
C언어 자료형

자료형

자료형 : 데이터를 표현하는 방법

자료형		크기	값의 표현 범위
정수형	char	1 바이트	-128이상 +127이하
	short	2 바이트	-32,768이상 +32,767이하
	int	4 바이트	-2,147,483,648이상 +2,147,483,647이하
	long	4 바이트	-2,147,483,648이상 +2,147,483,647이하
	long long	8 바이트	-9,223,372,036,854,775,808 이상 +9,223,372,036,854,775,807 이하
실수형	float	4 바이트	$\pm 3.4 \times 10^{-37}$ 이상 $\pm 3.4 \times 10^{+38}$ 이하
	double	8 바이트	$\pm 1.7 \times 10^{-307}$ 이상 $\pm 1.7 \times 10^{+308}$ 이하
	long double	8 바이트 이상	double 이상의 표현범위

자료형

1. 데이터의 표현방식이 다르므로, 최소 두 이상의 자료형이 필요

-컴퓨터가 정수와 실수를 표현하는 방식이 다르기 때문

2. 메모리 공간의 적절한 사용을 위해서 다양한 크기의 자료형이 필요

예) 5,000개의 정수를 저장 - short형으로 표현 ($5,000 * 2 = 10,000$)
- int형으로 표현 ($5,000 * 4 = 20,000$)

자료형

```
1  #include <stdio.h>
2
3  int main(void)
4  {
5      char ch1 = 'A';
6      char ch2 = 'C';
7
8      ...;
9  }
```



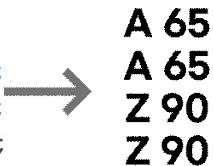
```
1  #include <stdio.h>
2
3  int main(void)
4  {
5      char ch1 = 65;
6      char ch2 = 67;
7
8      ...;
9  }
```

문자형x -> 정수형o

문자가 컴파일러에 의해 아스키코드 값으로 변환된다.

자료형

```
1  #include <stdio.h>
2
3  int main(void)
4  {
5      char ch1 = 'A', ch2 = 65;
6      int ch3 = 'Z', ch4 = 90;
7
8      printf("%c %d\n", ch1, ch1);
9      printf("%c %d\n", ch2, ch2);
10     printf("%c %d\n", ch3, ch3);
11     printf("%c %d\n", ch4, ch4);
12     return 0;
13 }
```



A 65
A 65
Z 90
Z 90

자료형

자료형 변환 = 데이터 표현방식 변경

- 자동 형 변환
- 강제 형 변환

<대입 연산 과정 중의 자동 형 변환>

1. double num = 123 ; #정수를 실수형으로 자동 형 변환

2. int num = 1.234 ; #실수를 정수형으로 자동 형 변환

3. int num = 129 ;

char ch = num ; #int형 -> char형 / 4Byte->1Byte

int형 : 00000000 00000000 00000000 10000001 = 129

char형 : 10000001 = -127

자료형

<정수 승격에 의한 자동 형 변환>

```
int main(void)
{
    short num1 = 15, num2 = 25 ;
    short num3 = num1 + num2 ;
    ...
}
```

<피연산자 자료형 불일치로 인한 자동 형 변환>

```
double num = 1.23 + 12 ;
```

int -> long -> long long -> float -> double -> long double

자료형

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     char num1 = 1, num2 = 2, result1 = 0;
6     short num3 = 300, num4 = 400, result2 = 0;
7
8     printf("size of num1 & num2 : %d, %d \n", sizeof(num1), sizeof(num2)); size of num1 & num2 : 1, 1
9     printf("size of num3 & num4 : %d, %d \n", sizeof(num3), sizeof(num4)); size of num3 & num4 : 2, 2
10
11     printf("size of char add : %d \n", sizeof(num1+num2)); size of char add : 4
12     printf("size of short add : %d \n", sizeof(num3 + num4)); size of short add : 4
13
14     result1 = num1 + num2;
15     result2 = num3 + num4;
16     printf("size of result1 & size of result2 : %d, %d \n", sizeof(result1), sizeof(result2)); size of result1 & result2 : 1, 2
17     return 0;
18 }
```

CPU가 처리하기에 가장 적합한 크기의 정수 자료형 = int
+) 실수 자료형 = double

자료형

<강제 형 변환>

```
1  #include <stdio.h>
2
3  int main(void)
4  {
5      int num1 = 3, num2 = 4;
6      double divResult;
7      divResult = num1 / num2;
8      printf("결과 : %f\n", divResult);
9      return 0;
10 }
```



```
1  #include <stdio.h>
2
3  int main(void)
4  {
5      int num1 = 3, num2 = 4;
6      double divResult;
7      divResult = (double)num1 / num2;
8      printf("결과 : %f\n", divResult);
9      return 0;
10 }
```

결과 : 0.00....

연산결과의 자료형은 피연산자의
자료형과 일치한다

End