

Club de Algoritmia  
ESFM



# Strings

Perro

Facebook.com?

Mi nombre es

$3 + 6/8 = 21$

Hola soy un string

# Declaración



Utiliza el tipo string seguido del nombre que quieras

```
string nombre;
```

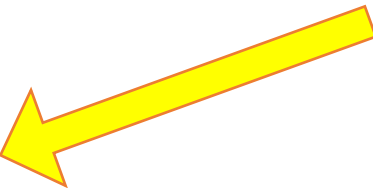
Puedes declarar varios al mismo tiempo

```
string palabra, cadena, s;
```

# ¿Cómo leer strings?

```
string palabra;  
cin >> palabra;
```

Lectura de una sola  
cadena **sin espacios**



```
string linea;  
getline(cin, linea);
```

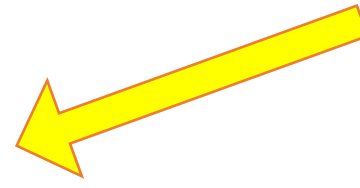
Lectura de  
toda una línea\*



## ¿Y la impresión?

```
cout << cadena;
```

Podemos utilizar **cout**  
directamente para imprimir un  
string completo en la consola



\*La función **getline(,)** se encarga de leer toda una línea de la consola, recibe de parámetros el objeto de lectura (el poderoso **cin**) y el **string** donde se guardará



# Hora de un ejemplo

# Operadores ready-to-use

==	Igual que	Comparar si dos strings son iguales
!=	Diferente de	Comparar si dos strings son distintos
> (>=)	Mayor (o igual) que	Mayor <b>lexicográfico*</b>
< (<=)	Menor (o igual) que	Menor <b>lexicográfico*</b>
+	<b>Concatenación</b>	Une dos string en uno solo
=	<b>Asignación</b>	Asignar el contenido a un string

\*Consulta el código ASCII para mas información <https://elcodigoascii.com.ar>

## Caracteres ASCII de control

00	NULL	(carácter nulo)
01	SOH	(inicio encabezado)
02	STX	(inicio texto)
03	ETX	(fin de texto)
04	EOT	(fin transmisión)
05	ENQ	(consulta)
06	ACK	(reconocimiento)
07	BEL	(timbre)
08	BS	(retroceso)
09	HT	(tab horizontal)
10	LF	(nueva línea)
11	VT	(tab vertical)
12	FF	(nueva página)
13	CR	(retorno de carro)
14	SO	(desplaza afuera)
15	SI	(desplaza adentro)
16	DLE	(esc.vínculo datos)
17	DC1	(control disp. 1)
18	DC2	(control disp. 2)
19	DC3	(control disp. 3)
20	DC4	(control disp. 4)
21	NAK	(conf. negativa)
22	SYN	(inactividad sínc)
23	ETB	(fin bloque trans)
24	CAN	(cancelar)
25	EM	(fin del medio)
26	SUB	(sustitución)
27	ESC	(escape)
28	FS	(sep. archivos)
29	GS	(sep. grupos)
30	RS	(sep. registros)
31	US	(sep. unidades)
127	DEL	(suprimir)

## Caracteres ASCII imprimibles

32	espacio	64	@	96	`
33	!	65	A	97	a
34	"	66	B	98	b
35	#	67	C	99	c
36	\$	68	D	100	d
37	%	69	E	101	e
38	&	70	F	102	f
39	'	71	G	103	g
40	(	72	H	104	h
41	)	73	I	105	i
42	*	74	J	106	j
43	+	75	K	107	k
44	,	76	L	108	l
45	-	77	M	109	m
46	.	78	N	110	n
47	/	79	O	111	o
48	0	80	P	112	p
49	1	81	Q	113	q
50	2	82	R	114	r
51	3	83	S	115	s
52	4	84	T	116	t
53	5	85	U	117	u
54	6	86	V	118	v
55	7	87	W	119	w
56	8	88	X	120	x
57	9	89	Y	121	y
58	:	90	Z	122	z
59	;	91	[	123	{
60	<	92	\	124	
61	=	93	]	125	}
62	>	94	^	126	~
63	?	95	_		

## ASCII extendido (Página de código 437)

128	Ç	160	á	192	Ł	224	Ó
129	ü	161	í	193	ł	225	õ
130	é	162	ó	194	Ł	226	Ô
131	â	163	ú	195	ł	227	Ò
132	ä	164	ñ	196	—	228	ö
133	à	165	Ñ	197	†	229	Õ
134	å	166	ª	198	ä	230	µ
135	ç	167	º	199	Ä	231	þ
136	ê	168	¿	200	Ł	232	ƀ
137	ë	169	®	201	Œ	233	Ú
138	è	170	¬	202	ℒ	234	Û
139	ï	171	½	203	℥	235	Ü
140	î	172	¼	204	℥	236	Ý
141	ì	173	¡	205	=	237	Ÿ
142	Ä	174	«	206	≠	238	ˉ
143	Å	175	»	207	□	239	˘
144	É	176	⋮	208	ø	240	≡
145	æ	177	⋮	209	Ð	241	±
146	Æ	178	⋮	210	È	242	≡
147	ô	179	⋮	211	Ê	243	¾
148	ö	180	⋮	212	Ë	244	¶
149	ò	181	À	213	Ì	245	§
150	û	182	Â	214	Í	246	÷
151	ù	183	Ã	215	Î	247	ˆ
152	ÿ	184	©	216	Ï	248	˚
153	Ö	185	℥	217	Ĵ	249	˙
154	Ü	186	℥	218	Œ	250	˙
155	ø	187	℥	219	█	251	˙
156	£	188	℥	220	█	252	˙
157	Ø	189	¢	221	⋮	253	˙
158	×	190	¥	222	⋮	254	■
159	f	191	γ	223	█	255	nbsp



# Hora de un ejemplo



¿Por qué se dice cadenas de caracteres?

D A F Z

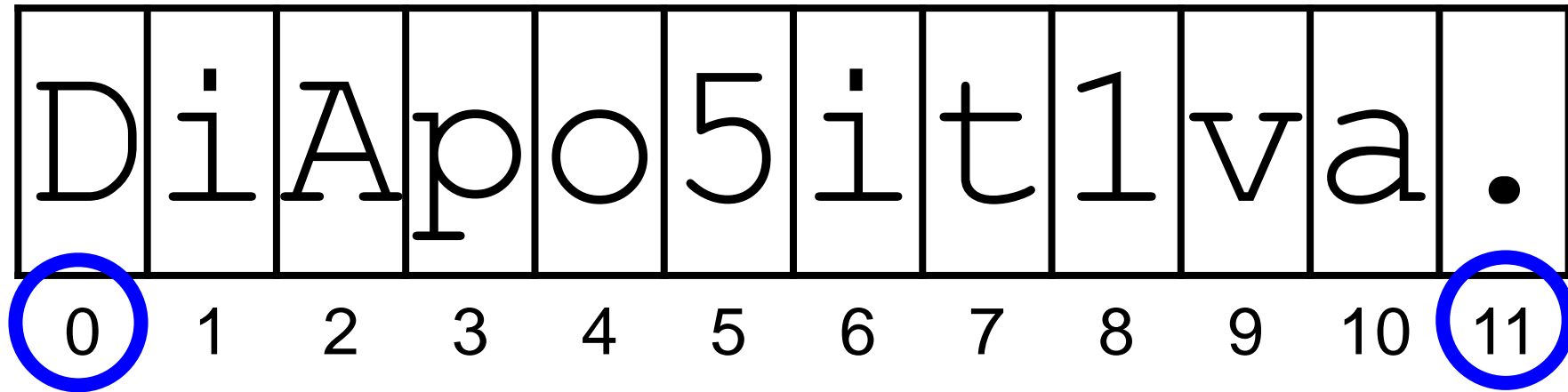
5 3 0 1

DiApo5it1va.

! ? \$ .

a e r t

Los string se almacenan como *arreglos de caracteres* enumerados siempre desde 0 hasta  $n-1$  con  $n$  su longitud



**Longitud: 12**

Podemos acceder a los caracteres por su índice con el operador `[]` y asignar otro carácter con el operador `=`

D	i	A	p	o	5	i	t	1	v	a	.
0	1	2	3	4	5	6	7	8	9	10	11

$s[0] = D$

$s[9] = v$

$s[5] = 5$

$s[11] = .$



# Funciones para string\*

<code>.size()</code>	Longitud de string	Te devuelve un entero, el número de caracteres del string
<code>.begin()</code> y <code>.end()</code>	Apuntadores al inicio y fin	Devuelve un apuntador al inicio y fin del string respectivamente**
<code>.append()</code> o <code>+=</code>	Concatenación	Concatenar al final otro string o caracter
<code>.empty()</code>	Menor (o igual) que	Devuelve verdadero si el string está vacío, falso caso contrario
<code>.find()</code>	<b>Búsqueda en string</b>	Devuelve el índice de la primera ocurrencia de un caracter en un string
<code>count(,,)</code>	<b>Conteo de caracter</b>	Cuenta número de apariciones de un caracter en el string

\*Referencia completa en <http://www.cplusplus.com/reference/string/string/>

\*\*Muy útil en `sort(,)`

# Hora de un ejemplo

# Conversiones a tipos de dato

## *fx* Functions

### Convert from strings

<b>stoi</b> <small>C++11</small>	Convert string to integer (function template )
<b>stol</b> <small>C++11</small>	Convert string to long int (function template )
<b>stoul</b> <small>C++11</small>	Convert string to unsigned integer (function template )
<b>stoll</b> <small>C++11</small>	Convert string to long long (function template )
<b>stoull</b> <small>C++11</small>	Convert string to unsigned long long (function template )
<b>stof</b> <small>C++11</small>	Convert string to float (function template )
<b>stod</b> <small>C++11</small>	Convert string to double (function template )
<b>stold</b> <small>C++</small>	Convert string to long double (function template )

### Convert to strings

<b>to_string</b> <small>C++11</small>	Convert numerical value to string (function )
<b>to_wstring</b> <small>C++11</small>	Convert numerical value to wide string (function )

### Range access

<b>begin</b> <small>C++11</small>	Iterator to beginning (function template )
<b>end</b> <small>C++11</small>	Iterator to end (function template )



# Hora de un ejemplo

# Palíndromos

sagas  
rasar  
soros  
sallas  
radar  
saeteas  
sopapos  
sotos  
kayak  
selles  
orero  
razar  
socos  
ateta  
alolaalela  
sebesnarran  
rapar  
sananas  
abanaba  
apocopa  
sayas  
abajaba  
pas  
sedes  
sanasaviva  
ososo  
seceso  
arara  
samas  
sacas  
nomon  
recocer  
sollos  
dalla  
sucus  
salas  
drayar  
rodador  
adamada  
abañaba  
rajarababa  
arenera  
sumus  
rever

## Caso 1: Longitud par

z	o	n	a	o	z
---	---	---	---	---	---

0 1 2 3 4 5

s	o	l	l	o	s
---	---	---	---	---	---

0 1 2 3 4 5

c	l	u	b	a	l	g	g	l	a	b	u	l	c
---	---	---	---	---	---	---	---	---	---	---	---	---	---

0 1 2 3 4 5 6 7 8 9 10 11 12 13



## Caso 2: Longitud impar

a	n	i	t	a	n	a
---	---	---	---	---	---	---

0 1 2 3 4 5 6

o	y	i	z	o
---	---	---	---	---

0 1 2 3 4

a	n	i	t	a	l	a	v	a	l	a	t	i	n	a
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14

# Hora de programar