

# Repàs i continuació del dia anterior

<https://docs.hektorprofe.net/python/metodos-de-las-colecciones/metodos-de-las-cadenas/>

## Exercici Intro:

```
text = 'EL FOCO\n @elfocoes \n 5 de des. de 2020 \n Natalia  
Lacunza en #Los40MusicAwards2020: \n \n  
        "Nunca colaboraría con Kidd Keo, me parece bastante  
gilipollas. No me escondo" \n @natalialacunza'
```

```
#1.- Passa tot el text a minúscules  
#2.- Canvia els punts per blancs  
#3.- Mira quans cops el text conté la paraula = "gilipollas"  
#3.- Mira quans cops el text conté la paraula = "natalia"
```

Consulta a :

[https://www.w3schools.com/python/python\\_ref\\_string.asp](https://www.w3schools.com/python/python_ref_string.asp)

Les funcions replace, count. Si fas un find de “saxo” què retorna ?

## 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	ƒ
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	Ý
142	Ä	174	«	206	≠	238	~
143	Å	175	»	207	¤	239	'
144	É	176	⋮	208	ð	240	≡
145	æ	177	⋮	209	Ð	241	±
146	Æ	178	⋮	210	Ê	242	≡
147	ô	179	⋮	211	Ë	243	¾
148	ö	180	†	212	È	244	¶
149	ò	181	À	213	Ì	245	§
150	û	182	Â	214	Í	246	÷
151	ù	183	À	215	Î	247	ˆ
152	ÿ	184	©	216	Ï	248	˚
153	Ö	185	¶	217	Ɔ	249	ˆ
154	Ü	186	¶	218	┐	250	ˆ
155	ø	187	┐	219	■	251	ˆ
156	£	188	┐	220	■	252	ˆ
157	Ø	189	¢	221	⋮	253	ˆ
158	×	190	¥	222	⋮	254	■
159	f	191	γ	223	■	255	nbsp

## P06. Quin és el resultat d'aquestes operacions:

a) Calcula

```
txt = "Castaña, Asunción, María, José"  
x = txt.upper()  
print(x)
```

b) Calcula

```
txt = "CATALÁN DE CANCIÓN TULÚN!";  
mytable = txt.maketrans("ÁÉÍÓÚ", "AEIOU");  
print(txt.translate(mytable));
```

c) Estudia aquestes instruccions. Com es calcula una majúscula a partir d'una minúscula ?

```
print(ord("A"), ord("a"))  
print(ord("B"), ord("b"))  
print(ord("C"), ord("c"))  
print(ord("Ñ"), ord("ñ"))  
print(ord("Á"), ord("á"))  
print(ord("É"), ord("é"))
```

d) Calcula

```
txt = "excursiones, televisiones, macarrones"  
x = txt.replace("es", "s")  
print(x)
```



# Tipus de dades : Llista

## Llistes ordenades de variables

### 2. Llistes i matrius

#### 2.1. Què són i com funcionen

- Pràctiques P04, P05

### 3. Introduir dades i mostrar per pantalla

#### 3.1. Comanda input()

- Pràctiques P06, P07

#### 3.2. Mostrar dades amb format

##### 3.2.1. Simple : fString


##### 3.2.2. Funció de cadena : format

##### 3.2.3. Avançat: Formateig de xifres

- Pràctiques P08, P09



## 2.1. Què són i com funcionen



```
In [1]: numeros = [1,2,3,4]

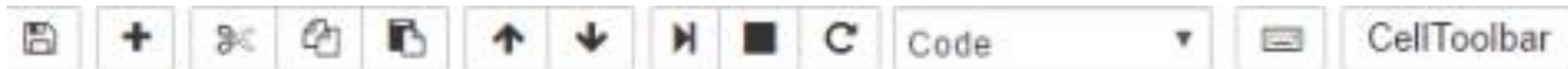
In [2]: datos = [4,"Una cadena",-15,3.14,"Otra cadena"]

In [3]: datos[0]
Out[3]: 4

In [4]: datos[-1]
Out[4]: 'Otra cadena'

In [5]: datos[-2:]
Out[5]: [3.14, 'Otra cadena']

In [6]: numeros + [5,6,7,8]
Out[6]: [1, 2, 3, 4, 5, 6, 7, 8]
```



```
In [7]: pares = [0,2,4,5,8,10]
```

```
In [8]: pares[3]= 6
```

Diferencia amb les cadenes

```
In [9]: pares
```

```
Out[9]: [0, 2, 4, 6, 8, 10]
```

```
In [10]: pares.append(12)
```

```
In [11]: pares
```

```
Out[11]: [0, 2, 4, 6, 8, 10, 12]
```

```
In [12]: pares.append(7*2)
```

```
In [13]: pares
```

```
Out[13]: [0, 2, 4, 6, 8, 10, 12, 14]
```



```
Out[13]: [0, 2, 4, 6, 8, 10, 12, 14]
```

```
In [14]: letras = ['a','b','c','d','e','f']
```

```
In [15]: letras[:3]
```

```
Out[15]: ['a', 'b', 'c']
```

```
In [16]: letras[:3] = ['A','B','C']
```

```
In [17]: letras
```

```
Out[17]: ['A', 'B', 'C', 'd', 'e', 'f']
```

```
In [18]: letras[:3] = []
```

```
In [19]: letras
```

```
Out[19]: ['d', 'e', 'f']
```

```
In [21]: letras = []
```

```
In [22]: letras
```

```
Out[22]: []
```

```
In [23]: len(letras)
```

```
Out[23]: 0
```

```
In [24]: len(pares)
```

```
Out[24]: 8
```

```
In [25]: a = [1,2,3]
         b = [4,5,6]
         c = [7,8,9]
         r = [a,b,c]
```

Es crea una llista de llistes o

```
In [26]: r
```

matriu

```
Out[26]: [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

```
In [27]: r[0]
```

```
Out[27]: [1, 2, 3]
```

```
In [28]: r[-1]
```

```
Out[28]: [7, 8, 9]
```

```
In [29]: r[0][0]
```

```
Out[29]: 1
```

```
In [30]: r[1][1]
```

```
Out[30]: 5
```

```
In [31]: r[2][2]
```

```
Out[31]: 9
```

```
In [32]: r[-1][-1]
```

```
Out[32]: 9
```



## # P04. Pràctica de llistes unidimensionals:

```
# Tenint la variable llista = [a,e,i,o,u], calcula  
# llista[1:4]  
# llista[:5]  
# llista[-4:]  
# llista[2]  
# llista[:]  
# llista[::-1]
```

<b>J</b>		<b>O</b>
	<b>J</b>	<b>O</b>

## # P05. Pràctica de llistes bidimensionals:

```
# Tenim aquest taulell de 3 en ratlla.  
# J |   | O  
#   | J | O  
#   |   |  
# Crea una taulell bi-dimensional amb les jugades.  
# Imprimeix el taulell  
# Demana una jugada de Jugador J(linia, columna)  
# Coloca la jugada al taulell  
# torna a imprimir el taulell
```

# Ajuda

```
# Tenim aquest taulell de 3 en ratlla.
# J |   | O
#   | J | O
#   |   |
# Crea una taulell bi-dimensional amb les jugades.
# Imprimeix el taulell
# Demana una jugada de Jugador J(linia, columna)
# Coloca la jugada al taulell
# torna a imprimir el taulell

def imprimir (taulell) :
    print (taulell[0][0], "|", taulell[0][1], "|", taulell[0][2])
    print (taulell[1][0], "|", taulell[1][1], "|", taulell[1][2])
    print (taulell[2][0], "|", taulell[2][1], "|", taulell[2][2])

#----- INICI


taulell = [["J", " ", "O"], [" ", "J", "O"], [" ", " ", " "]]
imprimir (taulell)
```



### 3. Introduir dades i mostrar per pantalla

- Introduir cadenes : Input = entra
- Convertir a valors numèrics

# 3.1. Comanda `input()`



```
In [1]: valor = input()
        algo

In [2]: valor
Out[2]: 'algo'

In [3]: valor = input()
        100

In [4]: valor
Out[4]: '100'

In [*]: valor = input("Introduce un valor: ")
        Introduce un valor: |

In [ ]:
```

Out[4]: '100'

```
In [5]: valor = input("Introduce un valor: ")
```

Introduce un valor: 100

```
In [6]: valor + 100
```

```
-----  
TypeError                                 Traceback (most recent call last)  
<ipython-input-6-5071d551e583> in <module>()  
----> 1 valor + 100
```

TypeError: Can't convert 'int' object to str implicitly

**Lo que tenemos que hacer es convertir este valor a un entero**

```
In [7]: valor = input("Introduce un número entero: ")
```

Introduce un número entero: 500

```
In [8]: valor = int(valor)
```

```
In [9]: valor
```

Out[9]: 500

```
In [10]: valor + 1000
```

Out[10]: 1500

```
In [11]: valor = input("Introduce un número entero: ")
```

Introduce un número entero: 10.50

```
In [14]: valor = float(valor)
```

```
In [15]: 10 + valor
```

```
Out[15]: 20.5
```

```
In [16]: valor
```

```
Out[16]: 10.5
```

---

```
In [ ]: valor = float( input("Introduce un número decimal o entero: ") )
```

---

# P06. Pràctica de demanar dades

Demana un número de mes (calendari) i mostra el seu nom:  
hauries de tenir una llista amb els noms ["gener", "febrer"..etc]

*Pista :*

*Pensa que el mes que has demanat ha de servir d'index de la llista de mesos, per tant hauràs de convertir el valor demanat per assegurar que és un número.*

# P07. Pràctica de demanar dades (avançat)

1. Demana les unitats comprades al 2018 i les del 2019 i 2020.  
Suma les 3 i mostra el missatge : “has comprat un total de : “ i mostra la suma de les compres de tots els anys.
2. Sobre l'exercici anterior :
  - Calcula la *mitjana aritmètica* i mostra-la
  - Com pots comparar les dades del 2021 amb la mitja dels anys anteriors per veure si vas bé? Tingues en compte que del 2021 només tenim dades fins al final de juny.Aproxima com pugis l'indicador i mostra'l.

Nota : *mitjana aritmètica* = *media* en castellà



## 3.2. Mostrar dades amb format

Les formes de donar format han anat variant en les diferents versions de Python, us mostro una de simple i després la més usada.

### 3.2.1. Simple : fString

f" algun text {variable} i més"

```
## declarem variables
nom_empresa = "Tantar"
tipus_empresa = "Educació"

## tanca amb {} el nom de la variable que vols mostrar
print(f"{nom_empresa} és una empresa de {tipus_empresa}.")
```

Tantar és una empresa de Educació.

## 3.2.2. Funció de cadena : format

```
In [1]: v = "otro texto"
        n = 10
        print("Un texto",v,"y un número",n)

Un texto otro texto y un número 10
```

Per especificar es fan servir **claudàtors** en lloc de la variable,  
i després la funció **.format** amb totes les variables separades per comes

```
In [2]: c = "Un texto {} y un número {}".format(v,n)
        c
```

```
Out[2]: 'Un texto otro texto y un número 10'
```

```
In [5]: print( "Un texto '{0}' y un número '{1}'".format(v,n) )

Un texto 'otro texto' y un número '10'
```

```
In [6]: print( "Un texto '{1}' y un número '{0}'".format(v,n) )

Un texto '10' y un número 'otro texto'
```

```
In [7]: print( "Un texto '{texto}' y un número '{numero}'".format(texto=v,numero=n) )
```

Un texto 'otro texto' y un número '10'

```
In [9]: print( "Un texto '{v}' y un número '{n}'".format(n=n,v=v) )
```

Un texto 'otro texto' y un número '10'

```
In [10]: print("{v},{v},{v}".format(v=v))
```

otro texto,otro texto,otro texto

```
In [14]: print( "{:>30}".format("palabra") ) # Alineamiento a la derecha en 30 caracteres
```

palabra

```
In [15]: print( "{:30}".format("palabra") ) # Alineamiento a la izquierda en 30 caracteres
```

palabra

```
In [16]: print( "{:^30}".format("palabra") ) # Alineamiento al centro en 30 caracteres
```

```
palabra
```

```
In [17]: print( "{:.3}".format("palabra") ) # Truncamiento a 3 caracteres
```

```
pal
```

```
In [20]: print( "{:>30.3}".format("palabra") ) # Alineamiento a la derecha en 30  
caracteres con truncamiento de 3
```

```
pal
```

## 3.2.3. Avançat: Formateig de xifres

```
In [23]: # Formateo de números enteros, rellenos con espacios
print("{}".format(10))
print("{}".format(100))
print("{}".format(1000))
```

10  
100  
1000

```
In [24]: # Formateo de números enteros, rellenos con espacios
print("{:4d}".format(10))
print("{:4d}".format(100))
print("{:4d}".format(1000))
```

10  
100  
1000

Reserva cuatro posiciones enteras alineadas a la derecha.

```
# Formateo de números en
print("{:04d}".format(10))
print("{:04d}".format(100))
print("{:04d}".format(1000))
```

0010  
0100  
1000

Reserva cuatro posiciones enteras alineadas a la derecha y rellenas con ceros.

```
In [27]: # Formateo de números flotantes, rellenos con espacios
print("{}".format(3.1415926))
```

3.1415926

```
In [29]: # Formateo de números flotantes, r
print("{:.3f}".format(3.1415926))
```

3.142

Reserva 3 posiciones  
decimales alineadas a la  
derecha.

```
In [30]: # Formateo de números flotantes, rellenos con espacios
print("{:.3f}".format(3.1415926))
print("{:.3f}".format(153.21))
```

3.142

153.210

No especifica las posiciones totales y queda  
descuadrado al hacer columnas de cifras  
distintas.

```
In [31]: # Formateo de números flotantes, rellenos con espacios
print("{:7.3f}".format(3.1415926))
print("{:7.3f}".format(153.21))
```

3.142  
153.210

Especifica 7 posiciones totales, de las cuales 3  
son decimales. Y alinear a la derecha.

```
# Formateo de números flotantes, rel
print("{:07.3f}".format(3.1415926))
print("{:07.3f}".format(153.21))
```

003.142

153.210

```
print("{:7.3f}".format(3.1416))  
print("{:7.3f}".format(-3.1416))
```

```
3.142  
-3.142
```

```
print("{:-7.3f}".format(3.1416))  
print("{:-7.3f}".format(-3.1416))
```

```
3.142  
-3.142
```

```
print("{:+7.3f}".format(3.1416))  
print("{:+7.3f}".format(-3.1416))
```

```
+3.142  
-3.142
```

Si se omite el signo en el formato o se incluye un signo negativo, las cifras positivas aparecen sin signo pero reservando la posición.

Si en el formato se incluye un signo positivo, todas las cifras aparecen con signo.

# P08. Práctica de format

1. Formatea los siguientes valores para mostrar el resultado indicado:

- "Hola Mundo" → Alineado a la derecha en 20 caracteres
- "Hola Mundo" → Truncamiento en el cuarto carácter (índice 3)
- "Hola Mundo" → Alineamiento al centro en 20 caracteres con truncamiento en el segundo carácter (índice 1)
- 150 → Formateo a 5 números enteros rellenos con ceros
- 7887 → Formateo a 7 números enteros rellenos con espacios
- 20.02 → Formateo a 3 números enteros y 3 números decimales



# P09. Pràctica de format

Programa els càlculs al quadern, copiant els enunciats com a comentari

- **Demana** 3 xifres (poden ser senceres o amb decimals, positives o negatives).
- Imagina que són entrades o sortides a un compte bancari.
- Imprimeix-les una assota de l'altre de forma que quedin ben alineades considerant el punt decimal, i mostra només el signe negatiu.
- Al costat de cada xifra imprimeix el saldo, però formatat amb zeros i que mostri sempre el signe.

**Ajuda : Per fer proves, assigna valors fixes a les variables.**

-3.44	-00003.44
1200.00	+01196.56
67.00	+01263.56

# Algunes solucions

```
llista = ['a','e','i','o','u']  
#P04 - calcula llista[1:4], llista[:5], llista[-4:],  
llista[2], llista[:], llista[::-1]
```

```
print(llista[1:4])  
print(llista[:5])  
print(llista[-4:])  
print(llista[2])  
print(llista[:])  
print(llista[::-1])
```

```
['e', 'i', 'o']  
['a', 'e', 'i', 'o', 'u']  
['e', 'i', 'o', 'u']  
i  
['a', 'e', 'i', 'o', 'u']  
['u', 'o', 'i', 'e', 'a']
```

## # P05. Pràctica de llistes bidimensionals:

```
#
# Tenim aquest taulell de 3 en ratlla.
# J |   | O
#   | J | O
#   |   |
# Crea una taulell bi-dimensional amb la jugada.
# Imprimeix el taulell
# Demana una jugada de les X (linia, columna)
# Coloca la jugada i torna a imprimir el taulell

def imprimir (taulell) :
    print (taulell[0][0], "|", taulell[0][1], "|", taulell[0][2])
    print (taulell[1][0], "|", taulell[1][1], "|", taulell[1][2])
    print (taulell[2][0], "|", taulell[2][1], "|", taulell[2][2])

#----- INICI
taulell = [["J", " ", "O"], [" ", "J", "O"], [" ", " ", " "]]
imprimir (taulell)

lin = int(input ("entri linea (1-3): "))
col = int(input ("entri columna (1-3): "))

taulell[lin-1][col-1] = "J"
imprimir (taulell)
```

## #P06.-----

#Demana un número de mes (calendari) i mostra el seu  
#nom: hauries de tenir una llista amb els noms ["gener",  
#"febrer"..etc]

```
nom_mes = ["gener", "febrer", "març", "abril", "maig", "juny",  
"juliol", "agost", "setembre","octubre","novembre","desembre"]
```

```
mes = int(input("Número de Mes : ")) -1  
print ("El mes es " + nom_mes [mes])
```