

1010100011011000111000011000 100110001010101001111100

11101010010101010 111101010010 1010101110

0101000101010100 0000110001 10011000101 1001111001000

01001010101 11110101001010101

01010100011011000111000011000111 100010101

110101 10101 01111010100 0101010111010100101010100

01000101010100011100001100

1101 00101 0001110101001010 101011101000

011101010010101 1010010101010111101

010100010101010001110001110001100010 100111110010

01010010 11110 10010101010111010100101010

110101001 01000111101010010101

10101001010101000 11110101

01 100010101010001 000 1100110001

01110101001010101000111101010010101010111

1010 0111000011000111

11101010010101010001111010100101 10101110 1001010101000111

010101010001001 1110101010001 1010001110000110001 001100

0001001110000111 101010010101

0101010101011 001110101010001010

0101000101010100011100 001111001100010101010011111001

110 10100101010 11101010010101

PROFUNDIZANDO EN EL TIPO DE DATO STRING EN C#

DOCUMENTACIÓN

¿ Qué es string ?



¿Qué es .NET ?



Immutable Unicode Indexación Comparación Interpolación

</>

```
while (alive) {  
    eat();  
    sleep();  
    code();  
    repeat();  
}
```

•

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("In code we trust");  
    }  
}
```


VARIABLES DE TIPOS ELEMENTALES

Representación en memoria

```
int i = 3;  
Int j;  
j = I + 1;  
boolean b1 = 1 == j;  
boolean b2 = i < j;
```

i	3
j	4
b1	false
b2	true

- Los operadores relacionales comparan el contenido de los bloques de memoria.


```
var userList = GetUsers();

var activeUsersLinq = userList.Where

var activeUsers = new List<User>();
foreach (var user in userList)
{
    if (user.IsActive)
    {
        activeUsers.Add(user);
    }
}
```

Métodos

Length	Trim()
ToUpper()	StartsWith():
ToLower()	EndsWith():
IndexOf()	
LastIndexOf()	
Replace()	
Substring()	

Propiedades

Length
IsEmpty
Chars
IsReadOnly

```
public class Something
{
    public Something()
    {
        Number = 10;
    }
    public int Number {
```



DelftStack