1. El método Write() escribe los datos especificados en el flujo de salida estándar.

Código: Console.Write(“¡Hola mundo!”);

Console.Write(“¿Cómo están?”);

En consola: ¡Hola mundo!¿Cómo están?

El método WriteLine() escribe los datos especificados y luego sigue un salto de línea.

Código: Console.WriteLine(“¡Hola mundo!”);

Console.WriteLine(“¿Cómo están?”);

En consola: ¡Hola mundo!

¿Cómo están?

**La diferencia se encuentra en el salto de línea.**

El método ReadKey() obtiene la tecla carácter o función que presione el usuario.

Programa:

Console.Write("Hola");

Console.ReadKey();

Console.Write("Mundo");

1. **Secuencias de escape:**

¿Qué son? Las secuencias de escape son combinaciones de caracteres que cuentan de una barra diagonal inversa seguidas de una letra o de una combinación de dígitos. Se suelen utilizar para para especificar acciones como retornos de carro y movimientos de tabulación en terminales e impresoras.

\n: se utiliza para los saltos de línea.

\t: se utiliza para realizar una tabulación.

\": se utiliza para agregar comillas a las impresiones.

\\: se utiliza para indicarle al motor de expresiones regulares que por más que los caracteres tengan un significado especial en su sintaxis, deseas que los trate como texto normal. Por ejemplo, si deseo escribir “Hola mundo\” debería poner entre las comillas del Write “Hola mundo\\”. **Console.Write(“Hola mundo\\”);**

Programa:

Código: Console.Write("Hola a todos!\n");

Console.Write("Esta es \"mi primera impresión\"\n");

Console.Write("de \t texto en \t .Net\\\\\\\\ :)");

Consola: Hola a todos!

Esta es "mi primera impresión"

de texto en .Net\\\\ :)

Press any key to continue…

1. string st = "c:\\windows\\system";

Console.WriteLine(st);

Para solucionar el problema debo agregar la doble barra a la sintaxis.

1. string st = Console.ReadLine();

if(st.Length == 0){

Console.WriteLine("Hola Mundo");

}

else{

Console.WriteLine("Hola " + st);

}

1. **a)** string st = Console.ReadLine();

if(st == "Juan"){

Console.WriteLine("Hola amigo!");

}

else{

if(st == "Maria"){

Console.WriteLine("Buen día señora");

}

else{

if(st == "Alberto"){

Console.WriteLine("Hola Alberto");

}

}

}

**b)** string st = Console.ReadLine();

switch(st){

case "Juan":

Console.WriteLine("Hola amigo!");

break;

case "Maria":

Console.WriteLine("Buen dia señora");

break;

case "Alberto":

Console.WriteLine("Hola Alberto");

break;

}

1. string st = Console.ReadLine();

while (st.Length != 0){

Console.WriteLine(st.Length);

st = Console.ReadLine();

}

1. La instrucción Console.WriteLine("10".Length); imprime un “3” en terminal.
2. Es válida

string st;

Console.WriteLine(st=Console.ReadLine());

1. string st1 = Console.ReadLine();

if(st1[0] == st1[st1.Length-1]){

Console.WriteLine("Son simétricas");

}

else{

Console.WriteLine("No son simétricas");

}

1. for (int i=1; i<=1000; i++){

if(i%17 == 0){

Console.WriteLine(i);

}

if(i%29 == 0){

Console.WriteLine(i);

}

}

1. Console.WriteLine("10/3 = " + 10 / 3); //SIN RESIDUOS

Console.WriteLine("10.0/3 = " + 10.0 / 3); //CON RESIDUOS

Console.WriteLine("10/3.0 = " + 10 / 3.0); //CON RESIDUOS

int a = 10, b = 3;

Console.WriteLine("Si a y b son variables enteras, si a=10 y b=3"); Console.WriteLine("entonces a/b = " + a / b);

double c = 3;

Console.WriteLine("Si c es una variable double, c=3");

Console.WriteLine("entonces a/c = " + a / c); //CON RESIDUOS

1. Puedo concluir que si divido con dos enteros, el resultado va a ser una división entera (con los residuos fuera del resultado), pero con tener un operando en punto flotante, la división pasa a ser una división con residuos dentro del resultado.
2. El operador “+” entre un String y un dato numérico se encarga de concatenar.

1. string st = Console.ReadLine();

int stint = int.Parse(st);

for (int i=1; i<=stint; i++){

if(stint % i == 0){

Console.WriteLine(i);

}

}

1. La solución es reemplazar el operador AND “&” por el operador AND en cortocircuito “&&”. De esta forma, una vez que analiza que la variable B vale cero no evalúa la segunda condición (la cual es una división entre a y cero).

int a = 30;

int b = 0;

if ((b != 0) && (a/b > 5)) Console.WriteLine(a/b);

1. int b = 12;

int c = 16;

int a = (b<c) ? b : c;

Console.WriteLine(a); //muestra del resultado

1. **int** i = 0;

for (**int** i = 1; i <= 10;){

Console.WriteLine(i++);

}

Declara dos veces la variable **i**. Se soluciona eliminando la segunda declaración.

**int** i = 0;

for (i = 1; i <= 10;){

Console.WriteLine(i++);

}

1. int i = 1;

if (--i == 0){ //I PASA A VALER 0 Y LA CONDICIÓN ES VERDADERA

Console.WriteLine("cero");

}

if (i++ == 0){

Console.WriteLine(i);

Console.WriteLine("cero");

}

Console.WriteLine(i);

Console.ReadKey();