

Ejercicio 1 – Tema 7			1 Pto
Cuál de los siguientes diccionarios tiene todas sus claves no numéricas (no son tipo int, float ni complex)?			
A	b=float(False) {'2/25':'verano', b:False, 1:'verano'}		A
B	b=False {True:'V', 1:'uno', '4':'cuatro', b:'F'}		B
C	b=float(False) {'enero':0, '2':66, '5/25':3.45, ('10',):b}	X	C
D	b=float(False) {b:('1', '0'), '44':('3','5')}		D

Ejercicio 2 – Tema 71 Pto

¿Cuál de las siguientes instrucciones debería usar (junto a otras) para **generar** un gráfico como el siguiente con matplotlib? Los datos a representar están en el DataFrame **resfríos**:

	mes	humedad	cantidad
0	enero	60	2
1	febrero	75	23
2	marzo	79	32
3	abril	68	31
4	mayo	62	45
5	junio	55	61

Relación Resfríos Humnedad Ambiente

% Humedad promedio	Cant Diagnosticos Resfrío
55	61
60	2
62	45
68	31
75	23
79	32

A	plt.grid('Relación Resfríos Humnedad Ambiente')		A
B	ax.scatter(resfrios['humedad'], resfrios['cantidad'])	X	B
C	ax.plot(meses, humedad)		C
D	ax.pie(resfrio['cantidad'], labels=resfrio['humedad'])		D

Ejercicio 3 – Tema 7			1 Pto																																
Dado el Dataframe <i>menu</i> :																																			
<table><tr><th></th><th>día</th><th>plato</th><th>calorías</th></tr><tr><td>0</td><td>lun o mier</td><td>guiso</td><td>750.0</td></tr><tr><td>1</td><td>lun o mier</td><td>milanesas</td><td>560.0</td></tr><tr><td>2</td><td>mar o jue</td><td>pasta</td><td>NaN</td></tr><tr><td>3</td><td>mar o jue</td><td>pastel</td><td>NaN</td></tr><tr><td>4</td><td>viernes</td><td>guiso</td><td>800.0</td></tr><tr><td>5</td><td>sáb o dom</td><td>asado</td><td>750.0</td></tr><tr><td>6</td><td>sáb o dom</td><td>pasta</td><td>750.0</td></tr></table>					día	plato	calorías	0	lun o mier	guiso	750.0	1	lun o mier	milanesas	560.0	2	mar o jue	pasta	NaN	3	mar o jue	pastel	NaN	4	viernes	guiso	800.0	5	sáb o dom	asado	750.0	6	sáb o dom	pasta	750.0
	día	plato	calorías																																
0	lun o mier	guiso	750.0																																
1	lun o mier	milanesas	560.0																																
2	mar o jue	pasta	NaN																																
3	mar o jue	pastel	NaN																																
4	viernes	guiso	800.0																																
5	sáb o dom	asado	750.0																																
6	sáb o dom	pasta	750.0																																
¿Cuál de las siguientes instrucciones debería usar para producir la siguiente salida?																																			
<table><tr><th></th><th>día</th><th>plato</th><th>calorías</th></tr><tr><td>0</td><td>lun o mier</td><td>guiso</td><td>750.0</td></tr><tr><td>1</td><td>lun o mier</td><td>milanesas</td><td>560.0</td></tr><tr><td>4</td><td>viernes</td><td>guiso</td><td>800.0</td></tr><tr><td>5</td><td>sáb o dom</td><td>asado</td><td>750.0</td></tr><tr><td>6</td><td>sáb o dom</td><td>pasta</td><td>750.0</td></tr></table>					día	plato	calorías	0	lun o mier	guiso	750.0	1	lun o mier	milanesas	560.0	4	viernes	guiso	800.0	5	sáb o dom	asado	750.0	6	sáb o dom	pasta	750.0								
	día	plato	calorías																																
0	lun o mier	guiso	750.0																																
1	lun o mier	milanesas	560.0																																
4	viernes	guiso	800.0																																
5	sáb o dom	asado	750.0																																
6	sáb o dom	pasta	750.0																																
A	menu[menu['calorías'].between(7, 100)]		A																																
B	menu.sort_values(by=['día','calorías'])		B																																
C	menu[(~menu['calorías'].isnull())]		X C																																
D	menu[['calorías','plato']]		D																																

Ejercicio 4 – Tema 7			1 Pto
Dado los archivos vtas-suc1.txt y vtas-suc2.txt que contienen las ventas mensuales del primer semestre de cada sucursal:			
vtas-suc1.txt		vtas-suc2.txt	
ene,69.5		ene,19.5	
feb,33.0		feb,43.0	
mar,20.0		mar,0.0	
abr,10.0		abr,110.0	
may,0		may,11	
jun,0		jun,66	
¿Con cuál conjunto de modos de apertura funcionará adecuadamente el siguiente programa?			
<pre>suc1=open('vtas-suc1.txt', ...) # modo 1? todas1=suc1.readlines() suc1.close() suc2=open('vtas-suc2.txt', ...) # modo 2? todas2=suc2.readlines() suc2.close() semestre=open('vtas-semester.txt', ...) # modo 3? for i in range(len(todas1)): lin='suc1,'+todas1[i] semestre.write(lin) lin='suc2,'+todas2[i] semestre.write(lin) semestre.close()</pre>			
Al finalizar quedará 1 archivo adicional: vtas-semester.txt , que contendrá todas las líneas de los archivos originales ordenados por mes e intercaladas las sucursales:			
<pre>suc1,ene,69.5 suc2,ene,19.5 suc1,feb,33.0 suc2,feb,43.0 suc1,mar,20.0 suc2,mar,0.0 suc1,abr,10.0 suc2,abr,110.0 suc1,may,0 suc2,may,11 suc1,jun,0 suc2,jun,66</pre>			
A	modo 1 'w' modo 2 'r' modo 3 'w'		A
B	modo 1 'r+' modo 2 'r' modo 3 'w'	X	B
C	modo 1 'a' modo 2 'w' modo 3 'r'		C
D	modo 1 'a' modo 2 'a' modo 3 'a'		D

Ejercicio 5 – Tema 7		1 Pto	
¿Cuál programa realiza un ingreso validado de un número, que debe ser alguno de los siguientes: -14, 21, 5, 66, 7, 89, 104, -3, -206? ✓ No se debe aceptar un número que no sea alguno de esos ✓ El programa no debe dar fallo si se ingresa un dato que no sea número ✓ El programa debe insistir en la solicitud hasta obtener lo esperado Nota: Si se le envía a un print() una lista con un* delante muestra los elementos sin corchetes, separados por espacio Ej: print(*[1,2,3,4]) -> 1 2 3 4			
A	<pre>validos=(-14,21,5,66,7,89,104,-3,-206) print('Ingresá uno de estos números') print(*validos) while not False: try: num=int(input()) if num in validos: print(num) except ValueError: print('Debe ser un número')</pre>		A
B	<pre>validos=(-14,21,5,66,7,89,104,-3,-206) print('Ingresá uno de estos números') print(*validos) for i in range(len(validos)): num=int(input()) try: if num==validos[i]: print(num) else: print('No es válido') except ValueError: print('Debe ser un número')</pre>		B
C	<pre>validos=(-14,21,5,66,7,89,104,-3,-206) print('Ingresá uno de estos números') print(*validos) ok=False while not ok: try: num=int(input()) if num in validos: ok=True else: print('No es válido') except ValueError: print('Debe ser un número') print(num)</pre>	X	C
D	<pre>validos=(-14,21,5,66,7,89,104,-3,-206) print('Ingresá uno de estos números') print(*validos) try: num=int(input()) if num not in validos: print(num) else: print('No es válido') except ValueError: print('Debe ser un número')</pre>		D

Ejercicio 6 – Tema 7				2 Ptos	
Dado el Dataframe <i>menu</i> :					
	día	plato	calorías		
0	lun o mier	guiso	750.0		
1	lun o mier	milanesas	560.0		
2	mar o jue	pasta	NaN		
3	mar o jue	pastel	NaN		
4	viernes	guiso	800.0		
5	sáb o dom	asado	750.0		
6	sáb o dom	pasta	750.0		
¿Qué salida produce la siguiente instrucción?					
<pre>print(menu.sort_values (by=['día' , 'calorías'] ,ascending=[True,False]))</pre>					
A	calorías	plato			A
	0 750.0	guiso			
	1 560.0	milanesas			
	2 NaN	pasta			
	3 NaN	pastel			
	4 800.0	guiso			
	5 750.0	asado			
	6 750.0	pasta			
B	día	plato	calorías	X	B
	0 lun o mier	guiso	750.0		
	1 lun o mier	milanesas	560.0		
	2 mar o jue	pasta	NaN		
	3 mar o jue	pastel	NaN		
	5 sáb o dom	asado	750.0		
	6 sáb o dom	pasta	750.0		
	4 viernes	guiso	800.0		
C	día				C
	lun o mier	560.0			
	mar o jue	NaN			
	sáb o dom	750.0			
	viernes	800.0			
D	plato				D
	guiso	2			
	pasta	2			
	milanesas	1			
	pastel	1			
	asado	1			

Ejercicio 7 – Tema 7			2 Ptos
Dado el siguiente programa: <pre>def calc1(num): return (num,-num) def calc2(num): return (num*2>0,num) def calc3(num): return (num//2,num%2) numeros=[-16,1407,23,-200] # línea de código faltante print(nuevaLista)</pre> ¿Cuál debería ser la línea de código faltante para que la salida sea la siguiente? <pre>[(-8, 0), (703, 1), (11, 1), (-100, 0)]</pre>			
A	nuevaLista=list(map(calc3,numeros))	X	A
B	nuevaLista=list(reversed(numeros))		B
C	nuevaLista=list(map(calc2,numeros))		C
D	nuevaLista=list(map(calc1,numeros))		D

Ejercicio 8 – Tema 7		2 Ptos	
Dado el siguiente programa:			
<pre>frutas=... #?? puestos={'DP':'Don Pedro','CR':'Cistóbal Rey', 'EB':'El Brocal','E2':'ER II', 'VS':'Valle Sarrundis','LB':'La Bolsa'} tipo=input('Qué fruta debe reponer? ') print('Lista de Puestos del Mercado que venden',tipo.upper()) for clave in frutas: if tipo.lower() in frutas[clave]: print(puestos[clave])</pre>			
¿Cuál de las opciones de la estructura frutas funcionará correctamente para producir la siguiente salida?			
<pre>Qué fruta debe reponer? naranja ombligo Lista de Puestos del Mercado que venden NARANJA OMBLIGO ER II La Bolsa</pre>			
A	<pre>frutas={1:['durazno chato','durazno blanco', 'pelón amarillo'], 4:['ciruela blanca','ciruela remolacha', 'gotita de miel'], 0:['pelón amarillo','pelón blanco', 'gotita de miel'], 2:['manzana romme','manzana deliciosa'], 3:['naranja jugo','naranja ombligo'], 5:['mandarina nova','naranja ombligo']}</pre>		A
B	<pre>frutas=['durazno chato','durazno blanco','pelón amarillo', 'ciruela blanca','ciruela remolacha', 'gotita de miel','pelón amarillo','pelón blanco', 'gotita de miel','manzana romme', 'manzana deliciosa','naranja jugo', 'naranja ombligo','mandarina nova', 'naranja ombligo']</pre>		B
C	<pre>frutas={1:'durazno chato',1:'durazno blanco', 1:'pelón amarillo',4:'ciruela blanca', 4:'ciruela remolacha',4:'gotita de miel', 0:'pelón amarillo',0:'pelón blanco', 0:'gotita de miel',2:'manzana romme', 2:'manzana deliciosa',3:'naranja jugo', 3:'naranja ombligo',5:'mandarina nova', 5:'naranja ombligo'}</pre>		C
D	<pre>frutas={'CR':['durazno chato','durazno blanco', 'pelón amarillo'], 'VS':['ciruela blanca','ciruela remolacha', 'gotita de miel'], 'DP':['pelón amarillo','pelón blanco', 'gotita de miel'], 'EB':['manzana romme','manzana deliciosa'], 'E2':['naranja jugo','naranja ombligo'], 'LB':['mandarina nova','naranja ombligo']}</pre>	X	D

Ejercicio 9 – Tema 7			2 Ptos
¿Cuál es la salida del siguiente programa? <pre>def edita(txt): return txt+'s' def patron(txt): cant=0 for voc in 'AEIOU': cant+=txt.count(voc) return len(txt)/2>cant frutas=['CIRUELA', 'FRUTILLA','MANZANA','NARANJA'] salida=list(filter(patron,frutas)) mostrar=list(map(edita,salida)) print(mostrar)</pre>			
A	['FRUTILLAs', 'MANZANAs', 'NARANJAs']	X	A
B	['MANZANA', 'NARANJA']		B
C	['CIRUELAs', 'MANZANAs']		C
D	['ciruela', 'frutilla']		D