

표현식 (Expressions)

표현식	값
4	4
-6	-6
4 + 21	25
q = 5 * 2	10
3 + (c = 1 + 2)	6
2 > 1	1
2 < 1	0
a	a의 값
x = ++q % 3	x와 q의 값에 의해 결정
q > 3	q의 값에 의해 결정
a * (b / c + d) / 20	a, b, c, d의 값에 의해 결정

⏪ ⏩ ↺ ⏴ ⏵ ⏶ ⏷

- 표현식의 주요 기능은 값을 계산하는 것
- 대입하는 것은 부가적인 것(side-effect)
- True -> 1, False -> 0

문장 (Statement)

/* Statements */

```
int x, y, apples; // declaration statement
apples = 3; // assignment statement
; //null statement
7;
1 + 2;
x = 4;
++x;
x = 1 + (y = 5); // y = 5 is subexpression

while (x++ < 10) // while statement (structured statements)
    y = x + y;

printf("%d\n", y); // function statement

return 0; // return statement
```

⏪ ⏩ 🔍 🔄 📄 🗑️

/* Side Effects and Sequence Points */

```
x = 4; // main intent is evaluating expressions
y = 1 + x++;

while (x++ < 10) // (x++ < 10) is a full expression.
    printf("%d\n", x);

y = (4 + x++) + (6 + x++); // Not (4 + x++) nor (6 + x++) is a full expression.
```

⏪ ⏩ 🔍 🔄 📄 🗑️

- 명확한 의도를 담아서 표현해야 한다.