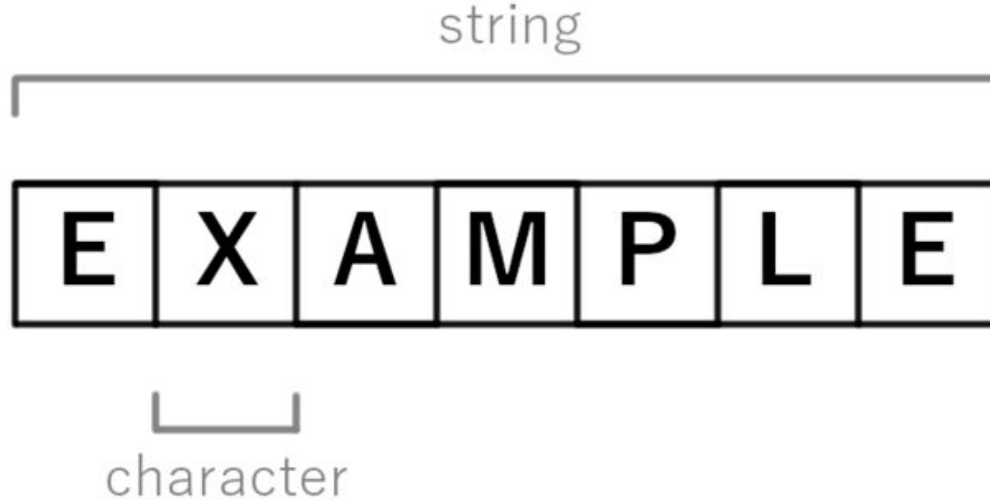



# Strings & Regex



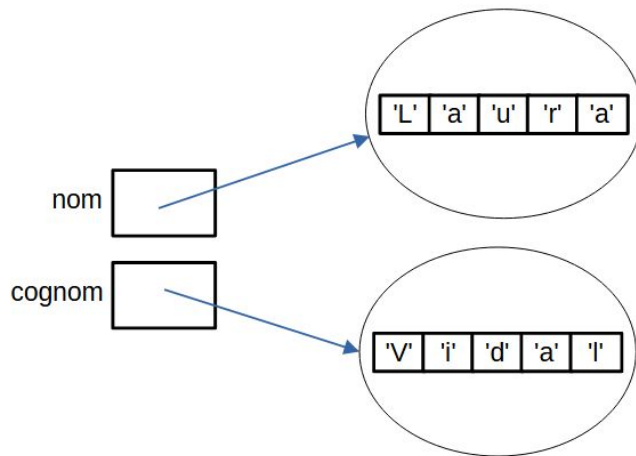
# Tipus String

En c# un String és una classe de la llibreria System que permet emmagatzemar una cadena de caràcters.

```
string nom = "Laura";  
string cognom = new string(['V','i','d','a','l']);  
Console.WriteLine($"{nom} {cognom}");
```

 Consola de depuración de Microsoft Visual Studio

Laura Vidal



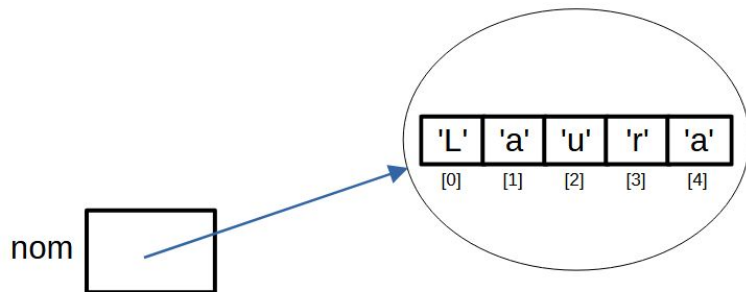
# Index

Els objecte de tipus String permeten ser accedit a través d'un índex. Per tant, és possible entendre un String com una cadena de caràcters.

```
string nom = "Laura";  
char inicial = nom[0];  
Console.WriteLine($"{nom} comença pel caràcter {inicial}");
```

C:\> Consola de depuración de Microsoft Visual Studio

Laura comença pel caràcter L



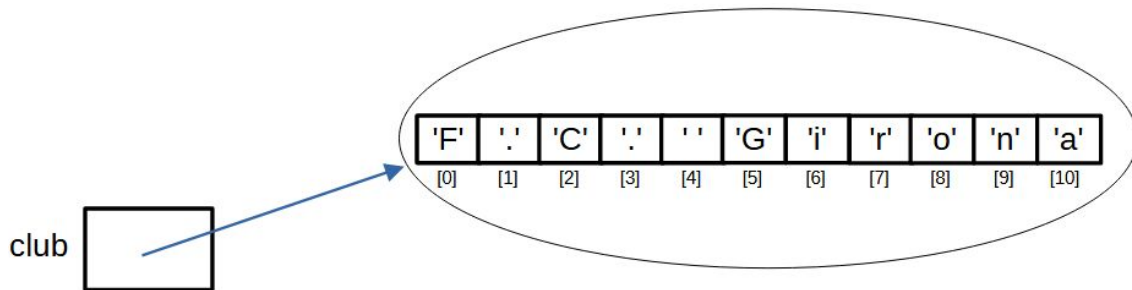
# Length

Els objectes de tipus String tenen la propietat Length que permet obtenir el nombre de caràcters de l'objecte.

```
string club = "F.C. Girona";  
Console.WriteLine(club.Length);
```

Consola de depuración de Microsoft Visual Studio


11



# Length

Utilitzant la propietat Length podem recòrrer tots els caràcter d'un objecte de tipus String. En l'exemple analitzem quantes vegades apareix el caràcter 'a' en una cadena de text.

```
string frase = "Bla bla bla bla bla bla bla bla bla bla";  
int countA = 0;  
for(int i = 0; i < frase.Length; i++)  
{  
    if (frase[i] == 'a' || frase[i] == 'A') countA++;  
}  
  
Console.WriteLine($"LA FRASE CONTÉ {countA} VEGADES EL CARÀCTER 'A'");
```


 Consola de depuración de Microsoft Visual Studio

```
LA FRASE CONTÉ 10 VEGADES EL CARÀCTER 'A'
```

# foreach

Alternativament, també es pot utilitzar l'estructura foreach per a recòrrer els caràcters d'un objecte de tipus String. El foreach utilitza el concepte d'**iterador**.

```
string frase = "Bla bla bla bla bla bla bla bla bla bla";  
int countA = 0;  
foreach (char car in frase)  
{  
    if (car == 'a' || car == 'A') countA++;  
}  
  
Console.WriteLine($"LA FRASE CONTÉ {countA} VEGADES EL CARÀCTER 'A'");
```

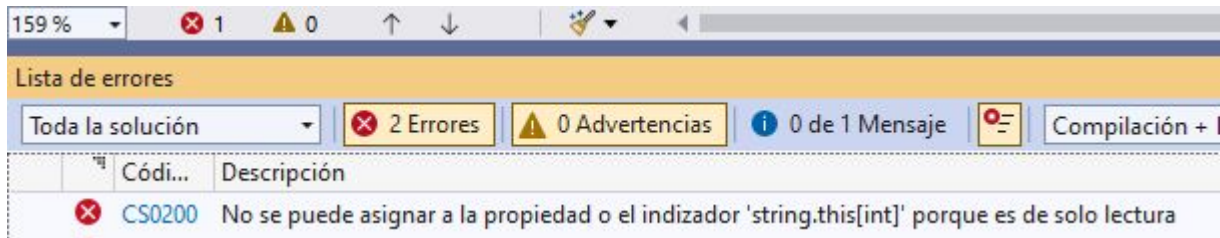
 Consola de depuración de Microsoft Visual Studio

```
LA FRASE CONTÉ 10 VEGADES EL CARÀCTER 'A'
```

# Immutable

El tipus String no es pot comparar amb altres tipus d'objectes doncs té la característica de ser immutable. Això significa que en ser creat un objecte de tipus String no es podrà modificar.

```
string nom = "Laura";  
nom[0] = 'R';
```

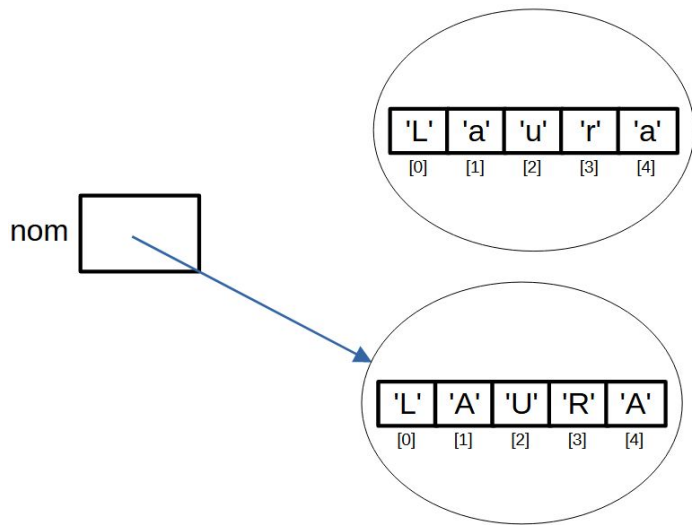


# Immutable

Els mètodes de la classe String que modifiquen una cadena de text, en realitat creen un nou objecte de tipus String.

En l'exemple, s'observa que el mètode ToUpper genera una nou objecte amb tots els caràcters en majúscules i s'assigna aquest nou objecte a la variable nom.

```
string nom = "Laura";  
nom = nom.ToUpper();
```

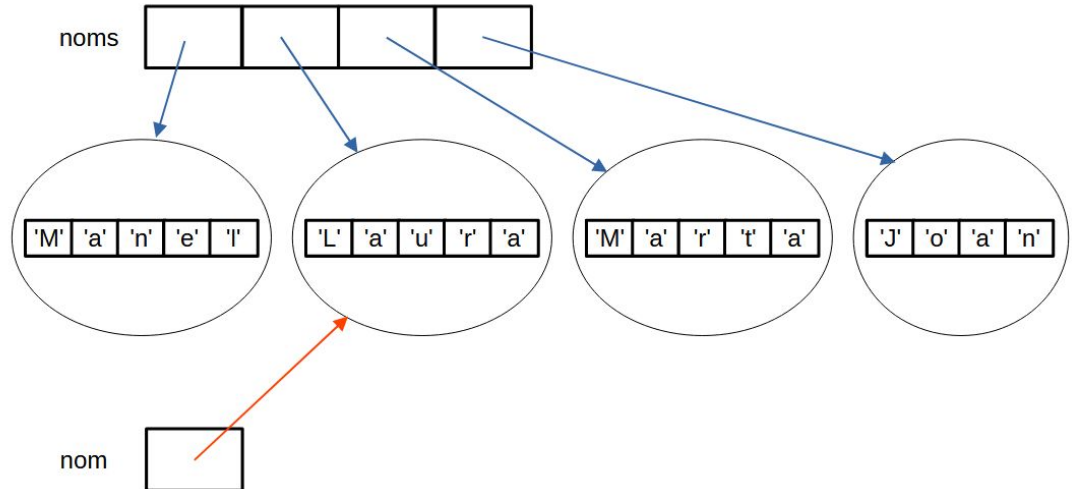




# Assignació

L'assignació d'objectes de tipus String funciona de la mateixa manera que en qualsevol altre objecte de tipus Employee o Producte.

```
string[] noms = ["Manel", "Laura", "Marta", "Joan"];  
string nom = noms[1];
```



# Suma

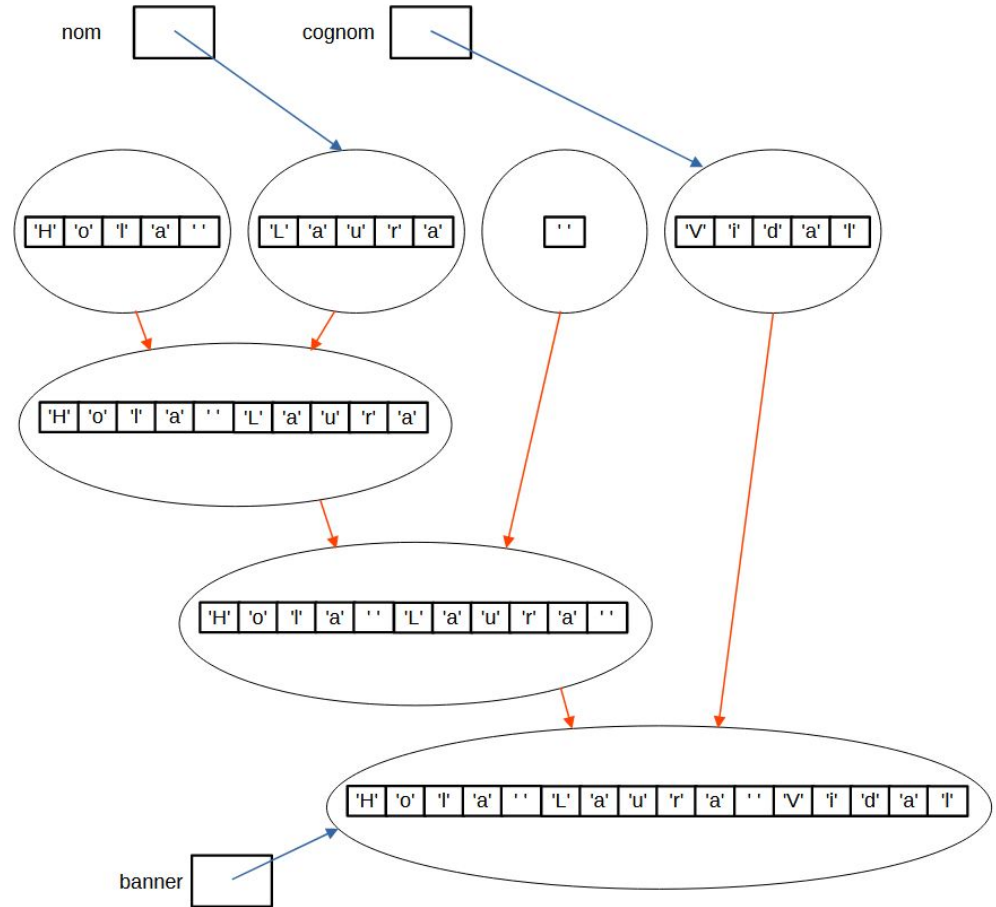
L'operador de la suma dels objectes de tipus String ha estat modificat respecte al funcionament estàndard d'un objecte genèric.

L'operador de suma està sobrecarregat i realitza la concatenació de dos objectes.

```
string nom = "Laura";  
string cognom = "Vidal";  
string banner = "Hola " + nom + " " + cognom;
```

# Suma

Utilitzar l'operador suma en moltes concatenacions pot suposar la creació de molts objectes de tipus String, doncs cada operador suma implica concatenar dos objectes de tipus String i crear un tercer objecte de tipus String.



# StringBuilder

En els casos que s'hagin de concatenar molts objectes de tipus String, el millor és utilitzar la classe StringBuilder.

```
StringBuilder sb = new StringBuilder();  
sb.Append("Hola");  
sb.Append(nom).Append(" ");  
sb.Append(cognom);  
  
banner = sb.ToString();
```

<https://www.shekhali.com/difference-between-string-and-stringbuilder-in-csharp/>

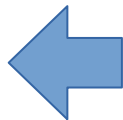
# Equals

Els objectes de la classe String disposen del mètode Equals per a comparar dos cadenes de text.

La classe String **sobreescriu** els operadors de igual (==) i de diferent (!=) de manera que el comportament sigui que mateix que el del mètode Equals.

```
string str1 = "Hola";  
string str2 = "hola";
```

```
bool iguals = str1.Equals(str2);  
bool iguals2 = str1 == str2;
```



Les dues comparacions d'igualtat són equivalents.

# Comparar strings

Els objectes de tipus String no es poden comparar utilitzant els operadors <, >, <=, >=.

```
string str1 = "Terra";  
string str2 = "Aigua";  
  
Console.WriteLine(str1>str2);
```

❌ CS0019 El operador '>' no se puede aplicar a operandos del tipo 'string' y 'string'

# Comparar strings

Per a comparar dos objectes de tipus String s'ha d'utilitzar el mètode CompareTo.

```
string str1 = "Terra";  
string str2 = "Aigua";  
  
Console.WriteLine(str1.CompareTo(str2));
```

El mètode CompareTo retorna el següent:

- Si **str1 > str2** retorna **1**
- Si **str1 < str2** retorna **-1**
- Si **str1 == str2** retorna **0**

# Mètodes de la classe String

La classe String té un conjunt de mètodes que es poden utilitzar. Alguns del mètodes més habituals són:

Mètode	Descripció	Retorn
IndexOf	Troba l'índex de la primera aparició d'un caràcter o subcadena especificat. <a href="https://www.dotnetperls.com/indexof">https://www.dotnetperls.com/indexof</a>	int
StartsWith / EndsWith	Comprova si una cadena comença amb una subcadena especificada. <a href="https://www.dotnetperls.com/startswith">https://www.dotnetperls.com/startswith</a>	bool
Contains	Comprova si una cadena conté una subcadena especificada. <a href="https://www.dotnetperls.com/contains">https://www.dotnetperls.com/contains</a>	bool



# Mètodes de la classe String

Mètode	Descripció	Retorn
ToUpper / ToLower	Converteix una cadena a majúscules / minúscules. <a href="https://www.dotnetperls.com/tolower">https://www.dotnetperls.com/tolower</a>	String
Split	Divideix una cadena en un array basant-se en un delimitador especificat. <a href="https://www.dotnetperls.com/split">https://www.dotnetperls.com/split</a>	String[]
Join	Combina un array de cadenes en una sola cadena amb un delimitador especificat. <a href="https://www.dotnetperls.com/string-join">https://www.dotnetperls.com/string-join</a>	String

# Mètodes de la classe String

<b>Mètode</b>	<b>Descripció</b>	<b>Retorn</b>
Trim	Elimina els espais en blanc inicials i finals. <a href="https://www.dotnetperls.com/contains">https://www.dotnetperls.com/contains</a>	string
Replace	Substitueix les ocurrències d'una subcadena per una altra subcadena. <a href="https://www.dotnetperls.com/replace">https://www.dotnetperls.com/replace</a>	string
Substr	Extreu una subcadena de la cadena. <a href="https://www.dotnetperls.com/substring">https://www.dotnetperls.com/substring</a>	string

# Regex

Les Regex (Expressions Regulars) permeten valorar si una cadena de text compleix un cert patró.

<https://www.dotnetperls.com/regex>

Regex per a comprovar que una cadena conté només digits enters.

```
string strNumero = "12345";  
string patro = @"^\d+$";  
if (Regex.IsMatch(strNumero, patro))  
    Console.WriteLine("NUMERO POSITIU CORRECTE");  
else  
    Console.WriteLine("NÚMERO POSITIU INCORRECTE");
```

# Regex

Regex per a comprovar que una cadena conté el format correcte d'una data.

```
string strData = "12/09/2010";  
string patro = @"^\d{2}/\d{2}/\d{4}$";  
if (Regex.IsMatch(strData, patro))  
    Console.WriteLine("FORMAT DE DATA CORRECTE");  
else  
    Console.WriteLine("FORMAT DE DATA INCORRECTE");
```

# Regex

Regex per a comprovar que una cadena conté el format correcte d'un correu electrònic.

```
string strEmail = "u1500023@institutmontilivi.cat";  
string patro = @"^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$";  
if (Regex.IsMatch(strEmail, patro))  
    Console.WriteLine("FORMAT DE EMAIL CORRECTE");  
else  
    Console.WriteLine("FORMAT DE EMAIL INCORRECTE");
```

# Exercicis de Strings

1) Donat el següent codi, obtenir el primer i l'últim enter de la cadena.

```
String linia = "34;44;12;11;99;55;5;0;11;34;32";  
int firstNumber;  
int lastNumber;
```

2) Donat el següent codi, obtenir la mitjana de tots els números de la cadena.

```
String linia = "34;44;12;11;99;55;5;0;11;34;32";  
int suma;  
int contador;  
double mitjana;
```

# Exercicis de Strings

3) Donat el següent codi, obtenir el nom, el mes i l'any de la cadena de text.

```
String linia = "Lionel;Messi;24/06/1987;67;1,7";  
string nom;  
int mes;  
int any;
```

4) Donat el següent codi, obtenir el inicial del nom.

```
String linia = "Lionel;Messi;24/06/1987;67;1,7";  
char inicialNom;
```

5) Donat el següent codi, obtenir la inicial del cognom.

```
String linia = "Lionel;Messi;24/06/1987;67;1,7";  
int inicialCognom;
```

# Exercicis de Strings

6) Fes un programa en C# que demani un email i que validi que aquest email és correcte.

- En cas que el correu no tingui un format correcte, tornat a demanar a l'usuari que torni a demanar el correu.
- El programa ha d'eliminar els possibles espais en blanc en cas que n'hi hagi.

7) Donat el següent codi, analitza el que fa i executa'l. Modifica el codi utilitzant la classe StringBuilder.

```
string str = "";  
for(int i=0;i<10;i=i+) str=str+'.';  
str = str.Substring(0,str.Length-1);  
Console.WriteLine(str);
```



# Exercicis de Strings

8) Modifica el mètode ToString de la classe Inventari, de manera que utilitzi la classe StringBuilder per a concatenar els diferents productes.