# **C# Operators**

## 

## **Оператори (Operators)** - використовуються для виконання операцій над змінними та значеннями. У наведеному нижче прикладі ми використовуємо оператор *“+”* для додавання двох значень:

int x = 100 + 50;

Хоча оператор *“+”* часто використовується для додавання двох значень, як у наведеному вище прикладі, його також можна використовувати для додавання змінної та значення або змінної та іншої змінної:

int sum1 = 100 + 50; // 150 (100 + 50)

int sum2 = sum1 + 250; // 400 (150 + 250)

int sum3 = sum2 + sum2; // 800 (400 + 400)

**Арифметичні оператори (Arithmetic Operators)** - використовуються для виконання звичайних математичних операцій:

| **+** | **Addition (Додавання)** | **Додає два значення** | **x + y** |  |
| --- | --- | --- | --- | --- |
| **-** | **Subtraction (Віднімання)** | **Віднімає одне значення від іншого** | **x - y** |  |
| **\*** | **Multiplication (Множення)** | **Множить два значення** | **x \* y** |  |
| **/** | **Division (Ділення)** | **Ділить одне значення на інше** | **x / y** |  |
| **%** | **Modulus (Ділення по модулю)** | **Returns the division remainder** | **x % y** |  |
| **++** | **Increment (Збільшення)** | **Збільшує значення змінної на 1** | **x++** |  |
| **--** | **Decrement (Зменшення)** | **Зменшує значення змінної на 1** | **x--** |  |

# **Оператори присвоювання C#**

# **(C# Assignment Operators)**

**Оператори присвоювання (Assignment Operators)** - використовуються для присвоювання значень змінним.

У наведеному нижче прикладі ми використовуємо оператор присвоювання “=” для присвоєння значення 10 змінній з іменем x:

int x = 10;

Оператор присвоювання додавання “+=” додає значення до змінної:

int x = 10;

x += 5;

//Input: 15

**Список усіх операторів присвоювання:**

| **=** | **x = 5** | **x = 5** |  |
| --- | --- | --- | --- |
| **+=** | **x += 3** | **x = x + 3** |  |
| **-=** | **x -= 3** | **x = x - 3** |  |
| **\*=** | **x \*= 3** | **x = x \* 3** |  |
| **/=** | **x /= 3** | **x = x / 3** |  |
| **%=** | **x %= 3** | **x = x % 3** |  |
| **&=** | **x &= 3** | **x = x & 3** |  |
| **|=** | **x |= 3** | **x = x | 3** |  |
| **^=** | **x ^= 3** | **x = x ^ 3** |  |
| **>>=** | **x >>= 3** | **x = x >> 3** |  |
| **<<=** | **x <<= 3** | **x = x << 3** |  |

**Оператори порівняння у C#**

**(C# Comparison Operators)**

**Оператори порівняння (Comparison operators**) - використовуються для порівняння двох значень (або змінних). Це важливо у програмуванні, оскільки допомагає нам знаходити відповіді та приймати рішення.

Значення, яке повертає оператор порівняння, є або *True*, або *False*.

У наступному прикладі ми використовуємо оператор більше ніж “>”, щоб з'ясувати, чи 5 більше ніж 3:

int x = 5;

int y = 3;

Console.WriteLine(x > y); // returns True because 5 is greater than 3

**Список усіх операторів порівняння :**

| **==** | **Equal to (дорівнює)** | **x == y** |  |
| --- | --- | --- | --- |
| **!=** | **Not equal**  **(не дорівнює)** | **x != y** |  |
| **>** | **Greater than (більш ніж)** | **x > y** |  |
| **<** | **Less than**  **(менш ніж)** | **x < y** |  |
| **>=** | **Greater than or equal to**  **(більше рівне)** | **x >= y** |  |
| **<=** | **Less than or equal to**  **(менше рівне)** | **x <= y** |  |

# **Логічні оператори C#**

# **(C# Logical Operators)**

Як і у випадку з операторами порівняння, ви також можете перевіряти значення *True* або *False* за допомогою логічних операторів.

Логічні оператори використовуються для визначення логіки між змінними або значеннями:

| **&&** | **Logical and (логічне “і”)** | **Повертає True, якщо обидва оператори істинні** | **x < 5 && x < 10** |  |
| --- | --- | --- | --- | --- |
| **||** | **Logical or**  **(логічне “або”)** | **Повертає True, якщо один з операторів істинний** | **x < 5 || x < 4** |  |
| **!** | **Logical not**  **(логічне “ні”)** | **Реверсує результат, повертає False, якщо результат істинний** | **!(x < 5 && x < 10)** |  |