

연산자의 종류

연산자	연산자의 종류
산술 연산자	+, -, *, /, %
논리 연산자	&&, , !
비교 연산자	==, <, >, <=, >=, !=
대입 연산자	=
삼항 연산자	?
비트 연산자	^, &, , >>, <<

if

```
if(조건){
    코드
}
else if(조건){
    코드
}
else if(조건){
    코드
}
else{
    코드
}
```

```
#include<stdio.h>
int main()
{
    int a;
    scanf("%d", &a);
    if (a < 100) {
        printf("a는 100보다 작습니다.\n");
    }
    else if (a == 100) {
        printf("a는 100과 같습니다.\n");
    }
    else {
        printf("a는 100보다 큼니다.\n");
    }
    return 0;
}
```

input:

59

output:

a는 100보다 작습니다.

if문은 if(){} 로 사용한다. ()안에는 조건식이 들어가며, {}안에는 조건식이 참일 때 실행할 코드들이 들어간다.

if 조건이 참일 때 실행하고 싶은 조건들이 else if 안에 들어가며, if와 else if가 모두 거짓이면 else를 실행한다.

하나의 조건문 안에는 하나의 if와 0개 이상의 else if, 하나 이하의 else가 존재해야 한다.

switch

```
switch(변수){  
case 값:  
    코드  
    break;  
case 값:  
    코드  
    break;  
default:  
    코드  
    break;  
}
```

```
#include<stdio.h>  
int main()  
{  
    int a, b;  
    char op;  
    double ans;  
    scanf("%d %c %d", &a, &op, &b);  
    switch (op) {  
    case '+':  
        ans = a + b;  
        break;  
    case '-':  
        ans = a - b;  
        break;  
    case '*':  
        ans = a * b;  
        break;  
    case '/':  
        ans = (double)a / b;  
        break;  
    default:  
        break;  
    }  
  
    printf("%.2lf\n", ans);  
    return 0;  
}
```

input:

6+3

output:

9.00

switch문은 switch(){ }로 사용한다. 다양한 연산자를 사용할 수 있는 if문과 달리 switch문은 case 옆에 작성한 것과 ()안에 작성한 변수를 서로 같은지 비교하는 것만 가능하다.

case는 if문의 if와 else if와 같다고 생각하면 되고,
default는 if문의 else와 같다고 생각하면 된다.

break를 써주지 않으면 실행된 case부터 break를 만나기 전까지 전부 실행하므로,
switch문에서는 break의 유무가 가장 중요하다.

※ switch문은 if문보다 빠르다. 그러나, 현대에 와서는 유의미한 가치를 가지지는 않는다.

※ 실수형 서식지정자는 %뒤에 .숫자를 붙여서 소숫점을 제한할 수 있다.

ex) %.2lf : 소수점 둘째자리까지

%.4lf : 소수점 넷째자리까지