# Temas de C#

## $"" en C#

En C#, la sintaxis $"" se utiliza para crear interpolación de cadenas (string interpolation). Esto significa que puedes insertar directamente variables o expresiones dentro de una cadena sin tener que hacer concatenaciones manuales.  
Ejemplo básico:  
  
int edad = 25;  
string nombre = "Ezequiel";  
string mensaje = $"Hola, {nombre}. Tienes {edad} años.";  
Console.WriteLine(mensaje);  
  
Salida:  
Hola, Ezequiel. Tienes 25 años.  
  
También puedes usar expresiones dentro de {}:  
Ejemplo con expresión:  
  
int a = 5;  
int b = 10;  
Console.WriteLine($"La suma de {a} y {b} es {a + b}.");  
  
Salida:  
La suma de 5 y 10 es 15.

## @"" , \t y \n en C#

@"" permite crear una cadena literal que respete el formato tal cual se escribe, evitando escapar caracteres especiales (como las barras invertidas).  
Ejemplo:  
  
string ruta = @"C:\Users\Ezequiel\Documents\archivo.txt";  
Console.WriteLine(ruta);  
  
Salida:  
C:\Users\Ezequiel\Documents\archivo.txt  
  
\t representa un tabulador horizontal:  
  
string texto = "Nombre\tEdad\tCiudad";  
Console.WriteLine(texto);  
  
Salida:  
Nombre Edad Ciudad  
  
\n representa un salto de línea o nueva línea:  
  
string mensaje = "Hola,\nBienvenido a la comunidad.\n¡Esperamos que disfrutes!";  
Console.WriteLine(mensaje);  
  
Salida:  
Hola,  
Bienvenido a la comunidad.  
¡Esperamos que disfrutes!

## += y ++ en C#

+= es un operador de asignación compuesta que suma un valor a una variable y asigna el resultado a esa misma variable.  
Ejemplo:  
  
int numero = 10;  
numero += 5; // Es equivalente a numero = numero + 5;  
Console.WriteLine(numero); // Imprime: 15  
  
++ es un operador de incremento que suma 1 a la variable.   
Ejemplo:  
  
int contador = 10;  
contador++; // Es equivalente a contador = contador + 1;  
Console.WriteLine(contador); // Imprime: 11

## Clase Random en C#

La clase Random se utiliza para generar números aleatorios.   
Ejemplo básico de generación de número aleatorio:  
  
Random random = new Random();  
int numeroAleatorio = random.Next(1, 101); // Número entre 1 y 100  
Console.WriteLine(numeroAleatorio);  
  
También puedes generar un número decimal aleatorio:  
  
double numeroDecimal = random.NextDouble() \* 10; // Número entre 0.0 y 10.0  
Console.WriteLine(numeroDecimal);

## Clase Math en C#

La clase Math proporciona métodos y constantes para realizar operaciones matemáticas comunes.  
Ejemplos de métodos de Math:  
  
int valor = Math.Abs(-5); // Devuelve 5  
double potencia = Math.Pow(2, 3); // 2^3 = 8  
double raiz = Math.Sqrt(16); // Devuelve 4  
  
Constantes:  
  
Math.PI // Valor de π  
Math.E // Base del logaritmo natural

## Matriz Unidireccional en C#

Una matriz unidireccional almacena una secuencia lineal de elementos del mismo tipo.  
Ejemplo de declaración e inicialización:  
  
int[] numeros = { 10, 20, 30, 40, 50 };

Para recorrer la matriz:  
for (int i = 0; i < numeros.Length; i++) {  
 Console.WriteLine($"Elemento en posición {i}: {numeros[i]}");}

## Asignación de valores a una matriz unidireccional

Puedes asignar valores a una matriz de varias formas:  
Declaración y asignación por separado:  
  
int[] numeros = new int[5];  
numeros[0] = 10;  
numeros[1] = 20;  
numeros[2] = 30;  
numeros[3] = 40;  
numeros[4] = 50;  
  
Con un bucle for:  
  
for (int i = 0; i < numeros.Length; i++) {  
 numeros[i] = (i + 1) \* 10; // Asigna 10, 20, 30, 40, 50  
}

## Uso de la propiedad Length en una matriz unidireccional

La propiedad Length permite obtener el número de elementos en una matriz.  
  
int[] numeros = { 10, 20, 30, 40, 50 };  
Console.WriteLine("La longitud de la matriz es: " + numeros.Length); // Imprime: 5  
  
Para recorrer la matriz:  
  
for (int i = 0; i < numeros.Length; i++) {  
 Console.WriteLine($"Elemento en posición {i}: {numeros[i]}");  
}  
  
Para acceder al último elemento:  
  
int ultimoElemento = numeros[numeros.Length - 1];  
Console.WriteLine("Último elemento: " + ultimoElemento);