

Resumen LIBRO.

PALABRAS RESERVADAS EN C# (Pag 96)

Todo identificador comienza con un carácter , en mayúsculas o minúsculas, nunca con numero..

1H2(MAL), hola(Bien),true(mal),resta2-3(bien),\$prueba(bien), 2numeros(Mal). (Pag 97).

Tipos de datos en c# (Pag 99) Diagrama.

Literales Caracteres (Pag 101) - Console.WriteLine("Hola \n Mundo"); Diagrama

Declaración de Variables (Pag 102)

Tipo_de_datos Nombre_de_variable;

Tipo_de_datos Var=valor;

Tipo_de_datos Var1,var2,var3;

Tipo_de_datos Var;

Char letra='a';

Float numero;

Bool condicion123;

Condicion123=true;

Ejemplos en (Pag 104)

Conversores de tipos de datos en c#. (Pag 105).

Operadores Aritméticos (Pag 107).

Operadores Relacionales (Pag 108).

Operadores Lógicos (Pag 109).

Operadores de asignación (Pag 110).

- Constantes y variables.

Operaciones de entrada/Salida de datos en C# (Pag 114).

Sentencias alternativa simple IF (Pag 117)

Sentencias Alternativas Dobles IF ELSE (Pag 118)

Sentencias Alternativa Múltiple 1 IF ELSEIF ELSEIF (Pag 118)

Sentencias Alternativa Múltiple 2 Case ,default (Pag 118)

Sentencias Repetitivas Bucle While (Pag 123)

- Bucle Do While (Pag 125)
- Bucle For (Pag 125)

Sentencias de Salto [Break,Continue y Return] (Pag 127)

- Goto , goto default , switch (Pag 129)

TEMA 4 FUNCIONES

Subprogramas (Pag 141)

Llamadas a una función en c# (Pag 145)

Parametros de una función (Pag 154)

- Paso por valor, y referencia (Pag 155,156)

Funciones de Entrada/Salida (Pag 159)

- Write y WriteLine , Read y Readline , Clear

Funciones numéricas (Pag 159)

Ver desglose documentación .

Funciones de manipulación de cadenas (Pag 160)

Ver desglose documentación.

Funciones de fecha y hora (Pag 160)

Funciones de carácter (Pag 160)

Funciones aleatorias (Pag 161)

Funciones Recursivas (Pag 161)

TEMA 5 ARRAYS

Arrays (Pag 170)

Declaración y creación de Arrays (Pag 171)

Acceso a los datos de Un Array (Pag 173)

Inicializar un Array (Pag 173)

Indices de un Array (Pag 174)

Array de Caracteres en c# (Pag 174)

Arrays como parámetros (Pag 175)

Bucle foreach para la manipulación de arrays (Pag 180)

Ejemplos Programas (Pag 182)

Arrays multidimensionales (Pag 201)

Declaración y creación de arrays multidimensiones (Pag 202)

Inicialización de arrays multidimensiones (Pag 202)

Cadena de caracteres resumen de funciones (Pag 211-212)

Estructuras (Pag 217)

Enumeraciones (Pag 224)

Definir variables, cadenas.

```
int divide = 0;

string cadena,cadena2;

Random alea1 = new Random(); // Genera el aleatorio en variable alea1
Random alea2 = new Random(); // Genera el aleatorio en variable alea2
```

Declaraciones de Matrices.

```
int[] vector1 = new int[15];
int[] vector2 = new int[10];
int[] vector3 = new int[25];
int[,] vector4 = new int[5,5];
int[] sumavector = new int[5];
```

Definición de Estructuras.

```
struct vectorcerdos
{
    public string accion;
    public string codigocerdo;
    public string estado;
    public string peso;
}
```

```
vectorcerdos[] clase = new vectorcerdos[100]; // ESTRUCTURA CERDOS LINEAS CORRECTAS
vectorcerdosok[] clase2 = new vectorcerdosok[100]; // ESTRUCTURA TEMPORAL, PARA COMPROVAR LAS ACCIONES
```

Definición de Ficheros.

```
FileStream flujo = new FileStream("cerdos.txt", FileMode.Open, FileAccess.Read);
StreamReader fichero = new StreamReader(flujo);

while ((linea = fichero.ReadLine()) != null) // Leer el fichero , hasta que no sea el final {}

// Graba las variables y las asigna a la estructura vectorcerdos (clase).

clase[x].accion = partes[0];
clase[x].codigocerdo = partes[1].ToUpper(); // Le pasamos el registro si esta en minusculas, lo cambiamos.. a mayusculas.
clase[x].estado = partes[2].ToUpper();
clase[x].peso = partes[3];
```

Escritura de variables con texto

```
Console.WriteLine("Contenido : " + cadena2); // Indica la linea de Contenido de la matriz 1.

Console.ReadLine();

opcion = int.Parse(Console.ReadLine()); // Leer un texto de una variable para asignarlo.

numero_op3 = System.Int32.Parse(Console.ReadLine());

Console.Clear(); // Borrar Pantalla.

Console.WriteLine(cadena1.ToUpper()); // Convertimos la cadena1 en Mayusculas.
```

Obtener la longitud de la cadena:

```
String cadena = "Frase de prueba";
```

```
Int longitud = cadena.length;
```

Left: que nos devuelve un número x de caracteres a partir de lado izquierdo de una cadena.

Right: que nos devuelve un número x de caracteres a partir del lado derecho de una cadena.

Mid: que nos devuelve una nueva cadena a partir de una cantidad x de caracteres de otra cadena.

Asignación de variables

```
cadena5 = Convert.ToString(vector3[i]); // Convertir variable int a string.
```

```
long1 = cadena1.Length; //Calculamos la longitud de la cadena1.
```

```
a = num3.Next(30, 60); // Almacena número del 30 al 60 en variable a.
```

```
myString = numero_op3.ToString(); // la convierto a tipo string de entero int,
```

```
divide = sumatotal % 2; // módulo de número total de caracteres dividido para 2... si el valor es 0 par si el 1 impar.
```

```
vector1[i] = alea1.Next(-10,10); // La paso el numero aleatorio al vector 1 , entre -10 y 10 / Ejemplo de asignación en vector
```

```
cadena = Convert.ToString(vector1[i]); // Convierto un int en String.. cadena.. para el arrastre de la presentación.
```

SubProgramas:

```
Opcion1(); // Llamada del programa..
```

```
Static void opcion1()
```

```
{
```

```
    // Argumentos...
```

```
}
```

FUNCIONES:

```
resultado = opcion2(cadena1,cadena2); // pasamos los argumentos a la función.
```

```
Static string opcion2(string cadena1,string cadena2)
```

```
{
```

```
    // Argumentos..
```

```
    return rr; // Devuelve el argumento de la operación y lo almacena en la variable resultado..
```

```
}
```

ARRAYS

```

modificarNombre(ref productos[pos]);

static void modificarNombre(ref string nom)
{
    nom = Console.ReadLine();
}

pasararray(ref nivell[0]);
Static void pasararray(ref string nom);
{
    Console.Write(" EL ARRAY ES : " + nombre) // Resultado : "ABECEDARIO"
}

```

Condicionales, bucles.

```

if (devolver_op3 == 0) // Si es la Variable Pasada por Return = 0 -> PAR
{
    Console.WriteLine(" " + cadena_op3 + " PAR");
}

else if (devolver_op3 == 1) // Si es la Variable Pasada Por Return = 1 --> IMPAR
{
    //Console.WriteLine(" IMPAR ");
    Console.WriteLine(" " + cadena_op3 + " IMPAR");
}

while (n <= mayor) Condicional While.
{
    Console.WriteLine(n);
    n++;
}

for (n = 0; n < 10; n++) // Condicional For
{
    acumulacadena = acumulacadena + myString;
}

for (y = 0; y <= 4; y++) // Doble Bucle For
{
    for (x = 0; x <= 4; x++)
    {
        vector4[y, x] = vector3[i];

        cadena7 = Convert.ToString(vector4[y,x]);
        cadena8 = cadena8 + " " + cadena7;

        i = i + 1;
    }
    Console.WriteLine(" " + cadena8);
    cadena7 = "";
    cadena8 = "";
}

```

// Estos dos procedimientos hacen los mismo, se repasan el array de arriba abajo.

```
for (n = 0; n <= 9; n++)
{
    Console.WriteLine(" ARRAY ALMACEN " + nivel11[n]);
}

foreach (string carlos in nivel11) // Carlos tiene que ser cadena, porque el array nivel11 , guarda cadenas "ABECEDARIO"
{
    Console.WriteLine(" ARRAY foreach. : {0}",carlos);
    Console.WriteLine(" ARRAY foreach. 2 : {0}", carlos.Length);
}
```

| | | |
|-------|------------|-------|
| ----- | 19-05-2016 | ----- |
|-------|------------|-------|