

C# 기본 문법

HyoJoon Han
동국대학교
han6343@dongguk.edu



연산자

- 연산자
 - +, -, *, / 와 같이 이미 정의된 연산을 수행하는 기호
- 피연산자
 - 연산의 대상 또는 연산자의 동작을 받는 변수나 값

그림 4-5 연산자와 피연산자



- 산술 연산자

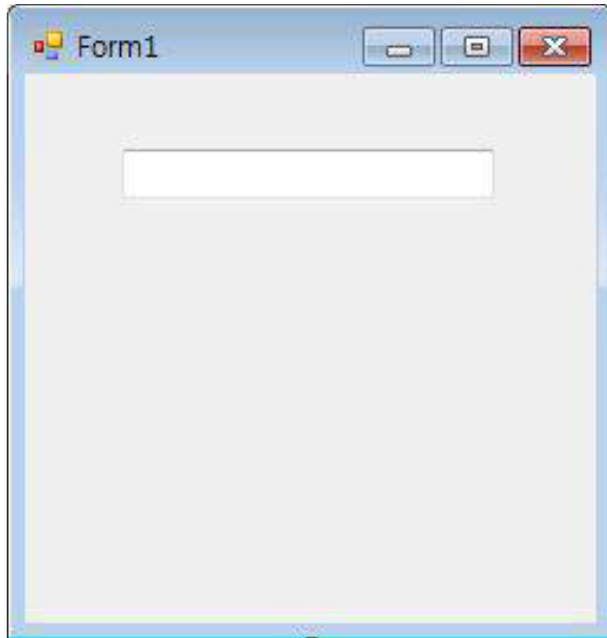
| 연산자 | 일반 형식 | 의미 | 사용 예 |
|-------|----------------|-------------------------|----------------|
| + | $x+y$ | 덧셈 | $2+5$ |
| - | $x-y$ | 뺄셈 | $5-3$ |
| * | $x*y$ | 곱셈 | $4*6$ |
| / | x/y | 나눗셈 | $8/3$ |
| Pow() | Math.Pow(x, y) | 거듭제곱 | Math.Pow(3, 4) |
| % | $x\%y$ | x를 y로 나눈 나머지를 구함(mod연산) | $8\%3$ |

그림 4-6 Mod 연산의 예

$$\begin{array}{r}
 7 \leftarrow \text{몫} \\
 2 \overline{) 15} \\
 \underline{14} \\
 1 \leftarrow \text{나머지}
 \end{array}$$

- 실습 – 산술 연산자 사용하기

1) 윈도우 폼 디자인



2) 속성 설정

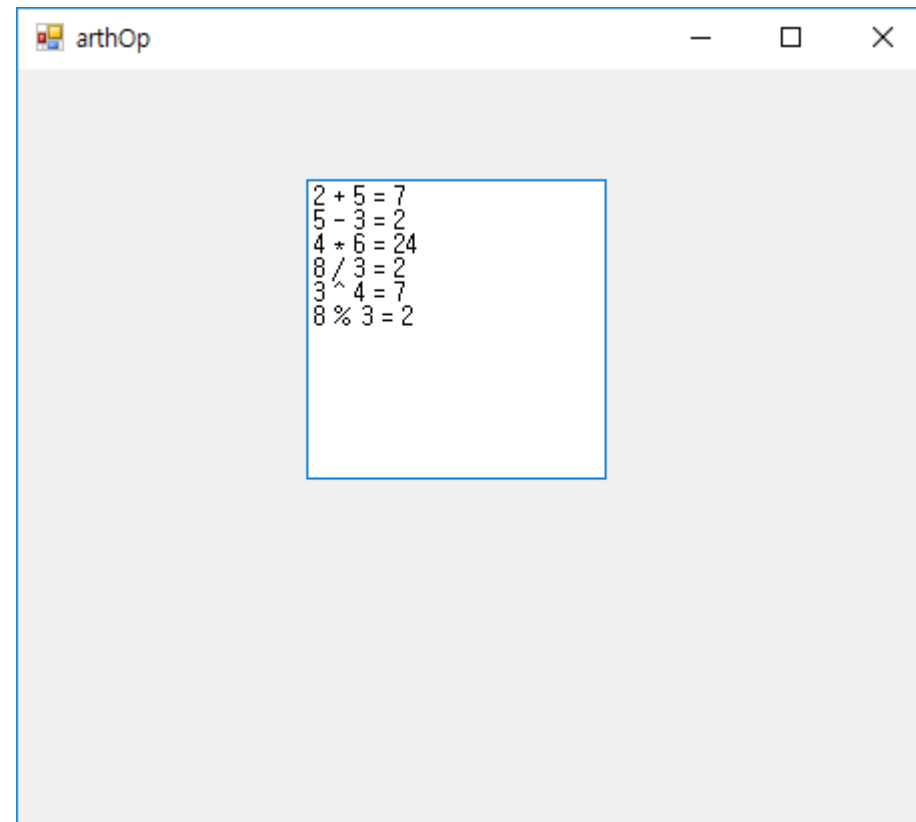
| 컨트롤 | 속성 | 속성값 |
|----------|-----------|----------|
| Form1 | Text | 산술 연산자 |
| | MultiLine | True |
| TextBox1 | Size | 150, 150 |
| | Text | (빈칸) |

3) 코드 작성

```
private void arthOp_Load(object sender, EventArgs e)
{
    int plus = 2 + 5;
    int minus = 5 - 3;
    int mul = 4 * 6;
    double div = 8 / 3;
    int pow = (int)Math.Pow(3, 4);
    int mod = 8 % 3;

    textBox1.Text = "2 + 5 = " + plus.ToString() + Environment.NewLine +
        "5 - 3 = " + minus.ToString() + Environment.NewLine +
        "4 * 6 = " + mul.ToString() + Environment.NewLine +
        "8 / 3 = " + div.ToString() + Environment.NewLine +
        "3 ^ 4 = " + pow.ToString() + Environment.NewLine +
        "8 % 3 = " + mod.ToString();
}
```

4) 실행 결과

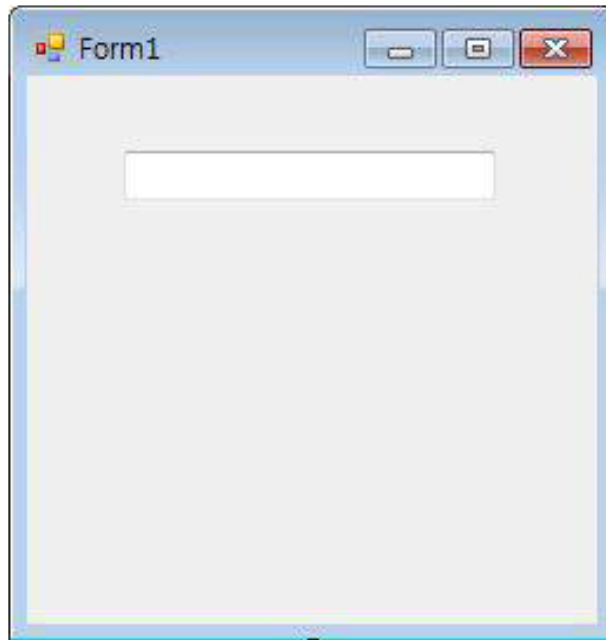


- 관계(비교) 연산자
 - 두 피연산자의 대소 관계를 비교하여 참이면 True, 거짓이면 False 값을 반환

| 연산자 | 일반 형식 | 의미 | 사용 예 |
|-----|------------|-------------------------|--------------------------------------|
| < | $x < y$ | 작다(x가 작으면 True) | $9 < 15 \rightarrow \text{True}$ |
| <= | $x \leq y$ | 작거나 같다(x가 작거나 같으면 True) | $21 \leq 21 \rightarrow \text{True}$ |
| > | $x > y$ | 크다(x가 크면 True) | $31 > 33 \rightarrow \text{False}$ |
| >= | $x \geq y$ | 크거나 같다(x가 크거나 같으면 True) | $11 \geq 10 \rightarrow \text{True}$ |
| == | $x == y$ | 같다(x와 y가 같으면 True) | $23 == 25 \rightarrow \text{False}$ |
| != | $x != y$ | 같지 않다(x와 y가 다르면 True) | $24 != 24 \rightarrow \text{False}$ |

- 실습 – 관계 연산자 사용하기

1) 윈도우 폼 디자인



2) 속성 설정

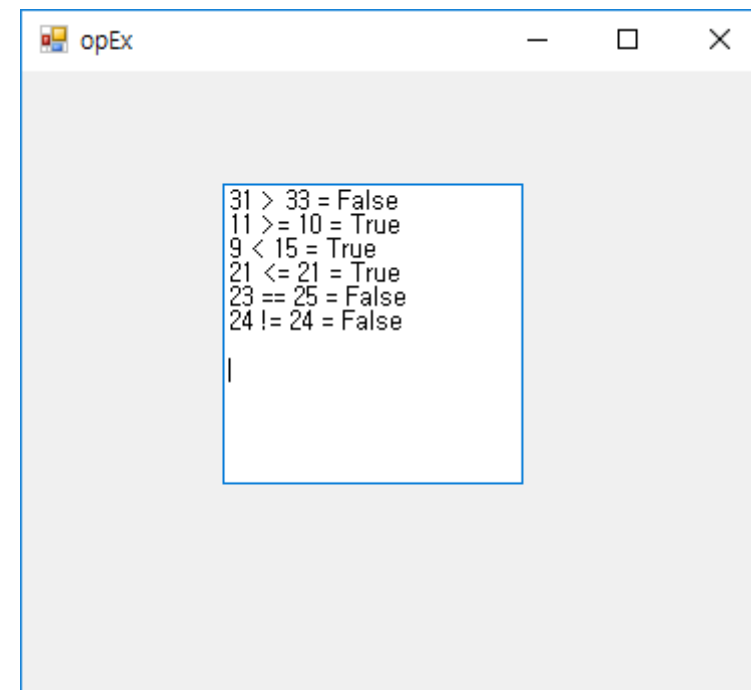
| 컨트롤 | 속성 | 속성값 |
|----------|-----------|----------|
| Form1 | Text | 관계 연산자 |
| | MultiLine | True |
| TextBox1 | Size | 150, 150 |
| | Text | (빈칸) |

3) 코드 작성

```
private void opEx_Load(object sender, EventArgs e)
{
    bool op1 = 31 > 33;
    bool op2 = 11 >= 10;
    bool op3 = 9 < 15;
    bool op4 = 21 <= 21;
    bool op5 = 23 == 25;
    bool op6 = 24 != 24;

    textBox1.Text = "31 > 33 = " + op1.ToString() + Environment.NewLine +
        "11 >= 10 = " + op2.ToString() + Environment.NewLine +
        "9 < 15 = " + op3.ToString() + Environment.NewLine +
        "21 <= 21 = " + op4.ToString() + Environment.NewLine +
        "23 == 25 = " + op5.ToString() + Environment.NewLine +
        "24 != 24 = " + op6.ToString();
}
```

4) 실행 결과

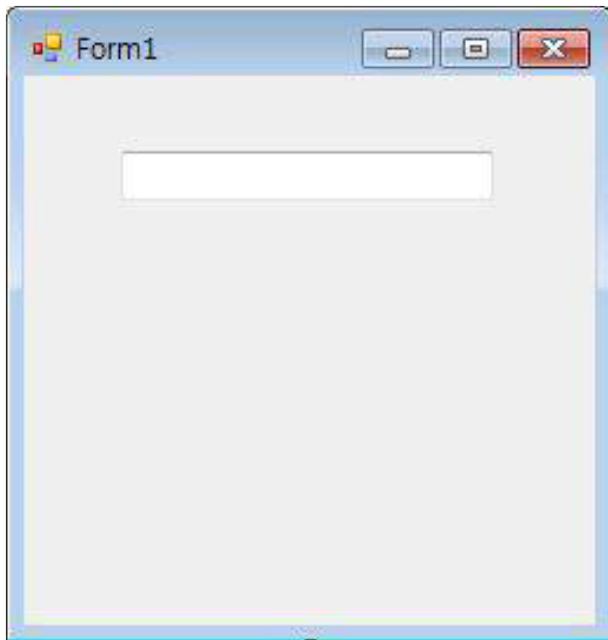


- 논리 연산자
 - 논리식의 참과 거짓을 판정하여 True/False를 반환

| 연산자 | 일반 형식 | 의미 |
|-----|--------------|------------------------------|
| and | $x \& y$ | x, y 둘다 True 일 때만 True |
| or | $x y$ | x, y 둘 중 하나가 True 이면 True |
| xor | $x \wedge y$ | x, y가 서로 다르면 True |
| not | $!x$ | True 이면 False, False 이면 True |

- 실습 – 논리 연산자 사용하기

1) 윈도우 폼 디자인



2) 속성 설정

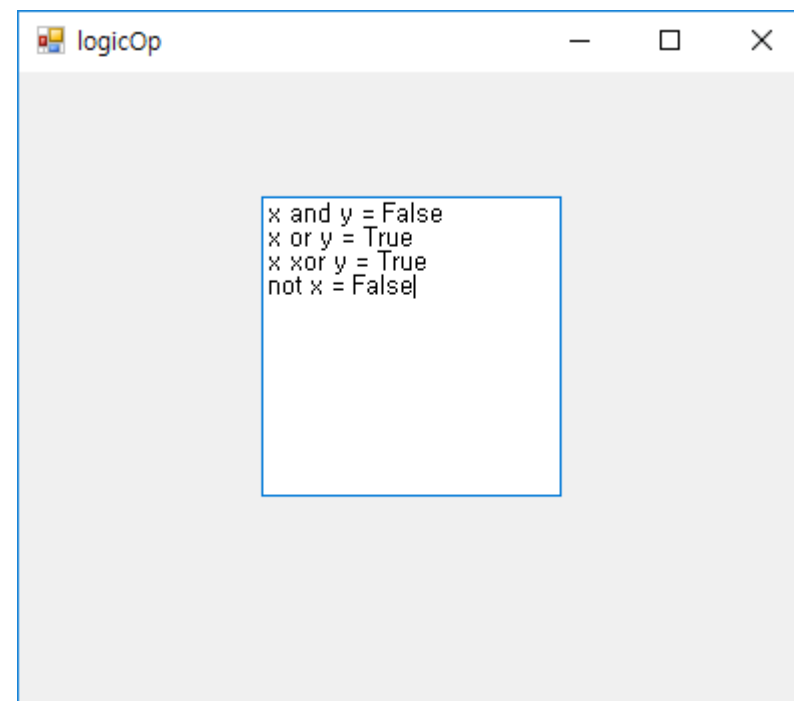
| 컨트롤 | 속성 | 속성값 |
|----------|-----------|----------|
| Form1 | Text | 논리 연산자 |
| | MultiLine | True |
| TextBox1 | Size | 150, 150 |
| | Text | (빈칸) |

3) 코드 작성

```
private void logicOp_Load(object sender, EventArgs e)
{
    bool x = true, y = false;
    bool op1 = x & y;
    bool op2 = x | y;
    bool op3 = x ^ y;
    bool op4 = !x;

    textBox1.Text = "x and y = " + op1.ToString() + Environment.NewLine +
        "x or y = " + op2.ToString() + Environment.NewLine +
        "x xor y = " + op3.ToString() + Environment.NewLine +
        "not x = " + op4.ToString();
}
```

4) 실행 결과



- 대입 연산자

표 4-9 대입 연산자의 종류

| 연산자 | 일반 형식 | 의미 |
|-----|---------|---|
| = | $x = y$ | <ul style="list-style-type: none"> y의 값을 x에 저장한다. 오른쪽의 값을 왼쪽에 대입한다. |

$x = 10$

→ 올바른 표현

$x = y + 10$

→ 올바른 표현

$10 = x$

→ 잘못된 표현

$x + y = 15$

→ 잘못된 표현

형식 | (①) = (②)

- 1의 자리에는 값을 저장할 수 있는 변수 등이 와야 하고, 2의 자리에는 값 자체나 값을 담고 있는 변수 등이 올 수 있다.

- 복합 연산자
 - 자료형에 적용하는 기본 연산자와 = 연산자를 함께 사용

| 연산자 | 일반 형식 | 의미 | |
|-----|---------|---------------------------|--------------|
| += | $x+=y$ | y의 값을 x에 더하고 결과를 x에 저장 | $x = x + y$ |
| -= | $x-=y$ | y의 값을 x에서 빼고 결과를 x에 저장 | $x = x - y$ |
| *= | $x*=y$ | y의 값을 x에 곱하고 결과를 x에 저장 | $x = x * y$ |
| /= | $x/=y$ | y의 값을 x에 나하고 결과를 x에 저장 | $x = x / y$ |
| %= | $x\%=y$ | y의 값을 x에 나눈 나머지 결과를 x에 저장 | $x = x \% y$ |

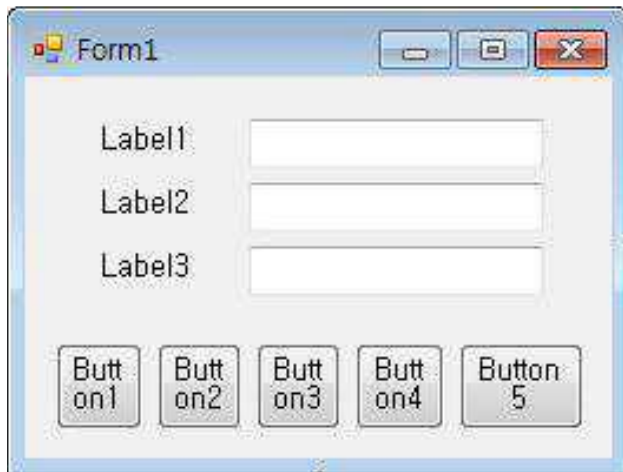
- 증감 연산자
 - 피연산자를 1씩 증가하거나 감소할 수 있음

| 연산자 | 일반 형식 | 의미 | |
|-----|-------|-------------------------|-----------|
| ++ | x++ | 연산을 수행 한 후 x의 값을 1 증가 | 후위 증가 연산자 |
| | ++x | 연산을 수행 하기 전에 x의 값을 1 증가 | 전위 증가 연산자 |
| -- | x-- | 연산을 수행 한 후 x의 값을 1 감소 | 후위 감소 연산자 |
| | --x | 연산을 수행 하기 전에 x의 값을 1 감소 | 전위 감소 연산자 |

- 연산자 우선순위
 - 연산자는 크게 괄호 -> 산술 연산자 -> 관계 연산자 -> 논리 연산자 순으로 계산
 - 비교 연산자끼리는 우선순위가 같아 왼쪽에서 오른쪽으로 진행

- 실습 – 간단한 사칙연산 계산기 만들기

1) 윈도우 폼 디자인



2) 속성 설정

| 컨트롤 종류 | 속성 | 속성 값 | 컨트롤 종류 | 속성 | 속성 값 |
|----------|------|----------|----------|------|------|
| Form1 | Text | 사칙연산 계산기 | TextBox3 | Text | (빈칸) |
| Label1 | Text | 입력값1 | Button1 | Text | + |
| Label2 | Text | 입력값2 | Button2 | Text | - |
| Label3 | Text | 결과값 | Button3 | Text | * |
| TextBox1 | Text | (빈칸) | Button4 | Text | / |
| TextBox2 | Text | (빈칸) | Button5 | Text | 지우기 |

3) 코드 작성

```
private void button1_Click(object sender, EventArgs e)
{
    int x = int.Parse(textBox1.Text);
    int y = int.Parse(textBox2.Text);
    int result = x + y;
    textBox3.Text = result.ToString();
}

private void button2_Click(object sender, EventArgs e)
{
    int x = int.Parse(textBox1.Text);
    int y = int.Parse(textBox2.Text);
    int result = x - y;
    textBox3.Text = result.ToString();
}

private void button3_Click(object sender, EventArgs e)
{
    int x = int.Parse(textBox1.Text);
    int y = int.Parse(textBox2.Text);
    int result = x * y;
    textBox3.Text = result.ToString();
}
```

3) 코드 작성

```
private void button4_Click(object sender, EventArgs e)
{
    int x = int.Parse(textBox1.Text);
    int y = int.Parse(textBox2.Text);
    int result = x / y;
    textBox3.Text = result.ToString();
}

private void button5_Click(object sender, EventArgs e)
{
    textBox1.Text = "";
    textBox2.Text = "";
    textBox3.Text = "";
    textBox1.Focus();
}
```

4) 실행 결과

