될 항목(변수, 상수, 특정 값, 계산식 등)

➡ `printf()` 사용 예

`printf("%d", 123);` //123을 정수로 출력

`printf("%s", "123");` //123을 문자열로 출력

`printf("%c %d\n", 'A', 35);` //'A', 35를 문자와 정수로 출력 후 바꿈

`printf("max=%d, min=%d\n", 200, 50);` //max=200, min=50 출력 후 줄 바꿈

제어 문자의 종류 및 기능

제어문자	기능
Wn	다음 행의 처음으로 간다. (new line)
Wa	벨 소리를 낸다.(alert)
Wb	커서를 한 문자 뒤로 이동시킨다. (backspace)
Wf	프린터의 페이지를 넘긴다. (form feed)
Wr	커서를 현재행의 처음으로 이동시킨다. (carriage return)
Wt	다음 탭 위치로 커서를 이동시킨다. (horizontal tab)
WW	역슬래시를 출력한다.
W'	' 를 출력한다.
W"	" 를 출력한다.
W%	%를 출력한다.

변환 문자의 종류 및 기능

제어문자	변환문자	기능
정수형	%d	10진 int 정수형 출력
	%u	10진 unsigned int 정수형 출력
	%x	16진 int 정수형 출력
	%o	8진 int 정수형 출력
문자형	%c	하나의 문자 출력
문자열형	%s	문자열 출력
실수형	%f	float 실수형으로 출력
	%lf	double 실수형으로 출력
	%e	실수형을 지수 표기법으로 출력
	%g	%e와 %f의 적당한 형태로 출력

수정자의 종류 및 기능

수정자	예	기능
flag	-	왼쪽 정렬로 출력하고 빈 영역에 공백을 표시한다.
	+	+ 부호를 함께 나타낸다.
	#	8진수는 맨 앞에 0을, 16진수는 0x를 붙인다.
digit(s)	%5.3f	최소 필드 폭을 나타낸다. 오른쪽 정렬로 출력한다.
	%10s	

➡ printf() 수정자(Modifier) 사용 예

- ① `printf("%10d", 123);` //10자리 공간에 오른쪽 정렬로 출력
- ② `printf("%+10d", 123);` //10자리 공간, '+' 부호와 함께 출력
- ③ `printf("%-10d", 123);` //10자리 공간, 왼쪽 정렬
- ④ `printf("%010d", 123);` //10자리 공간, 빈 공간은 0으로 채움, 오른쪽 정렬
- ⑤ `printf("%ld", 1234567);` //long형 값으로 변환
- ⑥ `printf("%5.2f", 123.456);` //다섯 자리 공간, 소수점 이하 두 자리
- ⑦ `printf("%10.4f", 123.456);` //10자리 공간, 소수점 이하 네 자리

	1	2	3	4	5	6	7	8	9	10
①								1	2	3
②							+	1	2	3
③	1	2	3							
④	0	0	0	0	0	0	0	1	2	3
⑤	1	2	3	4	5	6	7			
⑥	1	2	3	.	4	6				
⑦	1	2	3	.	4	5	6	0		

printf() 응용

[예제 2.1] printf() 응용

```
#include <stdio.h>
int main()
{
    int in = 1234;
    double dou = 1.23456789;
    char ch = 'a';
    char str[] = "apple";

    printf("%6dWn", in);
    printf("%06dWn", in);
    printf("%uWn", in);
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS C:\lecture_src\ctest> cd "c:\lecture_
1234
001234
1234
1234
1234
1.234568
1.235
1.234568
a
a
apple
apple
PS C:\lecture_src\ctest>
```

```
printf("%dWn", in);
printf("%3dWn", in); //데이터보다 작은 자리수이면 원래 길이대로 출력
```

```
printf("%lfWn", dou);
printf("%7.3lfWn", dou);
printf("%12.6lfWn", dou);
```

```
printf("%cWn", ch);
printf("%3cWn", ch);
```

```
printf("%sWn", str);
printf("%10sWn", str);
return 0;
```

```
}
```

입력 함수 scanf()

➡ scanf(), scanf_s() 형식

`scanf("입력 형식 지정 문자열", 인수1주소, 인수2주소, ...);`

`scanf_s("입력 형식 지정 문자열", 인수1주소, sizeof(인수1), ...);`

➡ scanf(), scanf_s() 형식 사용 예

`int int_variable;`

`char char_variable;`

`scanf("%d", &int_variable);`

`scanf("%c", &char_variable);`

scanf() 응용

[예제 2.2] scanf() 응용

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int in;
    char ch;
    double dou;

    printf("input int_variable : ");
    scanf("%d", &in);

    while (getchar() != '\n'); //두 번째 입력 전, 버퍼에 저장되어 있는 값을 삭제

    printf("input char_variable : ");
    scanf("%c", &ch);

    printf("input double_variable : ");
    scanf("%lf", &dou);

    printf("\nin=%d, ch=%c, dou=%.2lf\n", in, ch, dou);
    return 0;
}
```

PROBLEMS	OUTPUT	디버그 콘솔	터미널
	input int_variable : 23 input char_variable : A input double_variable : 23.45678 in=23, ch=A, dou=23.46 PS C:\lecture_src\ctest>		

여러 가지 문자 입력 함수

사용법	라이브러리	버퍼이용여부	문자입력작동	echo 기능
<code>scanf("%c", &ch);</code>	<code><stdio.h></code>	O	문자키+〈enter〉를 누르면 입력 작동	문자키를 누르면 화면에 바로 표시
<code>ch = getchar();</code>	<code><stdio.h></code>	O	문자키+〈enter〉를 누르면 입력 작동	문자키를 누르면 화면에 바로 표시
<code>ch = getche();</code>	<code><conio.h></code>	X	문자키를 누르면 바로 입력 작동	문자키를 누르면 화면에 바로 표시
<code>ch = getch();</code>	<code><conio.h></code>	X	문자키를 누르면 바로 입력 작동	문자키 표시 안 됨

여러 가지 문자 입력 함수 사용

```
#include <stdio.h>
#include <conio.h>
int main()
{
    char ch;

    printf("scanf() 문자 입력 : ");
    scanf("%c", &ch);
    printf(" => ch=%c\n", ch);
```

while (getchar() != 'Wn'); //두 번째 입력 전, 버퍼에 저장되어 있는 값을 삭제

```
printf("getchar() 문자 입력 : ");
ch = getchar();
printf(" => ch=%c\n", ch);
```

```
printf("getche() 문자 입력 : ");
ch = getche();
printf(" => ch=%c\n", ch);
```

```
printf("getch() 문자 입력 : ");
ch = getch();
printf(" => ch=%c\n", ch);
return 0;
```

```
}
```

scanf() 문자 입력 : a => ch=a getchar() 문자 입력 : a => ch=a getche() 문자 입력 : a => ch=a getch() 문자 입력 : => ch=a

식에서의 형변환

```
#include <stdio.h>
int main()
{
    int ina=1, inb=2;
    double dou=2.0;
    double result_da, result_db;

    result_da = (ina + inb) / inb; // 결과는?
    result_db = (ina + inb) / dou; // 결과는?

    printf("result_da=%.2lf\n", result_da);
    printf("result_db=%.2lf\n", result_db);

    return 0;
}
```

result_da=1.00 result_db=1.50

개념 확인학습 & 적용 확인학습 & 응용 프로그래밍

- 다음 파일에 있는 문제들의 해답을 스스로 작성 해 보신 후 개념 & 적용 확인 학습 영상을 학습 하시기 바랍니다.
 - c_02장_자료형과 입출력_ex.pdf
- 퀴즈와 과제가 출제되었습니다.
 - 개념 & 적용 확인 학습 영상을 학습하신 후 과제와 퀴즈를 수행 하시기 바랍니다.

Q & A

- “자료형과 입출력”에 대한 학습이 모두 끝났습니다.
- 모든 내용을 이해 하셨나요?
- 아직 이해가 안되는 내용이 있다면 다시 한번 복습하시기 바랍니다.
- 질문은 한림 SmartLEAD 쪽지 또는 e-mail 또는 전화상담을 이용하시기 바랍니다.



- 퀴즈와 과제가 출제되었습니다. 마감시간에 늦지 않도록 주의해 주세요.
- 다음 시간에는 “연산자”에 대해 알아보겠습니다.
- 수고하셨습니다.^^