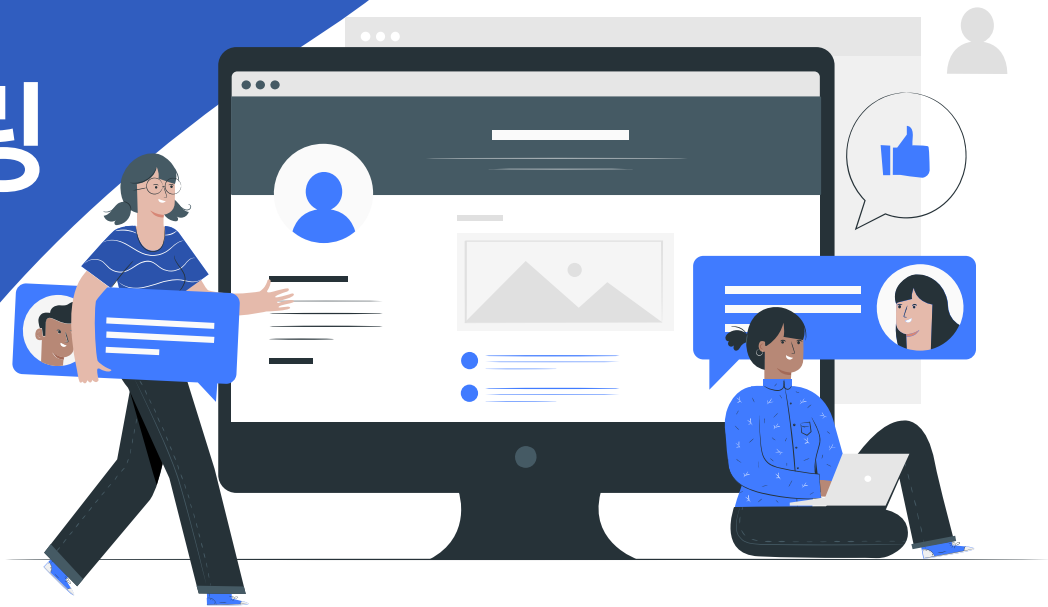


7주차 튜터링



01 함수2



목차

- 변수들을 서로 Swap하기
 - tmp를 이용한 방법
 - 비트 논리 연산자를 이용하는 방법.
- 매개변수 전달 방법

변수들을 서로 Swap하기

```
#include <stdio.h>

int main(void)
{
    int x = 1;
    int y = 2;

    int tmp = x;
    x = y;
    y = tmp;

    printf("x\t : %d\n", x);
    printf("y\t : %d\n", y);
    printf("tmp\t : %d\n", tmp);

    return 0;
}
```

- 임시 저장 공간인 tmp를 만들어 변수를 서로 swap 하는 방법.
- 이 방법은 너무 유명해서 다들 알고 있죠~^^

변수들을 서로 Swap하기

```
#include <stdio.h>

int main(void)
{
    int x = 1;
    int y = 2;

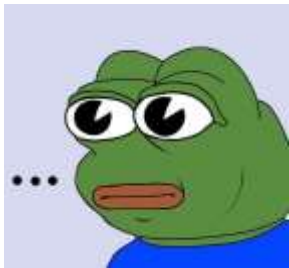
    x = x ^ y;
    y = x ^ y;
    x = x ^ y;

    printf("x\t : %d\n", x);
    printf("y\t : %d\n", y);

    return 0;
}
```

- 비트 연산자를 이용해서 변수를 swap 하는 방법

잉? ^는 뭐지??



비트 논리 연산자

	연산자	예시	설명
대입	=	a = b;	a = b;
산술	+=	a += b;	a = a + b;
	-=	a -= b;	a = a - b;
	*=	a *= b;	a = a * b;
	/=	a /= b;	a = a / b;
	%=	a %= b;	a = a % b;
(비트) 논리	&=	a &= b;	a = a & b;
	^=	a ^= b;	a = a ^ b;
	=	a = b;	a = a b;
쉬프트	<<=	a <<= b;	a = a << b;
	>>=	a >>= b;	a = a >> b;

- 사실 C언어의 연산자는 우리가 저번에 배운 연산자들 말고도 나머지 두 개가 더 있음.
- 바로 비트 논리 연산자 하고 쉬프트 연산자임

(쉬프트 연산자는 나올 기회가 있으면 설명하겠습니다.
우선 비트 논리 연산자는 나올 기회가 나와서 합니다.)

비트 논리 연산자

논리 연산자

- A와 B의 상황을 일정한 규칙(AND, OR, NOT)으로 연결해주는 연산자
- A와 B는 진리값(참, 거짓)을 사용
- 참 : 0이 아닌 모든 값
- 거짓 : 0

논리 연산자

A	B	A & B	A B	!A
거짓(0)	거짓(0)	거짓(0)	거짓(0)	참(1)
거짓(0)	참(1)	거짓(0)	참(1)	참(1)
참(1)	거짓(0)	거짓(0)	참(1)	거짓(0)
참(1)	참(1)	참(1)	참(1)	거짓(0)

- 관계 연산자 다 기억하죠~^^
- 그런데 논리 연산자가 여기서 끝이 아니었습니다. 더 있어요!

비트 논리 연산자

a	b	a&b	a b	a^b	~a
0	0	0	0	0	1
0	1	0	1	1	1
1	0	0	1	1	0
1	1	1	1	0	0

• 비트 논리 연산의 예

& (비트 AND 연산)	(비트 OR 연산)	^ (비트 XOR 연산)	~ (비트 NOT 연산)
01000001 & 10000001 ===== = 00000001	01000001 10000001 ===== = 11000001	01000001 ^ 10000001 ===== = 11000000	01000001 ~ ===== = 10111110

• \wedge == xor연산 이라고 부른다.

• xor : 논리식이 서로 다르면 참을 반환한다

... 라는 뜻임

비트 논리 연산자

A	B	A and B (A && B)	A or B (A B)	A xor B	!A
true	true	true	true	false	false
true	false	false	true	true	false
false	true	false	true	true	true
false	false	false	false	false	true

비트 논리 연산자

```
#include <stdio.h>

int main(void)
{
    int x = 1;
    int y = 2;

    x = x ^ y;
    y = x ^ y;
    x = x ^ y;

    printf("x\t : %d\n", x);
    printf("y\t : %d\n", y);

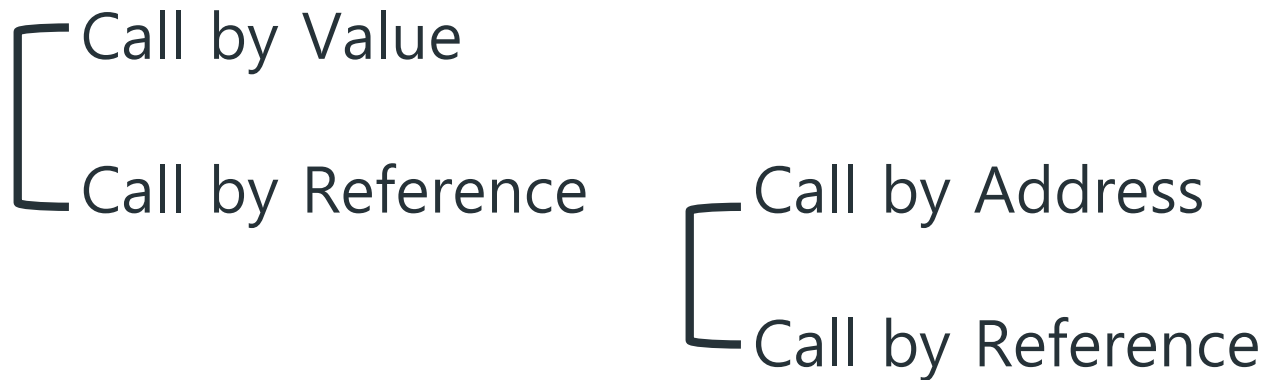
    return 0;
}
```

x = 1, y = 2 일 때.

a) $x = x \oplus y = 3$		0 0 0 1
	XOR	0 0 1 0
	=	0 0 1 1
b) $y = x \oplus y = 1$		0 0 1 1
	XOR	0 0 1 0
	=	0 0 0 1
c) $x = x \oplus y = 2$		0 0 1 1
	XOR	0 0 0 1
	=	0 0 1 0

결과 : x = 2, y = 1

매개변수 전달 방법



매개변수 전달 방법

```
#include <stdio.h>

void swap(int a, int b)
{
    int tmp = a;
    a = b;
    b = tmp;
}

int main(void)
{
    int x = 1;
    int y = 2;

    swap(x, y);
    printf("x : %d\n", x);
    printf("y : %d\n", y);
}
```

- 그럼 이 코드도 과연 swap이 될까...?

매개변수 전달 방법

```
#include <stdio.h>

void swap(int *a, int *b)
{
    int tmp = *a;
    *a = *b;
    *b = tmp;
}

int main(void)
{
    int x = 1;
    int y = 2;

    swap(&x, &y);
    printf("x : \t%d\n", x);
    printf("y : \t%d\n", y);
}
```

- 되게 하는 방법이 있다.
바로 포인터(*)이다!
- 포인터가 무엇인지는 다음에
기회가 된다면
설명하겠습니다... 😂



END