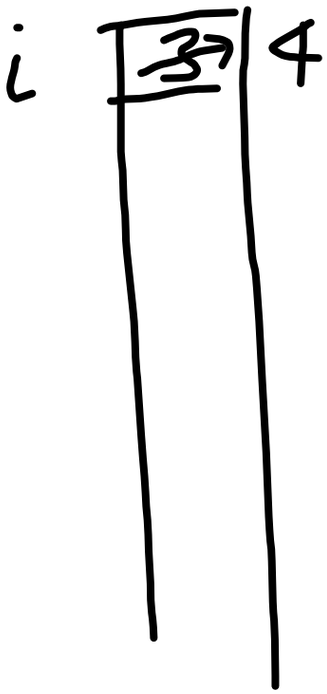
Math vs. Programming

$i = i + 1$ math (never true)

$i = i + 1;$ //programming



$i == i + 1$

"is assigned the value"

Math operators

int
float

+	add
-	subtract
*	multiply
/	divide

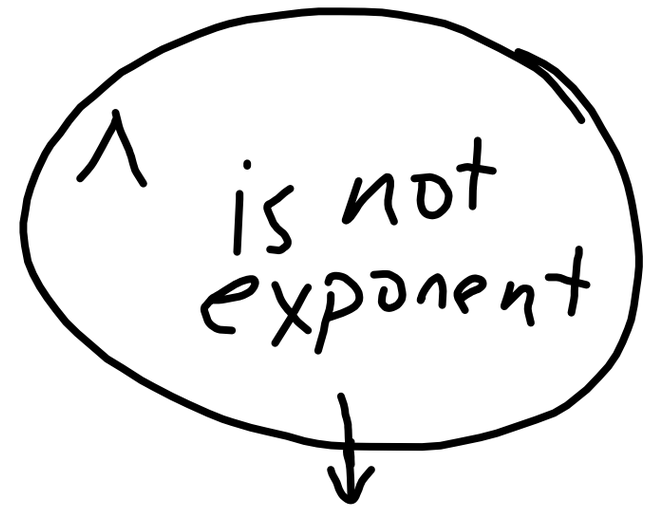
only int (long)

%	modulus

order of operations is in effect

PEMDAS

↑ modulus



pow( )

$e = mc^2$

```
e = m*c*c;
```

```
static void Main (string[] args)
{
    int a = 3, b = 2, c = 7;
    double x = 3.1, y = 4.2, z = 6.8;
    :
    a = b + c; //a = 9
    console.WriteLine(a);
    a = b * c; //a = 14
    a = c / b; // a =
    :
    :
}
```

c=7, b=2

a = c / b; // ~~a = 3.5~~

(3)

2 | 7

6

1

a | 3 |

truncate (get rid of decimal)

double x = 2.0, y = 7.0;

a = y / x;

3.5

a → 3

3.5

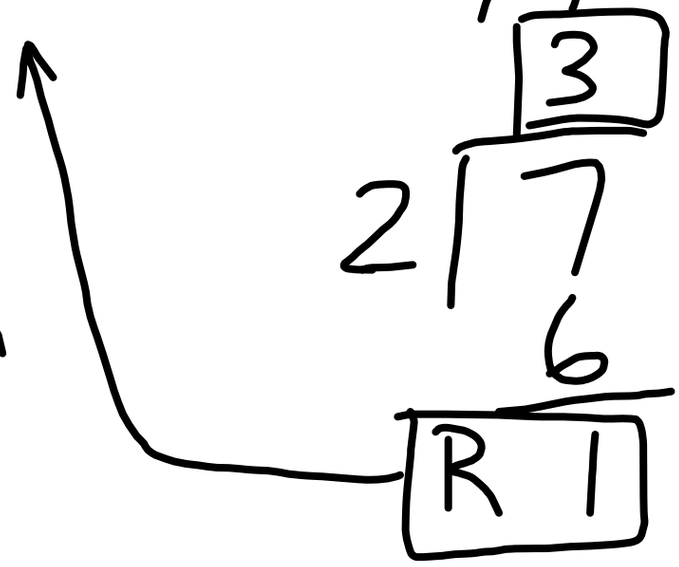
b=2, c=7

a = c / b; // a = 3

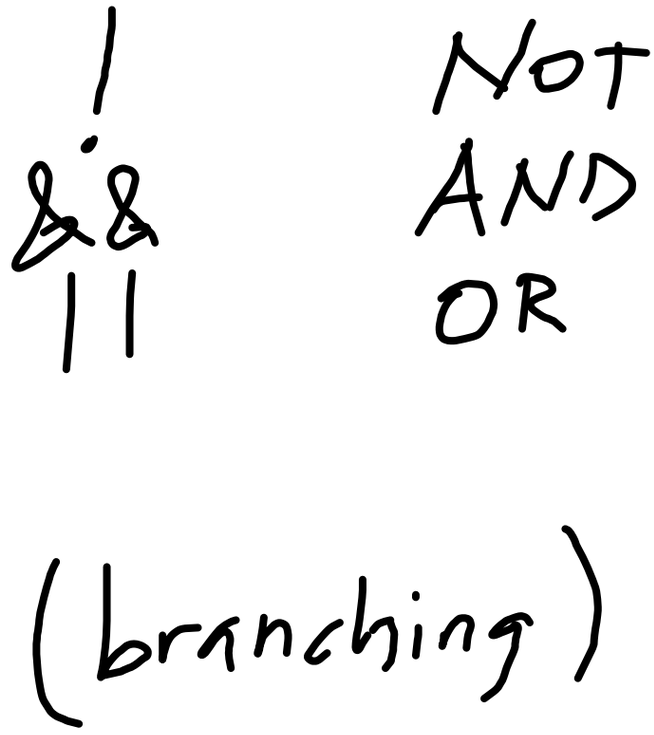
a = c % b; // modulus

modulus

finds remainder of int division



Logical operators



Compound operators

increment/decrement

int

+=

-=

*=

/=

%=

a += 7;

a = a + 7;

(add 1) ++ (c++)

(sub 1) --

a = a + 1;

a += 1;

a++;

++a;

Input and output

```
static void Main(string[] args)
   {
      Console.WriteLine("Hello, World!");

      const char sqrtSym = '\u221A';  // unicode square root symbol
      Console.WriteLine(sqrtSym + "" + 7);

      Console.Write("What is your name?  >> ");
      String name = Console.ReadLine();
      Console.WriteLine("Hello " + name);

      Console.Write("What is your age?   >> ");
      int age = Convert.ToInt32(Console.ReadLine());
      Console.WriteLine("age: " + age);

      Console.SetCursorPosition(10, 15);
      Console.Write(name);
      Console.SetCursorPosition(10, 16);
      Console.Write(age);
      Console.ReadKey();
   }
```

Math

```
static void Main(string[] args)
{
    Console.WriteLine("Hello, World!");
    int a = 7;
    int b = 3;
    double c = 7.8;
    double d = 3.4;
    int iAns = a + b;
    double fAns = c + d;

    Console.WriteLine($"a = {a}\t\tb = {b}\nc = {c}\t\td = {d}\n");
    Console.WriteLine("a + b = " + iAns);
    Console.WriteLine($"{c} + {d} = {fAns}");

    /* Do subtraction, multiplication, division */

    iAns = a / b;
    fAns = c / d;
    Console.WriteLine("a / b = " + iAns);
    iAns = a % b;
    Console.WriteLine("a % b = " + iAns);
    Console.WriteLine($"{c} / {d} = {fAns}");

    iAns = a / (int) c; // typecasting (casting)
    fAns = b / d;
    Console.WriteLine("a / c = " + iAns);
    Console.WriteLine($"{b} / {d} = {fAns}");

    Console.ReadKey();
}
```

Random Numbers

```
class Program
{
    static Random rng;
    static void Main(string[] args)
    {
        Console.WriteLine("Hello, World!");
        rng = new Random();

        int dice = rng.Next(6);
        dice = 1 + dice;
        Console.WriteLine(dice);
        dice = rng.Next(1, 7);
        Console.WriteLine(dice);

        Console.ReadKey();
    }
}
```

Math