



Strings

A.K.A. Cadena de caracteres.

Algo1-Mendez

Nicolás Riedel

Antes de arrancar
hagamos memoria sobre
el video ...

¿Para que usamos
Strings en C ?



Segun Wikipedia, un string es una sequencia ordenada de caracteres pertenecientes a un alfabeto.

En nuestro caso particular, el alfabeto son los posibles valores que tiene el tipo de dato Char.



Dejamos una pequeña definición

Hoy en día, gran parte de los lenguajes de programación modernos tienen el tipo de dato **String** como primitivo.

Pero en **C** no queda otra que agarrar la pala....



Bueno, dejemos de
dar vueltas

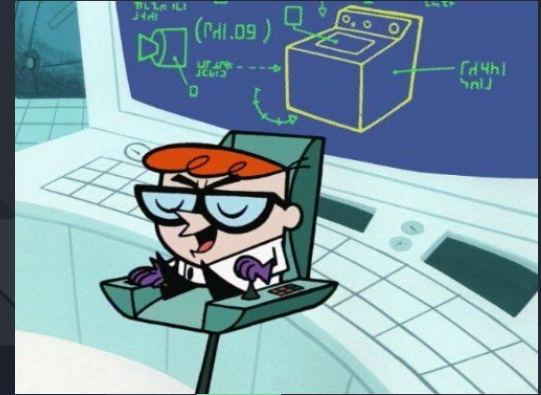


Como representamos un
String en el lenguaje C?



Basicamente lo pensamos como un array de caracteres.

string	array											
	0	1	2	3	4	5	6	7	8	9	10	11
	char	char	char	char	char	char	char	char	char	char	char	char
	'D'	'e'	'x'	't'	'e'	'r'	'_'	'c'	'a'	'p'	'o'	'\0'



Donde nos reservamos el “caracter nulo” para señalar el fin del string.



Bueno, vamo a codea

Como declaramos un "string"...

```
3  #define TAM 50
4
5  int main () {
6      char string [TAM];
7      return 0;
8  }
```

Recordar las buenas practicas....

Tambien podemos asignarle un valor en el momento que lo declaramos.

```
3  #define TAM 50
4
5  int main () {
6      char string [TAM] = "Hola a todes!!";
7      return 0;
8  }
```

Peeeeeeeeeeerooo

Ojo al tejo....



No es correcto ni posible
hacer esto :

```
3  #define TAM 50
4
5  int main () {
6      char string [TAM];
7      string = "Hola a todes!!";
8      return 0;
9  }
```



Si el string ya se encuentra declarado no
es posible asignarle un valor de esta
forma...

Llegan a nuestra ayuda
las bibliotecas estándar...


Particularmente :
“String.h”



Particularmente
vamos a usar estas
funciones:

- strlen
- strcmp
- strcpy
- strcat

Para usarla debemos incluir



```
#include <string.h>
```



Strlen

Recibe un string por parametro y nos devuelve su tamaño.

```
1  #include <string.h>
2  #include <stdio.h>
3
4  #define TAM 50
5
6  int main () {
7      char string [TAM] = "cinco";
8      int tam;
9      tam = strlen(string);
10     printf("%i\n", tam);
11     return 0;
12 }
```

Este programa imprime 5



Strcmp

Recibe dos strings y los compara. Si el primero es menor devuelve un numero negativo, y si el segundo es el menor entonces devuelve un nuemero positivo.

Caso que sean iguales, la funcion devuelve 0 .

```
1  #include <string.h>
2  #include <stdio.h>
3
4  #define TAM 50
5
6  int main () {
7
8      char string0 [TAM] = "Arbol";
9      char string1 [TAM] = "Vaca";
10     printf("%i\n", strcmp(string0, string1) );
11     printf("%i\n", strcmp(string1, string0) );
12     printf("%i\n", strcmp(string0, string0) );
13     return 0;
14 }
```

Este programa imprime:

- * Entero Negativo.
- * Entero Positivo.
- * Cero.



Strcpy

Recibe dos strings, copia el contenido del segundo en el primero.

```
5
6  int main () {
7
8      char origen [TAM] = "Christopher Nolan";
9      char destino [TAM];
10     strcpy(destino, origen);
11     printf("%s \n", destino);
12     return 0;
13 }
```

Este programa imprime : "Christopher Nolan"



Otro ejemplo de Strcpy

```
6  int main () {  
7      char destino [TAM];  
8      strcpy(destino, "Christopher Nolan");  
9      printf("%s \n", destino);  
10     return 0;  
11 }
```

Este es un ejemplo de como asignar un valor luego de declarar un string.



Strcat

Recibe 2 strings y los concatena, usando como destino el primero.

```
6  int main () {  
7      char string1 [TAM] = "Hola Soy ";  
8      char string2 [TAM] = "Nicolas Riedel";  
9      strcat(string1, string2);  
10     printf("%s \n", string1);  
11     return 0;  
12 }
```

La salida de este string es :
Hola Soy Nicolas Riedel



Ahora sigamos con un ejemplo





El villano **Manuel Cangrejo** se divierte mucho burlandose de sus subordinados.

Su burla es medio primitiva... el simplemente repite lo que dicen sus colaboradores pero intercambiando todas las vocales por la “i”.



Hacer un programa que reciba por teclado un mensaje y lo muestre por pantalla “Manulizado”.