

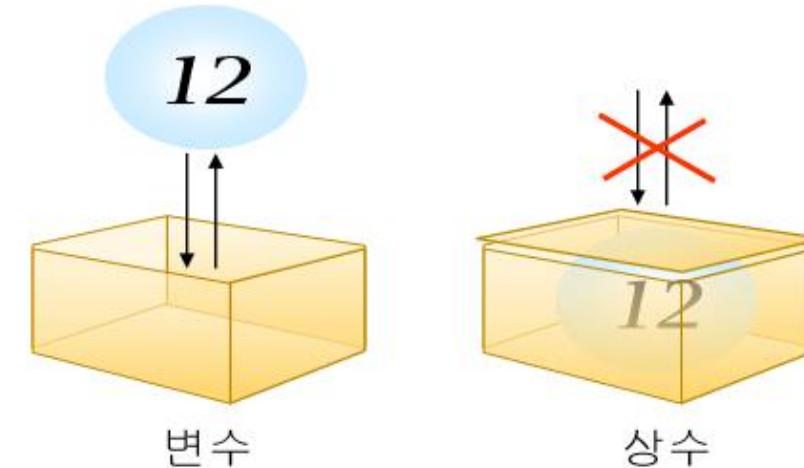
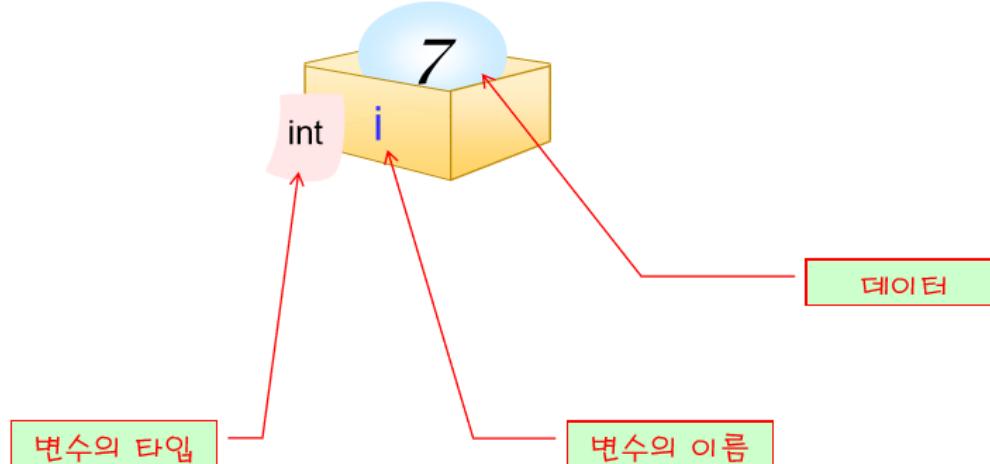


C언어 스터디 2주차

- **Printf()**함수에서 정수, 실수, 문자 각각의 형식제어문자는?
- **Scanf("%d", x);** 을 바르게 고치고 이 함수가 의미하는 바는?

# 변수와 상수

- 변수 : 메인 메모리에 생성되며 값을 언제든지 **변경이 가능**한 공간
- 상수 : 저장된 값이 **불가능**한 공간 (숫자, 문자, 문자열 모두 상수가 될 수 있음. ‘A’, “Hello”, 3.14 등등)



# 상수 (기호상수)

## 1. #define 문장 사용 ex) #define PI 3.141592

: 전처리기에 문장을 추가해서 PI라는 기호를 3.141592로 정의한다는 의미. 컴파일러가 동작하기 전에 기호를 찾아서 모두 숫자로 바꿈.  
**주의) 문장 뒤에 세미콜론이 붙지 않음!!**

## 2. const 키워드 사용 ex) const double PI = 3.141592;

: 변수 선언 앞에 **const**를 추가해주면 상수가 됨. 초기화 이후 값 **변경 불가**.

# 상수 (기호상수)

```
1 #include<stdio.h>
2 #define TAX_RATE 0.2
3
4 int main() {
5     const int MONTHS = 12;
6     int m_salary, y_salary;
7
8     printf("월급을 입력하세요 : ");
9     scanf("%d", &m_salary);
10
11    y_salary = MONTHS * m_salary;
12    printf("연봉은 %d입니다.\n", y_salary);
13    printf("세금은 %f입니다.\n", y_salary*TAX_RATE);
14
15    return 0;
16 }
```

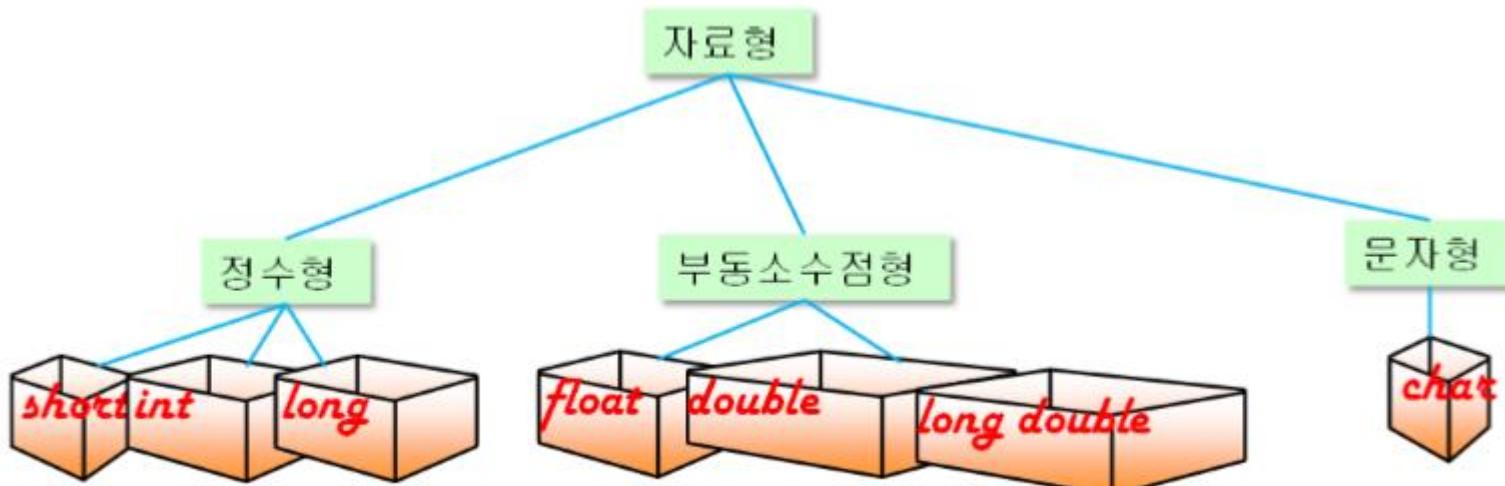
```
C:\WINDOWS\system32\cmd.exe
월급을 입력하세요 : 150
연봉은 1800입니다.
세금은 360.000000입니다.
계속하려면 아무 키나 누르십시오 . . .
```

# 변수의 자료형 – 정수, 실수

정수형 : char, short, int, long 이 있음. 3, 4, 10, 392 와 같은 정수를 담음.

실수형 : float, double, long double 이 있음. 3.1, 4.123 과 같은 실수를 담음

\*unsigned : 부호 없는 정수자료형 (unsigned int, unsigned long 등)



# 자료형

## 자료형은 왜 이렇게 많을까?

- 정수와 실수를 표현하기 위한 자료형
- 메모리 공간을 적절히 사용하기 위한 자료형

자료형		크기
정수형	char	1 바이트
	short	2 바이트
	int	4 바이트
	long	4 바이트
	long long	8 바이트
실수형	float	4 바이트
	double	8 바이트
	long double	8 바이트 이상

# 변수 선언 규칙

- 식별자는 영문과 숫자, \_(언더바)로 이루어진다.
- 식별자 중간에는 공백이 들어가면 안된다.
- 식별자의 첫 글자는 반드시 영문자 또는 밑줄 기호 \_ 이어야 한다. 또한 숫자로 시작할 수 없다.
- 대문자와 소문자는 구분된다. 즉, index, Index, INDEX는 모두 다른 변수이다.
- C언어의 키워드와 똑같은 식별자는 허용되지 않는다.

## \*키워드란?

if, for, int, return, while, else 등 C언어에서 고유한 의미를 가지고 있는 특별한 단어이다. 이를 예약어라고도 하며 사용자가 다시 정의하거나 사용하는 것이 금지되어 있다.

# 변수 선언의 예

★ short money;  
int grade;  
long average;

★ float height;  
double weight

★ char alpha;

X int 4dollar; //숫자로 시작불가!  
double return; //예약어 사용불가!  
float my height; //공백 사용불가!



정수형 변수, 정수 저장가능!

실수형 변수, 실수 저장가능!

정수형(문자형) 변수

Money = 20000;  
grade = 85;

height = 175.2;  
grade = 68.3;

alpha = 'a'



사용불가

# 변수 사용의 예

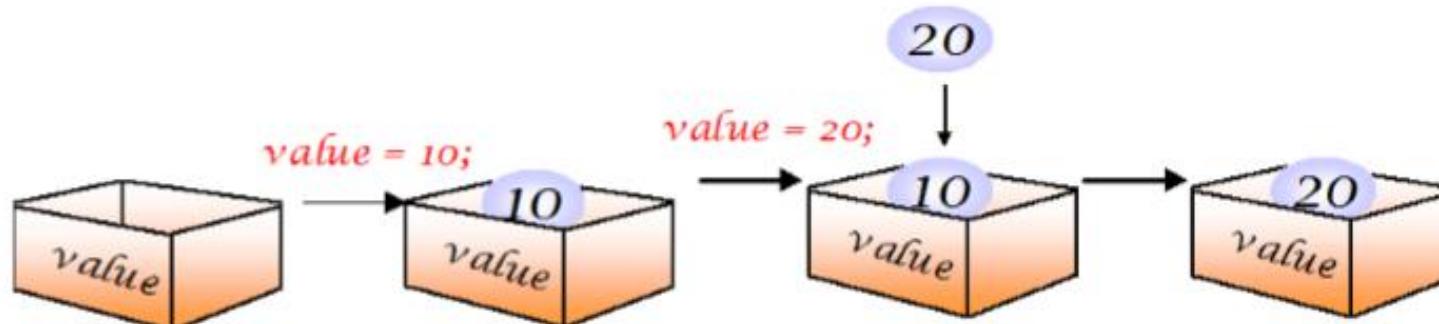
```
int value;
```

...

```
value = 10;
```

...

```
value = 20;
```



# 변수 사용의 예



**int value = 0; //**이처럼 사용하기도 함. 이를 **초기화**한다고 하며 선언과 동시에 0을 넣어준 것과 같음.

...

**value = 10;**

...

**value = 20;**

Q) 정수형 변수에 실수를 넣으면 어떻게 될까?

A) 소수점 뒤로는 버려지고 정수부분만 저장

Q) 변수를 선언하고 초기화 하지 않는다면 무슨 값이 들어있을까?

A) 지역변수는 알 수 없는 값이 들어가 있으며, 이를 흔히 쓰레기값 (garbage value)라고 부름.

# 자료형 char

Dec	Hex	문자	Dec	Hex	문자												
40	28	(	60	3C	<	80	50	P	100	64	d	120	78	x			
41	29	)	61	3D	=	81	51	Q	101	65	e	121	79	y			
42	2A	*	62	3E	>	82	52	R	102	66	f	122	7A	z			
43	2B	+	63	3F	?	83	53	S	103	67	g	123	7B	{			
44	2C	,	64	40	@	84	54	T	104	68	h	124	7C				
45	2D	-	65	41	A	85	55	U	105	69	i	125	7D	}			
46	2E	.	66	42	B	86	56	V	106	6A	j	126	7E	~			
47	2F	/	67	43	C	87	57	W	107	6B	k	127	7F	DEL			
48	30	0	68	44	D	88	58	X	108	6C	l						
49	31	1	69	45	E	89	59	Y	109	6D	m						
50	32	2	70	46	F	90	5A	Z	110	6E	n						
51	33	3	71	47	G	91	5B	[	111	6F	o						
52	34	4	72	48	H	92	5C	\	112	70	p						
53	35	5	73	49	I	93	5D	]	113	71	q						
54	36	6	74	4A	J	94	5E	^	114	72	r						
55	37	7	75	4B	K	95	5F	_	115	73	s						
56	38	8	76	4C	L	96	60	'	116	74	t						
57	39	9	77	4D	M	97	61	a	117	75	u						
58	3A	:	78	4E	N	98	62	b	118	76	v						
59	3B	;	79	4F	O	99	63	c	119	77	w						

<ASCII 코드 표>

- 원래는 정수형 자료형이지만 문자를 나타내는데 쓰임
- 문자를 나타내는 것은 컴퓨터에선 숫자로 저장됨
- 따라서 공통적인 규격이 필요한데 이가 **ASCII코드**

# 자료형 char

```
1 #include<stdio.h>
2
3 int main() {
4     char code1 = 65;
5     char code2 = 'A';
6
7     printf("%c %d\n", code1, code1);
8     printf("%c %d\n\n", code2, code2);
9
10    printf("%c %c %c\n", code1, code1 + 1, code1 + 2);
11    printf("%d %d %d\n", code1, code1 + 1, code1 + 2);
12
13    return 0;
14 }
```

```
C:\WINDOWS\system32\cmd.exe
65 A
65 A
계속하려면 아무 키나 누르십시오 . . .
```

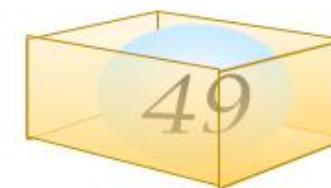
%c : 문자로 출력하는 형식제어문자  
%d : 정수로 출력하는 형식제어문자

# 중간 퀴즈



- 정수형 4가지와 실수형 3가지?

# 중간 퀴즈



# 실습 문제

- int형 변수 x와 y의 값을 서로 교환하는 프로그램을 작성해 보자.  
x와 y는 10, 20 으로 초기화 한다.  
별도의 변수가 필요하다면 선언하여 사용하라.

```
cmd C:\WINDOWS\system32\cmd.exe
x = 10, y = 20
x = 20, y = 10
계속하려면 아무 키나 누르십시오 . . .
```

# 실습 문제

```
1 #include<stdio.h>
2
3 int main(void)
4 {
5     int x = 10, y = 20;
6     int tmp;
7
8     printf("x = %d, y = %d\n", x, y);
9     tmp = x;
10    x = y;
11    y = tmp;
12
13    printf("x = %d, y = %d\n", x, y);
14
15    return 0;
16 }
```

# 실습 문제

- 상자의 부피를 구하는 프로그램을 작성하여 보자. 부피는 길이\*너비\*높이로 계산하며, 모두 double형의 실수로 입력 받는다.  
**hint)** `scanf("%lf %lf %lf", &w, &h, &l);` 와 같이 하면 3개의 값을 한번에 입력 받을 수 있다.

C:\WINDOWS\system32\cmd.exe

```
상자의 가로, 세로, 높이를 한번에 입력 : 30 30 30
상자의 부피는 27000.000000입니다.
계속하려면 아무 키나 누르십시오 . . .
```

# 실습 문제

```
1 #include<stdio.h>
2
3 int main(void)
4 {
5     double w, l, h;
6
7     printf("상자의 가로, 세로, 높이를 한번에 입력 : ");
8     scanf("%lf %lf %lf", &w, &l, &h);
9
10    printf("상자의 부피는 %lf입니다.\n", w * l * h);
11
12    return 0;
13 }
```

# 산술연산자

연산	연산자	수식
덧셈	+	$x + y$
뺄셈	-	$x - y$
곱셈	*	$x * y$
나눗셈	/	$x / y$
나머지	%	$x \% y$

다양한 수식을 바꿔보자)

$$y = mx + b \quad \rightarrow \quad y = m * x + b;$$

$$y = ax^2 + bx + c \quad \rightarrow \quad y = a * x * x + b * x + c;$$

$$m = \frac{x+y+z}{3} \quad \rightarrow \quad m = (x + y + z) / 3;$$

# %의 용도

```
1 #include<stdio.h>
2 #define SEC_PER_MINUTE 60
3
4 int main() {
5     int input, minute, second;
6
7     printf("초를 입력하세요 : ");
8     scanf("%d", &input);
9
10    minute = input / SEC_PER_MINUTE;
11    second = input % SEC_PER_MINUTE;
12
13    printf("%d초는 %d분 %d초입니다. \n", input, minute, second);
14    return 0;
15 }
```

```
C:\WINDOWS\system32\cmd.exe
초를 입력하세요 : 1000
1000초는 16분 40초입니다.
계속하려면 아무 키나 누르십시오 . . .
```

그 외) 나머지가 10이면 홀수 0이면 짝수임을 이용해서 구분하는 기능, 난수를 생성할 때 등등

# 증감 연산자

증감 연산자	차이점
$++x$	수식의 값은 증가된 $x$ 값이다.
$x++$	수식의 값은 증가되지 않은 원래의 $x$ 값이다.

$++, --$  는 1씩 증가하거나 감소시키는 연산자로  $x = x+1$ 과 같은 의미이다.

# 중간 연산자

```
1 #include<stdio.h>
2 #define SEC_PER_MINUTE 60
3
4 int main() {
5     int x = 10, y = 10;
6
7     printf("x=%d\n", x);
8     printf("++x의 값=%d\n", ++x);
9     printf("x=%d\n\n", x);
10
11    printf("y=%d\n", y);
12    printf("y++의 값=%d\n", y++);
13    printf("y=%d\n\n", y);
14
15    return 0;
16 }
```

```
C:\WINDOWS\system32\cmd.exe
x=10
++x의 값=11
x=11

y=10
y++의 값=10
y=11

계속하려면 아무 키나 누르십시오 . . .
```

# 복합 대입 연산자 & 조건 연산자

복합대입연산자	의미
$x += y$	$x = x + y$
$x -= y$	$x = x - y$
$x *= y$	$x = x * y$
$x /= y$	$x = x / y$
$x %= y$	$x = x \% y$

(조건) ? (참일 때 수행) : (거짓일 때 수행)

Ex)  $(a > b) ? (\max = a) : (\max = b);$

단항 연산자 : 피연산자가 하나인 연산자

ex)  $++, --$  등등

이항 연산자 : 피연산자가 두개인 연산자

ex)  $+, -, *, \%$  등등

삼항 연산자 : 피연산자가 세개인 연산자

ex)  $? :$

# 관계 연산자, 논리 연산자

연산	의미
$x == y$	$x$ 와 $y$ 가 같은가?
$x != y$	$x$ 와 $y$ 가 다른가?
$x > y$	$x$ 가 $y$ 보다 큰가?
$x < y$	$x$ 가 $y$ 보다 작은가?
$x >= y$	$x$ 가 $y$ 보다 크거나 같은가?
$x <= y$	$x$ 가 $y$ 보다 작거나 같은가?

연산	의미
$x \&\& y$	AND연산. $x$ 와 $y$ 가 모두 참이여야 참
$x    y$	OR연산. $x$ 나 $y$ 중 하나만 참이면 참
$!x$	NOT연산. $x$ 가 참이면 거짓, 거짓이면 참

주의))  $2 < x < 5$  처럼 쓰이지 않음!  
 $(2 < x) \&\& (x < 5)$

# 형변환

## ● 자동형변환

1) 대입 연산시 자동형변환 : 올림변환과 내림변환

올림변환 : `double f = 10;` 이면 f에는 10.0 저장

내림변환 : `int i = 3.14;` 이면 i에는 3 저장

2) 수식 연산시 자동형변환 : 원칙적으로 수식의 피연산자는 자료형이 동일해야 함. 서로 다른 자료형이 온다면 자동으로 형변환이 일어난다.

`int<unsigned int<long<unsigned long<float<double`

`ex) 10 + 3.14 => 10.0 + 3.14`

## ● 명시적 형변환

`(int)d, (double)i` 처럼 사용하며, 변수의 자료형이 바뀌는 것이 아닌 변수의 값이 변환되는 것이다. Ex) `f = (int)3.14;` -> `f = 3;`

# 실습 문제

- 
- 3개의 정수를 입력 받아서 최대값을 출력하는 프로그램을 작성하라. **hint) ?:연산자를 이용한다.**

```
C:\WINDOWS\system32\cmd.exe
```

```
3개의 정수를 입력하시오 : 19 2 12
```

```
최대값 : 19
```

```
계속하려면 아무 키나 누르십시오 . . .
```

# 실습 문제

```
1      #include<stdio.h>
2
3      int main(void)
4      {
5          int x, y, z;
6          int large;
7
8          printf("3개의 정수를 입력하시오 : ");
9          scanf("%d %d %d", &x, &y, &z);
10
11         large = x > y ? x : y;
12         large = large > z ? large : z;
13
14         printf("최대값 : %d\n", large);
15
16         return 0;
17     }
```

# 실습 문제

- 100보다 작은 정수를 입력 받아서 이를 십의 자리, 일의 자리로 분리하여 출력하는 프로그램을 작성하라.  
(입력 값은 100보다 작다고 가정)



C:\WINDOWS\system32\cmd.exe

```
정수를 입력하시오 : 93
십의 자리 : 9
일의 자리 : 3
계속하려면 아무 키나 누르십시오 . . .
```

# 실습 문제

```
1 #include<stdio.h>
2
3 int main(void)
4 {
5     int number;
6     int ten, one;
7
8     printf("정수를 입력하시오 : ");
9     scanf("%d", &number);
10
11    ten = number / 10;
12    one = number % 10;
13
14    printf("십의 자리 : %d\n일의 자리 : %d\n", ten, one);
15
16    return 0;
17 }
18 }
```

# 다음주



## 조건문과 반복문



**감사합니다**