

Buenos Aires, 10 de julio de 2023

TEMA 2
Hoja 1 de 7

APELLIDO:	CALIFICACIÓN:
NOMBRE:	
DNI (registrado en SIU Guaraní):	
E-MAIL:	
TEL:	DOCENTE (nombre y apellido):
AULA:	

Duración del examen: 1:30h. Completar con **letra clara, mayúscula e imprenta**. El examen consta de 13 preguntas de opción múltiple. Cada pregunta tiene una y sólo una respuesta correcta.

Las respuestas deben completarse en la siguiente matriz:

Opción	EJ. 1	EJ. 2	EJ. 3	EJ. 4	EJ. 5	EJ. 6	EJ. 7	EJ. 8	EJ. 9	EJ. 10	EJ. 11	EJ. 12	EJ. 13
1				X				X	X				
2						X					X		
3		X	X		X		X					X	
4	X									X			X

¡ATENCIÓN! Las respuestas sólo se considerarán válidas si se encuentran en la matriz. De haber diferencias entre la opción seleccionada en el ejercicio y en la matriz, se considerará como válida la de la matriz.

Ejercicio 0102		
Si a=33, cuál de las siguientes condiciones da verdadero (True)		
1.	a>=10 and a<=25 and not(a==47)	1.
2.	a>=10 and a<=25 or a==47	2.
3.	a<=10 and a>=25 and a==47	3.
4.	a<=10 or not(a>=25) or a!=47	X 4.

Ejercicio 0202		
¿Cuál de los siguientes ciclos itera exactamente 5 (cinco) veces?		
1.	for vueltas in range(1,5): print('Girando...')	1.
2.	vueltas=0 while vueltas<4: print('Girando...')	2.
3.	vueltas=1 while vueltas<=5: vueltas+=1 print('Girando...')	X 3.
4.	for vueltas in range(14,7,-2): print('Girando...')	4.

Ejercicio 0302			
¿Cuál sería una versión correcta del programa para validar el ingreso de un mes del año en número (1 al 12)?			
1.	<pre> pedir=True while pedir: try: mes=int(input('Mes: ')) if mes<=1 and mes>=12: pedir=False else: print('Seleccionar el mes entre 1 y 12') except ValueError: print('Sólo números enteros') </pre>		1.
2.	<pre> pedir=True while pedir: try: mes=int(input('Mes: ')) if mes>=1 or mes<=12: pedir=False else: print('Seleccionar el mes entre 1 y 12') except: print('Sólo números enteros') </pre>		2.
3.	<pre> pedir=True while pedir: try: mes=int(input('Mes: ')) if mes>=1 and mes<=12: pedir=False else: print('Seleccionar el mes entre 1 y 12') except: print('Sólo números enteros') </pre>	X	3.
4.	<pre> pedir=False while pedir: try: mes=int(input('Mes: ')) if mes in range(1,13): pedir=False else: print('Seleccionar el mes entre 1 y 12') except ValueError: print('Sólo números enteros') </pre>		4.

Ejercicio 0402			
Dado el siguiente bloque de código:			
<pre> lis=[1,23,4,9,66,108] primeros=[] segundos=[] for n in lis: if n%2 == 0: primeros.append(n) else: segundos.append(n) primeros += segundos </pre> <p>¿Cómo queda la lista primeros?</p>			
1.	[4, 66, 108, 1, 23, 9]	X	1.
2.	[1, 23, 9, 4, 66, 108]		2.
3.	[4, 66, 108]		3.
4.	[1, 23, 9]		4.

Ejercicio 0502			
<p>Dado el siguiente programa:</p> <pre>def elige(dicci): print('Seleccioná') for op in dicci: print(op, dicci[op]) seleccion=input() if seleccion not in dicci: seleccion='1' return seleccion menu={'1':'Pasta','2':'Pollo','3':'Carne','4':'Veggie'} opc=elige(menu) print('Elegiste',menu[opc])</pre> <p>¿Qué hace la función elige()?</p>			
1.	<p>Muestra el siguiente menú de opciones:</p> <pre>Seleccioná 1, Pasta 2, Pollo 3, Carne 4, Veggie</pre> <p>Y permite que el usuario sólo seleccione una opción válida. Luego se la entrega al programa principal</p>	X	1.
2.	<p>Muestra el siguiente menú de opciones:</p> <pre>Seleccioná 1 Pasta 2 Pollo 3 Carne 4 Veggie</pre> <p>Y permite que el usuario sólo seleccione una opción válida. Luego se la entrega al programa principal. El programa puede fallar.</p>	X	2.
3.	<p>Muestra el siguiente menú de opciones:</p> <pre>Seleccioná 1 Pasta 2 Pollo 3 Carne 4 Veggie</pre> <p>Y permite que el usuario seleccione sólo una opción. Luego se la entrega al programa principal sin validar. El programa nunca falla.</p>	X	3.
4.	<p>Muestra el siguiente menú de opciones:</p> <pre>Seleccioná 4 Veggie 3 Carne 2 Pollo 1 Pasta</pre> <p>Y permite que el usuario seleccione sólo una opción. Si la misma es entre 1 y 4, se la entrega al programa principal; si no, termina la ejecución y no imprime nada.</p>	X	4.

Ejercicio 0602			
<p>¿Qué salida por pantalla produce el siguiente programa?</p> <pre> notas={'Almirón':[4,5,7,8], 'Álvarez':[10,6,9], 'Céliz':[8,8,7], 'Gómez':[5,9,5,4]} for alumno in notas: notas[alumno].sort() for alumno in notas: print(alumno,notas[alumno]) </pre>			
1.	Almirón [8, 7, 5, 4] Álvarez [10, 9, 6] Céliz [8, 8, 7] Gómez [9, 5, 5, 4]		1.
2.	Almirón [4, 5, 7, 8] Álvarez [6, 9, 10] Céliz [7, 8, 8] Gómez [4, 5, 5, 9]	X	2.
3.	Almirón [4, 5, 7, 8] Álvarez [10, 6, 9] Céliz [8, 8, 7] Gómez [5, 9, 5, 4]		3.
4.	{'Almirón': [4, 5, 7, 8], 'Álvarez': [6, 9, 10], 'Céliz': [7, 8, 8], 'Gómez': [4, 5, 5, 9]}		4.

Ejercicio 0702			
<p>Dado el siguiente programa</p> <pre> lista_1 = [1,2,3,4,5,6] lista_2 = ["a","b","c","d","f","g"] letras = "" for i in range (0,4): j = lista_1[i] letras += lista_2[j] print(letras) </pre> <p>¿Cuál es el resultado de su ejecución ?</p>			
1.	Da un error de acceso a indice: list index out of range		1.
2.	Se imprime: abcd		2.
3.	Se imprime: bcdf	X	3.
4.	Se imprime: cdfg		4.

Ejercicio 0802		
<p>Una empresa de videojuegos cuenta con un archivo que contiene por cada fila: código de uno de sus juegos, código de jugador, nivel alcanzado. El path del archivo está guardado en la variable file_name.</p> <p>Por ejemplo: 567994;29345;Platinado 567900;29345;Platinado 587721;29346;Completo 567994;29346;Platinado</p> <p>Se desea ver en pantalla un listado de códigos de jugadores, sin repeticiones, que hayan alcanzado el nivel Platinado en al menos un juego.</p> <p>¿Cuál de los siguientes programas cumple con lo pedido ?</p>		
1.	Ninguna de las opciones hace lo solicitado.	X 1.
2.	<pre>def platino(line): return "Platinado" in line file = open(file_name, "r") lineas = file.readlines() file.close() platinados = {} for linea in lineas: if (platino(linea)): platinados[linea.split(";")[0]] = linea.split(";")[1] for jugador in platinados: print(jugador)</pre>	2.
3.	<pre>def platino(line): return "Platinado" in line file = open(file_name, "r") lineas = file.readlines() file.close() platinados = [] for linea in lineas: if (platino(linea)): platinados.append(linea.split(";")[1]) for jugador in platinados: print(jugador)</pre>	3.
4.	<pre>def platino(line): return "Platinado" in line file = open(file_name, "r") lineas = file.readlines() file.close() platinados = () for linea in lineas: if (platino(linea)): platinados.append(linea.split(";")[1]) for jugador in platinados: print(jugador)</pre>	4.

Ejercicio 0902					
Teniendo el siguiente DataFrame declarado como 'df' :					
	nombre	director	año	género	puntaje
0	Titanic	James Cameron	1997	romance	8.6
1	Kil Bill	Quentin Tarantino	2003	acción	NaN
2	Matrix	Hermanas Wachowski	1999	ciencia ficción	6.9
3	El padrino	Francis Ford Coppola	1972	drama	7.5
4	Avatar	James Cameron	2009	ciencia ficción	9.1
5	Casablanca	Michael Curtiz	1942	drama	6.0
6	El exorcista	William Friedkin	1973	terror	NaN
7	Soy leyenda	Francis Lawrence	2007	ciencia ficción	NaN
8	El club de la pelea	David Fincher	1999	drama	9.4
9	Mujercitas	Greta Gerwig	2019	drama	8.0

**¿Cuál de las siguientes opciones me devuelve la cantidad total de géneros distintos que se encuentran en la columna 'género' del DataFrame?
Se considera que el código tiene que funcionar de forma general, no sólo en los ejemplos mostrados en la imagen.**

1.	Ninguna de las opciones devuelve lo pedido	X	1.
2.	df.groupby('género').size()		2.
3.	df['género'].value_counts()		3.
4.	df[['género']].count()		4.

Ejercicio 1002			
Para el DataFrame del punto anterior, y considerando el siguiente código:			
<pre>p = df['puntaje'].between(6,8) a = df['año'] > 2002 nuevo = df[a p]</pre>			
¿Qué pasa si imprimimos/mostramos el contenido de 'nuevo'?			
1.	Muestra las películas que tienen puntaje mayor a 6 y menor a 8, y que además son posteriores al año 2002.		1.
2.	Muestra las películas que tienen puntaje mayor o igual a 6 y menor a 8, y que podrían o no ser posteriores al año 2002.		2.
3.	Muestra las películas que tienen puntaje mayor o igual a 6 y menor a 8 y/o que son posteriores al año 2002.		3.
4.	Muestra las películas que tienen puntaje mayor o igual a 6 y menor o igual a 8 y/o que son posteriores al año 2002.	X	4.

Ejercicio 1102			
Considerando el DataFrame del punto 9 y el siguiente código:			
df['puntaje'].isnull()			
¿Cuál de las siguientes afirmaciones es correcta?			
1.	El código no agrega una columna al dataFrame original. Sólo muestra una columna con los valores True (si la película tiene puntaje) o False (si no tiene).		1.
2.	El código no agrega una columna al dataFrame original. Sólo muestra una columna con los valores False (si la película tiene puntaje) o True (si no tiene).	X	2.
3.	El código agrega la columna 'p' al dataFrame original. Los valores de la columna van a ser True si la película tiene puntaje, False si no tiene.		3.
4.	El código agrega la columna 'p' al dataFrame original. Los valores de la columna van a ser False si la película tiene puntaje, True si no tiene.		4.

Ejercicio 1202			
¿Cuál de estas afirmaciones es correcta?			
1.	La función <code>plot</code> recibe 2 arreglos principales que pueden o no contener números.		1.
2.	Los labels son de las partes menos importantes del gráfico.		2.
3.	La función <code>bar</code> debe recibir dos arreglos del mismo tamaño para poder mostrar un gráfico.	X	3.
4.	El gráfico <code>pie</code> se utiliza para mostrar el tamaño total de un conjunto de datos.		4.

Ejercicio 1302			
<p>Se tiene el siguiente programa</p> <pre>def todosPos(x): return abs(x) def impares(x): return x%2!=0 a=[3,-1,22,4,-100,99] lista1=list(filter(impares,a)) lista2=list(map(todosPos,lista1))</pre> <p>¿Qué contenido tendrá lista2 al ejecutarlo?</p>			
1.	[22, 4]		1.
2.	[-100]		2.
3.	[22, 4, 100]		3.
4.	[3, 1, 99]	X	4.