

Chapter 4. Constant and Basic Data Types

April, 2016
Seungjae Baek

Dept. of software
Dankook University

<http://embedded.dankook.ac.kr/~baeksj>

- 기본 자료형
- 문자 표현 방식과 문자 자료형
- 상수
- 자료형 변환

■ 변수 (Variables)

- ✓ 이름이 부여된 기억 장소
- ✓ 변수는 자료형(type, 유형)을 가지고 있음

종류	설명
기본 타입 (Basic types)	a. 정수 타입 (integer types) b. 실수 타입 (floating-point types)
열거형 타입 (Enumerated types)	정수형의 상수 정의, 프로그램의 이해를 쉽게 해줌
void 타입 (The type void)	값을 지정하지 않음
사용자 정의 타입 (Derived types)	a. 포인터 (pointer types) b. 배열 (array types) c. 구조체 (structure types) d. 공용 (union types) e. 함수 (function types)

■ 기본 자료형의 종류

자료형	크기	값의 표현 범위
정수형	char	1 byte -128 to 127 or 0 to 255
	short	2 bytes -32,768 to 32,767
	int	4 bytes -2,147,483,648 to 2,147,483,647
	long	4 bytes -2,147,483,648 to 2,147,483,647
	long long	8 bytes -9,233,372,036,854,775,808 to +9,233,372,036,854,775,807
실수형	float	4 bytes 1.2E-38 to 3.4E+38
	double	8 bytes 2.3E-308 to 1.7E+308
	long double	8 bytes 이상 3.4E-4932 to 1.1E+4932

☞ 자료형의 크기는 시스템 별로 다름

☞ 정수형에는 **unsigned** 를 사용하여 **0** 과 정수만 표현 가능

■ sizeof 연산자

- ✓ Queries size of the object or type

```
sizeof (types);  
or sizeof expression;
```

사용법

```
int main(void)  
{  
    char ch=9;  
    int num=100;  
    float pipi=3.14;  
  
    printf("size of char : %d \n", sizeof(ch));  
  
    printf("size of int : %d \n", sizeof(inum));  
    printf("size of int : %d \n", sizeof inum);  
    printf("size of int : %d \n", sizeof(int));  
  
    printf("size of float : %d \n", sizeof float);  
    ...  
}
```

■ 자료형의 크기 예

```
#include <stdio.h>
int main(void)
{
    char ch=9;
    int inum=1052;
    double dnum=3.1415;
    printf("변수 ch의 크기: %d \n", sizeof(ch));
    printf("변수 inum의 크기: %d \n", sizeof(inum));
    printf("변수 dnum의 크기: %d \n", sizeof(dnum));
    printf("char의 크기: %d \n", sizeof(char));
    printf("int의 크기: %d \n", sizeof(int));
    printf("long의 크기: %d \n", sizeof(long));
    printf("long long의 크기: %d \n", sizeof(long long));
    printf("float의 크기: %d \n", sizeof(float));
    printf("double의 크기: %d \n", sizeof(double));
    return 0;
}
```

Results

```
변수 ch의 크기: 1
변수 inum의 크기: 4
변수 dnum의 크기: 8
char의 크기: 1
int의 크기: 4
long의 크기: 4
long long의 크기: 8
float의 크기: 4
double의 크기: 8
```

Q: 많은 자료형 중에 어떤 자료형을 선택해야 하는가?

■ 정수 표현을 위한 일반적인 자료형 선택

```
#include <stdio.h>
int main(void)
{
    char ch1=1, ch2=2, result1=0;
    short sh1=300, sh2=400, result2=0;

    printf("size of ch1 & ch2: %d, %d \n", sizeof(ch1), sizeof(ch2));
    printf("size of sh1 & sh2: %d, %d \n", sizeof(sh1), sizeof(sh2));
    printf("size of char add: %d \n", sizeof(ch1+ch2));
    printf("size of short add: %d \n", sizeof(sh1+sh2));

    result1=ch1+ch2;
    result2=sh1+sh2;

    printf("size of result1 & result2: %d, %d \n", sizeof(result1),
        sizeof(result2));

    return 0;
}
```

- 정수 표현을 위한 일반적인 자료형 선택

Results

```
size of ch1 & ch2: 1, 1  
size of sh1 & sh2: 2, 2  
size of char add: 4  
size of short add: 4  
size of result1 & result2: 1, 2
```

`sizeof(ch1+ch2)`
`sizeof(sh1+sh2)`

Q: 왜 **char** 크기와 **short** 크기가 **4** 바이트 인가 ?

A: CPU의 기본 연산 단위는 **int** (레지스터와 관련)

- 실수 표현을 위한 일반적인 자료형 선택
 - ✓ 값의 표현 범위 보다 중요한 것은 “정밀도”

실수 자료형	소수점 이하 정밀도	바이트 수
float	6 자리	4
double	15 자리	8
long double	18 자리	12 (8)

■ 실수 표현을 위한 일반적인 자료형 선택

```
#include <stdio.h>
int main(void)
{
    double rad;
    double area;
    printf("원의 반지름 입력: ");
    scanf("%lf", &rad);

    area = rad*rad*3.1415;
    printf("원의 넓이: %f \n", area);
    return 0;
}
```

%lf : double 입력 서식
%f : double 출력 서식

Results

```
원의 반지름 입력: 2.5
원의 넓이: 19.634375
```

■ ASCII 코드

- ✓ 미국 표준 협회 (ANSI) 에서 아스키 (ASCII) 코드를 제정
 - ANSI: American National Standards Institute
 - ASCII: American Standards Code for Information Interchange
- ✓ 총 128개의 문자로 구성

10진수	16진수	값
65	41	A
66	42	B
67	43	C
68	44	D
...
97	61	a
98	62	b
99	63	c

<ASCII table>

Q: A 와 a의 10진수 값의 차이는?

■ C 에서 문자 표현

문자 표현 1

```
int main()
{
    char ch1 = 'A';
    char ch2 = 'C';
    ...
}
```

문자 표현 2

```
int main()
{
    char ch1 = 65;
    char ch2 = 67;
    ...
}
```

char ch1

65

char ch2

67

☞ 메모리에는 문자가 아니라 숫자가 저장됨

<물리 메모리>

■ 문자 인식의 예

Q1: ch3 = 321 이면 출력되는 값은 ?

Q2: "Hello" 와 같은 문자열을 저장하고 싶으면?

```
#include <stdio.h>

int main(void)
{
    char ch1='A', ch2=65;
    int ch3='Z', ch4=90;

    printf("%c %d \n", ch1, ch1);
    printf("%c %d \n", ch2, ch2);
    printf("%c %d \n", ch3, ch3);
    printf("%c %d \n", ch4, ch4);
    return 0;
}
```

Results

```
A 65
A 65
Z 90
Z 90
```

%d : 정수값 출력
%c : 문자 출력

- 상수(常數, **constants**)의 정의
 - ✓ 상수는 값이 변하지 않는 데이터
- 상수의 종류
 - ✓ 이름이 없는 상수
 - **literal** 상수
 - ✓ 이름이 있는 심볼릭 (**symbolic**) 상수
 - **const** 상수

■ 이름이 없는 상수 (literal)

상수의 예

```
int main()
{
    int num = 10 + 20;
    ...
}
```



10

20

int num

30

type?

10

type?

20

<물리 메모리>

Q: type의 변수 자료형은 무엇일까요?

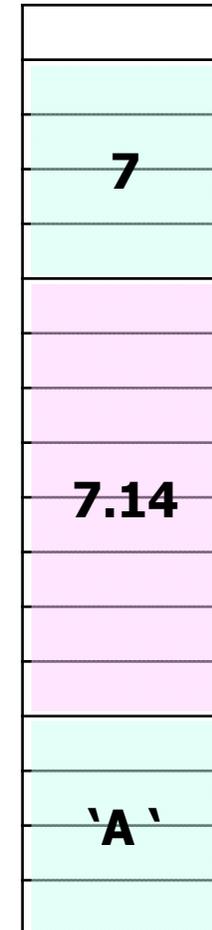
■ 상수의 자료형

변수의 예

```
#include <stdio.h>
int main(void)
{
    printf("literal int size: %d \n", sizeof(7));
    printf("literal double size: %d \n", sizeof(7.14));
    printf("literal char size: %d \n", sizeof('A'));
    return 0;
}
```

Results

```
literal int size: 4
literal double size: 8
literal char size: 4
```



Q: char의 변수 자료형은 **4byte** 일까요?

<물리 메모리>

- 접미사를 이용한 다양한 상수 표현
 - ✓ int 이외의 다양한 정수 표현
 - ✓ double 이외의 실수 표현

상수의 예

```
#include <stdio.h>
int main(void)
{
    float f1 = 5.678;
    float f2 = 3.23 + 5.12;
    return 0;
}
```

상수 값 뒤에 **f** 또는 **F** 추가
예) float f1 = 5.678f;

■ 접미사를 이용한 다양한 상수 표현

- ✓ 정수 표현을 위한 접미사

접미사	자료형	사용 예
U	unsigned int	unsigned int k = 1025U;
L	long	long k = 2467L
UL	unsigned long	unsigned long k = 3456UL
LL	long long	long long k = 5768LL
ULL	unsigned long long	unsigned long long k = 8979ULL

- ✓ 실수 표현을 위한 접미사

접미사	자료형	사용 예
F	float	float k = 1025F;
L	long double	long double k = 2467L

- 이름 있는 심볼릭 상수 : `const` 상수
 - ✓ 1) `const` 키워드 사용
 - 선언과 동시에 값을 초기화 해야 함
 - ✓ 2) 매크로 (`macro`)를 이용하는 방법

상수의 예

```
#include <stdio.h>
int main(void)
{
    const int MAX = 100;
    const double PI = 3.1415;
    MAX = 350;
    return 0;
}
```

error C2166: l-value specifies const object

Tip: 상수 이름 만들기

1. 상수 이름은 대문자로 표시

2. 둘 이상의 단어는 '_' 로 연결 (예, HOUSE_COUNT)

- 자료형 변환의 정의
 - ✓ 데이터의 표현을 바꾸는 것
 - ✓ 예) `char` 데이터를 `int` 데이터로 변환

- 자료형 변환의 종류
 - ✓ 1) 자동 형 변환 (묵시적 형 변환)
 - ✓ 2) 강제 형 변환 (명시적 형 변환)

■ 자동 형 변환 - 1

✓ : 대입 연산 과정에서 발생

형 변환 예 (1)

```
#include <stdio.h>
int main(void)
{
    int n1 = 3.1415;
    double n2 = 123;
    return 0;
}
```

test.c(4): warning C4244: 'initializing' : conversion from 'double' to 'int', possible loss of data

Q: Correct or not ?

형 변환 예 (2)

```
#include <stdio.h>
int main(void)
{
    int n3 = 129;
    char c1 = n3;
    return 0;
}
```

Q: Correct or not ?

■ 자동 형 변환 예

```
#include <stdio.h>
int main(void)
{
    double num1=245;
    int num2=3.1415;
    int num3=129;
    char ch=num3;

    printf("정수 245를 실수로: %f \n", num1);
    printf("실수 3.1415를 정수로: %d \n", num2);
    printf("큰 정수 129를 작은 정수로: %d \n", ch);
    return 0;
}
```

Results

```
정수 245를 실수로: 245.000000
실수 3.1415를 정수로: 3
큰 정수 129를 작은 정수로: -127
```

- 자동 형 변환 - 2
 - ✓ 정수의 승격 (Integral Promotion)

```
#include <stdio.h>
int main(void)
{
    short n1=15, n2=15;
    short n3=n1+n2;
    return 0;
}
```

정수의 승격, 대입 연산과정

Q: 정수의 승격이 의미가 있는가?

■ 자동 형 변환 - 3

- ✓ 피 연산자의 자료형 불일치로 발생

```
#include <stdio.h>
int main(void)
{
    double n1=5.15+19;
    return 0;
}
```

계산 가능?

- * 2가지로 계산이 가능
- 1) int로 변환
 - 2) double로 변환



<산술연산에서 형 변환 규칙>

Rule : 데이터 손실의 최소화

- 명시적 형 변환
 - ✓ 강제로 일으키는 형 변환

```
#include <stdio.h>
int main(void)
{
    int num1=3, num2=4;
    double divResult;
    divResult=(double)num1/num2;
    printf("나눗셈 결과: %f \n", divResult);
    return 0;
}
```

Type casting operator

Results

나눗셈 결과: 0.750000 (명시적 형 변환 : **O**)

나눗셈 결과: 0.000000 (명시적 형 변환 : **X**)

- C 언어의 기본 자료형 이해
 - ✓ 자료형의 종류 및 데이터의 표현 범위
 - ✓ sizeof 연산자
- 문자 표현 방식과 자료형
 - ✓ ASCII 코드
- 상수
 - ✓ 리터럴 상수와 심볼릭 상수
- 자료형의 변환

- 1) `int x;` 가 있을 때 `x`에 정수값이 얼마일 때 아래와 같은 결과가 출력되는가? 또한 이 때 정수 값과 2진수 값을 적으시오.
 - ✓ C 문장 : `printf("%c \n", x);`
 - ✓ 출력 결과 : C
 - ✓ 정수 값, 2진수 값 :