

11/05/23

Ejercicio 0107		
Seleccione la opción correcta		
1.	Usar un lenguaje de programación no nos permite usar lenguaje coloquial para comunicarnos con la computadora.	X
2.	Usar un lenguaje de programación es imprescindible para crear algoritmos de calidad.	
3.	Usar un lenguaje de programación nos asegura que nuestros programas funcionarán correctamente.	
4.	Usar un lenguaje de programación requiere que cumplamos con ciertas reglas y si cumplimos con ellas, podemos usar un lenguaje coloquial.	
5.	Usar un lenguaje de programación nos ayuda a que nuestros programas tengan más líneas.	

Ejercicio 0202		
¿Para qué sirven las funciones?		
1.	Si no creamos funciones los programas no andan, por lo que sirven para que nuestro programa funcione	
2.	Para usar estructuras iterativas	
3.	Para dividir los problemas en problemas más pequeños	X
4.	Las funciones no tienen un objetivo, se crean cuando uno quiere	
5.	Ninguna es correcta	

Ejercicio 0306		
¿Cuáles de los siguientes son programas válidos si se quiere mostrar por pantalla el valor de v y w?		
1.	<pre>print("V vale", v) print("W vale", v)</pre>	
2.	<pre>print("V vale", 4) print("W vale", 5)</pre>	
3.	<pre>def algo(v): print("V vale " + str(v)) def algo(w): print("W vale " + str(w)) v = 6 w = 8 algo(v) algo(w)</pre>	
4.	<pre>def algo_1(v): print("V vale " + str(v)) def algo_2(w): print("W vale " + str(w)) v = 6 w = 8 algo_1(v) algo_2(w)</pre>	
5.	<pre>def algo_1(x): print("V vale " + str(x)) def algo_2(y): print("W vale " + str(y)) v = 6 w = 8 algo_1(v) algo_2(w)</pre>	
6.	Ninguno es válido	
7.	1 y 2 son correctas	
8.	4 y 5 son correctas	X

Ejercicio 0403		
¿Entre cuántos "camino" nos permite "elegir" la estructura if?		
1.	0	
2.	1	
3.	2	X

4.	Los que queremos	X
5.	Ninguna opción es correcta	

Si bien la respuesta correcta es la 4 ya que la estructura if completa permite elegir entre muchos caminos, damos cómo correcta la 3 también ya que la pregunta presta confusión por uno de los gráficos de nuestros apuntes.

Ejercicio 0507		
¿Cuál de los siguientes bloques de código tiene una notación correcta?		
1.	<pre>if txt[0]=='a' and txt[0]=='a': print('El texto empieza con a') elif txt[0]=='a' and txt[0]=='a': print('El texto empieza con a') else: print('El texto no empieza con a')</pre>	X
2.	<pre>if txt[0]=='a' and txt[0]=='a': print('El texto empieza con a') elseif txt[0]=='a' and txt[0]=='a': print('El texto empieza con a') else: print('El texto no empieza con a')</pre>	
3.	<pre>if txt[0]=='a' and txt[0]=='a': print('El texto empieza con a') elif txt[0]=='a' not and txt[0]=='a': print('El texto empieza con a') else: print('El texto no empieza con a')</pre>	
4.	<pre>if txt[0]=='a' and txt[0]!='a': print('El texto empieza con a') else: print('El texto no empieza con a') else: print('El texto no empieza con a')</pre>	

Ejercicio 0601		
¿Para qué sirven las sentencias iterativas?		
1.	Para determinar el camino a seguir, es decir, determinar el flujo de nuestro programa.	
2.	Para ejecutar un mismo bloque de código repetidas veces.	X
3.	Para decidir cuando algo es verdadero o falso.	
4.	Ninguna opción es correcta	

Ejercicio 0704		
¿Cuántas veces imprime 'hola' el siguiente bloque de código?		
<pre>i=0 while i<=7: print('hola') i = i + 1</pre>		
1.	Imprime 7 veces	
2.	Imprime 8 veces	X
3.	Nunca va a parar de imprimir	
4.	El bloque de código está mal sintácticamente, va a dar un error antes de ejecutarlo	

Ejercicio 0802		
Si tengo la siguiente lista:		
<pre>lista = [2,4,6]</pre> <p>Y quiero agregar el elemento "8" al final.</p> <p>¿Cuál de los siguientes bloques de código inserta correctamente el elemento?</p>		
1.	lista[3] = 8	
2.	lista.append(8)	X
3.	lista.add(8)	
4.	lista = lista + 8	

Ejercicio 0903		
----------------	--	--

Se quiere hacer una función que devuelva true si la lista recibida contiene todos elementos pares. ¿Cuál de las siguientes funciones hace lo pedido?		
1.	<pre>def todos_pares(lista): tiene_todos_pares = False for elemento in lista: if elemento % 2 == 0: tiene_todos_pares = True return tiene_todos_pares</pre>	
2.	<pre>def todos_pares(lista): tiene_todos_pares = False for elemento in lista: if elemento % 2 != 0: tiene_todos_pares = True return tiene_todos_pares</pre>	
3.	<pre>def todos_pares(lista): tiene_todos_pares = True for elemento in lista: if elemento % 2 != 0: tiene_todos_pares = False return tiene_todos_pares</pre>	X
4.	<pre>def todos_pares(lista): tiene_todos_pares = True for elemento in lista: if elemento % 2 == 0: tiene_todos_pares = False return tiene_todos_pares</pre>	

Ejercicio 1002		
¿Cuál de los siguientes problemas podría ser resuelto usando la función map?		
1.	Quiero eliminar todos los elementos mayores a cien de una lista de números	
2.	Quiero multiplicar por dos a todos los elementos de una lista de números	X
3.	Quiero agregar un elemento nuevo a una lista	
4.	Quiero conseguir la suma de todos los elementos de una lista de números	

Ejercicio 1102		
Dado el siguiente fragmento de código		
<pre>def funcion_misteriosa(palabra): if(len(palabra)>=5): return palabra.upper() return palabra lista = ['Hola', 'cómo', 'están', 'todos?'] print(list(map(funcion_misteriosa ,lista)))</pre>		
¿Qué se imprimirá por pantalla cuando se ejecute?		
1.	['Hola', 'cómo', 'están', 'todos?']	
2.	['Hola', 'cómo', 'están', 'TODOS?']	
3.	['Hola', 'cómo', 'ESTÁN', 'TODOS?']	X
4.	['Hola', 'CÓMO', 'ESTÁN', 'TODOS?']	

Ejercicio 1207		
¿Cuál de los siguientes problemas podría ser resuelto usando la función filter?		
1.	Quiero agregar un elemento más al final de una lista de apellidos	
2.	Quiero quedarme sólo con los elementos que no contengan una 'Ñ' de una lista de nombres	X
3.	Quiero sumarle uno a todos los elementos de una lista de edades	
4.	Quiero obtener el máximo elemento de una lista de edades	

Ejercicio 1307		
Dado el siguiente fragmento de código		

```
def transformar(persona):
    return persona[0][0] + str(persona[1])

lista=[('Logan', 75), ('Kendall', 40), ('Roman', 35), ('Shiv', 30), ('Tom', 34), ('Connor',
54)]

lista_filtrada=list(map(transformar,lista))
print(lista_filtrada)
```

¿Qué se imprimirá por pantalla cuando se ejecute?

1.	['o75', 'e40', 'o35', 'h30', 'o34', 'o54']	
2.	[('L', '75'), ('K', '40'), ('R', '35'), ('S', '30'), ('T', '34'), ('C', '54')]	
3.	['L75', 'K40', 'R35', 'S30', 'T34', 'C54']	X
4.	['L-75', 'K-40', 'R-35', 'S-30', 'T-34', 'C-54']	
5.	['n75', 'l40', 'n35', 'v30', 'm34', 'r54']	
6.	[]	