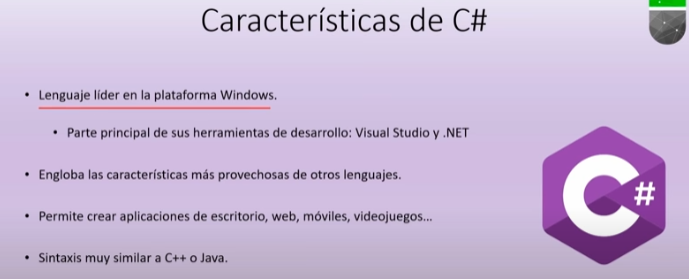
C# Video 1



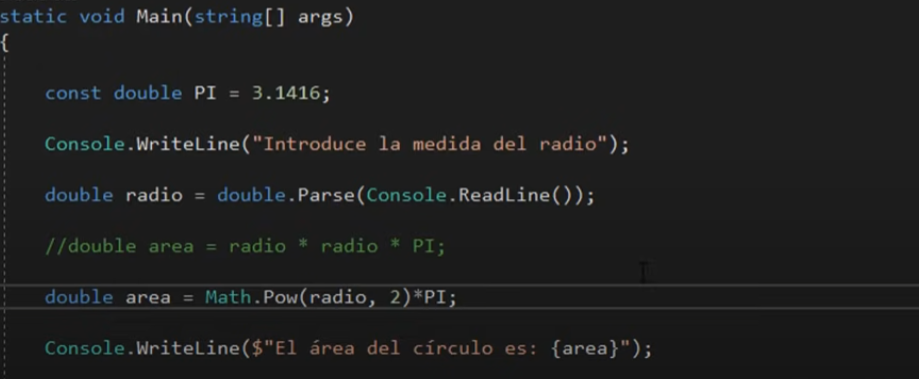
.Net es un framework de Microsoft para el desarrollo de software.



Video 8

Cw es Console.Writeline()

Clase math. Permite realizar diferentes cálculos matématicos: potencias, raíces. Método pow necesita 2 parámetros (base y exponente) math.pow (base, exponente).



La otra forma sería poner radio\*radio.

Videos 9-13: Métodos:

Son grupos de sentencias a los que se les da un nombre identificativo que realizan una tarea en concreto.  
La tarea no se realiza en cualquier momento, sino cuando se llama al método. Puede ser mas de una vez.

Sintaxis:

Tipodedatodevuelto nombredelmétodo (parámetros) {

Cuerpo del método (con return o sin él (void))

}

Pequeña obra de arte:

static void Main(string[] args)

{

Console.WriteLine("Escribe aquí el número para que se sume a 33");

double Numero1 = double.Parse(Console.ReadLine());

int N2 = 33;

double resultado = sumarnumeros(Numero1, N2);

Console.WriteLine("El número devuelto por el método es: " +resultado);

double sumarnumeros (double num1, int num2)

{

return num1 + num2;

}

Toda aplicación comienza a ejecutar la primera línea del método main.

Console.WriteLine("Escribe aquí el número para que se sume a 33");

double Numero1 = double.Parse(Console.ReadLine());

int N2 = 33;

sumarnumeros(Numero1, N2);

}

static void sumarnumeros(double num1, int num2)

{

double resultado = num1 + num2;

Console.WriteLine("El número devuelto por el método es: " + resultado);

Ahí retoqué un poco la obra de arte. Y la saqué del método main, aún dentro de la clase program. Para eso le tuve que poner al método static al inicio, si no tira error.

Método muy sencillo, con una sola línea de código, que es return se puede usar la flecha:

Se simplifica mucho la sintaxis. No se abren llaves, ni se pone el return.

Ámbito o alcances (o contexto) de métodos y variables

Cuando declaramos un método tiene llave de apertura y cierre, lo que está adentro es el ámbito o alcance. Solo será visible lo que allí se pone, en ese ámbito.

void primerMetodo()

{

int num1 = 5; //su ámbito (o alcance o visibilidad) es local

}

void segundoMetodo()

{

Console.WriteLine(num1); //Esto daría error.

}

Siempre se debe declarar una variable antes de usarla, es decir, debe estar escrita mas arriba. EXCEPCION: Las variables se pueden declarar dentro de la clase, pero fuera de los métodos.

A esas variables se las denomina campos de clase.

static void Main(string[] args)

{

primerMetodo();

segundoMetodo();

}

static void primerMetodo()

{

Console.WriteLine(num1);

//su ámbito (o alcance o visibilidad) es local

}

static int num1 = 5;

static void segundoMetodo()

{

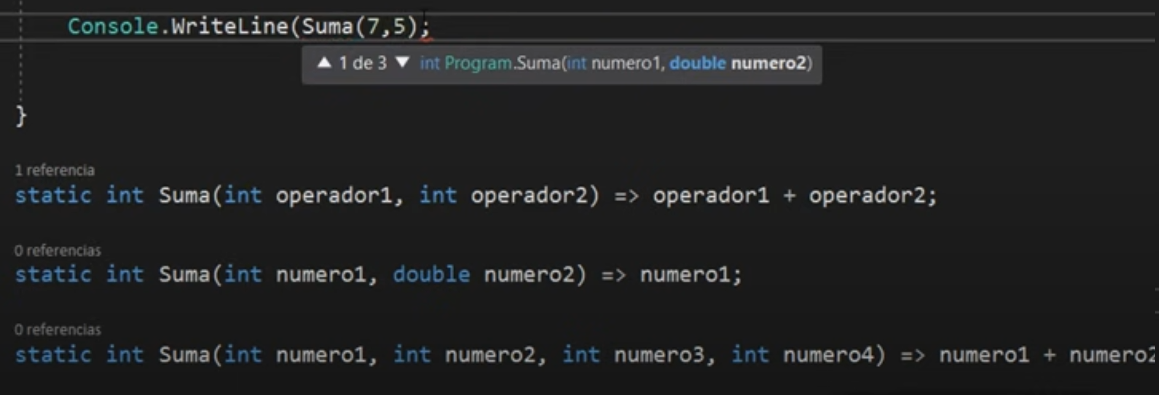
Console.WriteLine(num1);

}

Sobrecarga de métodos

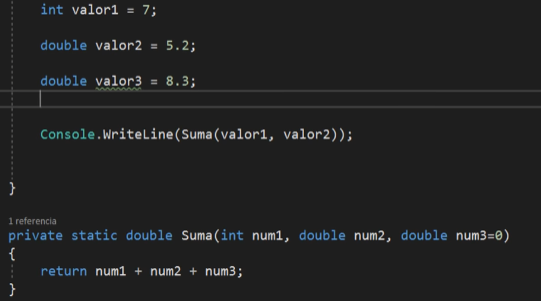
Cuando se llama mas de una vez al mismo método.

Pueden crearse 3 métodos que se llamen igual, si y solo si, reciben parámetros de distinto tipo de datos.



Parámetros opcionales:

Se pone el parámetro y se le asigna un valor por defecto. Así, se le puede pasar o no el parámetro, pisando al parámetro por defecto.



Se escriben siempre después de los parámetros obligatorios.