

# Test de Katas

Puntos totales 14/14

Todas las preguntas tienen una y una sola respuesta correcta.  
No existe en ningún caso la opción de "ninguna de las anteriores".

**0 de 0 puntos**

correo electrónico \*

dianamtlopez@hotmail.com

Nombre \*

DIANA MARIA

Apellidos \*

TORO LOPEZ

Pregunta nº 1

**1 de 1 puntos**



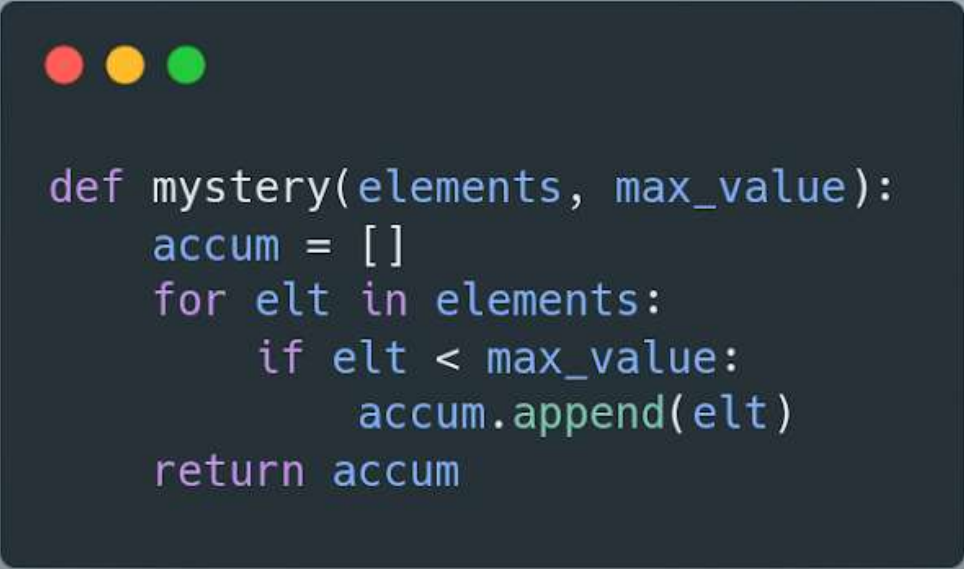
```
def h(x):  
    if x % 2 == 0:  
        return x * 2  
    else:  
        return x
```

¿Qué hace esta  
función?

\*

1/1

- ☐ Recibe una lista y la devuelve sin modificar.
- ☐ Recibe una cadena y si su longitud es mayor que 2, la duplica.
- ☒ Recibe un número. Si es par, lo devuelve multiplicado por dos y si es impar lo devuelve sin alterar.



```
def mystery(elements, max_value):  
    accum = []  
    for elt in elements:  
        if elt < max_value:  
            accum.append(elt)  
    return accum
```

¿Qué tipo devuelve la siguiente función?

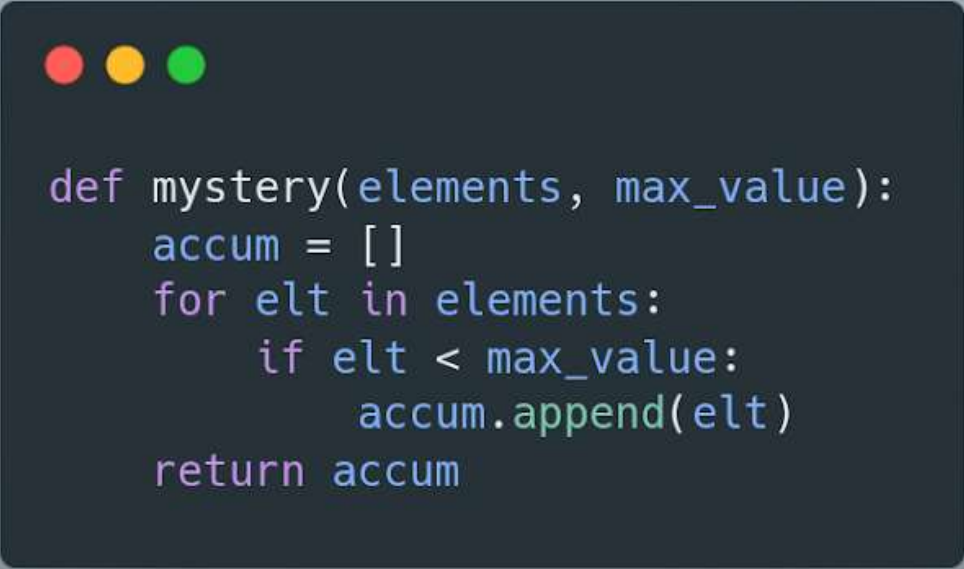
\*

1/1

- ☒ Una lista
- ☐ Un número
- ☐ Otra función

Pregunta nº 3

1 de 1 puntos




```
def mystery(elements, max_value):  
    accum = []  
    for elt in elements:  
        if elt < max_value:  
            accum.append(elt)  
    return accum
```

¿Qué hace la siguiente  
función?

\*

1/1

- ☐ Devuelve el valor máximo que hay en la lista `elements`.
- ☐ Provoca un error si la llamas de la siguiente manera: `mystery([], 5)`
- ☐ Devuelve una lista con los valores mayores que `max\_value`
- ☒ Devuelve una lista con los valores menores que `max\_value`.



```
def f(string):  
    vowels = set('aeiou')  
    normalized = string.lower()  
    return normalized[0] in vowels
```

¿Qué devuelve esta  
función?


\*

1/1

- ☐ Una cadena
- ☐ Un número
- ☒ Un booleano

Pregunta nº 5

1 de 1 puntos



```
def f(string):  
    vowels = set('aeiou')  
    normalized = string.lower()  
    return normalized[0] in vowels
```

¿Qué hace esta  
función?

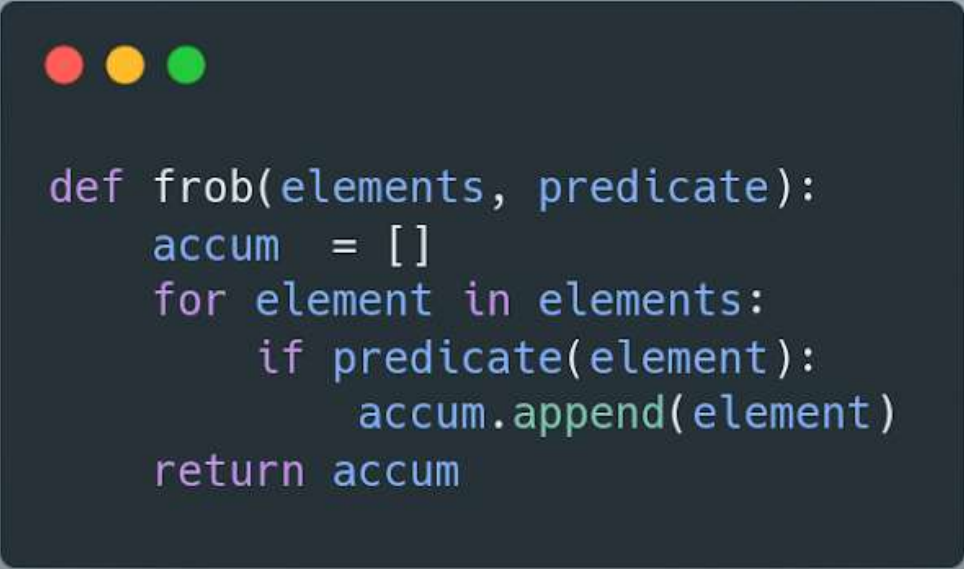
\*

1/1

- ☐ Es un predicado que determina si una cadena está hecha sólo de vocales.
- ☒ Es un predicado que indica si una palabra empieza por una vocal.
- ☐ Convierte una cadena de mayúsculas a minúsculas

Pregunta nº 6

1 de 1 puntos



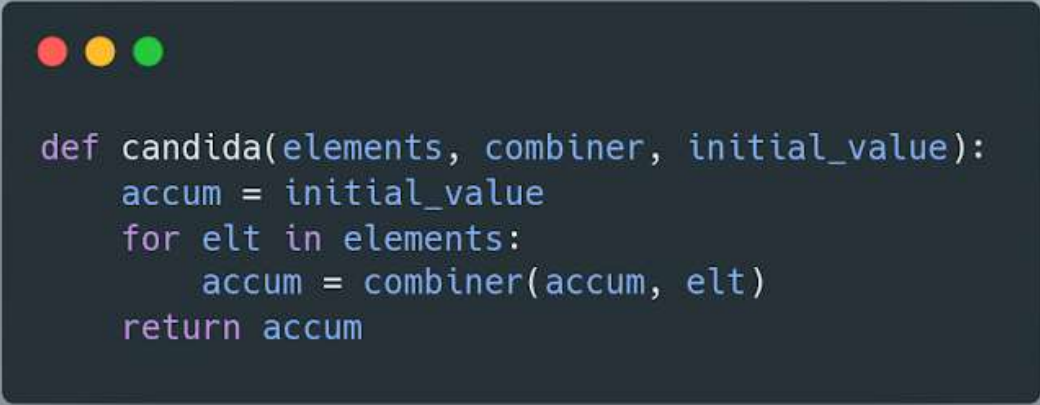
```
def frob(elements, predicate):  
    accum = []  
    for element in elements:  
        if predicate(element):  
            accum.append(element)  
    return accum
```

¿Qué hace esta  
función?

\*

1/1

- ☐ Recibe dos listas, y las une.
- ☐ Recibe una lista y un predicado. Devuelve una nueva lista con aquellos elementos para los cuales el predicado devuelve False.
- ☒ Recibe una lista y un predicado. Devuelve una nueva lista con aquellos elementos para los cuales el predicado devuelve True.



```
def candida(elements, combiner, initial_value):  
    accum = initial_value  
    for elt in elements:  
        accum = combiner(accum, elt)  
    return accum
```

¿Qué hace la siguiente  
función?

\*

1/1

- ☐ Recibe una lista y devuelve otra lista con los elementos originales modificados por la función combiner.
- ☐ Recibe una lista y un predicado y devuelve una nueva lista con aquellos elementos que pasan el test del predicado.
- ☒ Reduce una lista de valores a uno solo, combinándolos mediante la función combiner y el valor inicial.

Pregunta nº 8

1 de 1 puntos



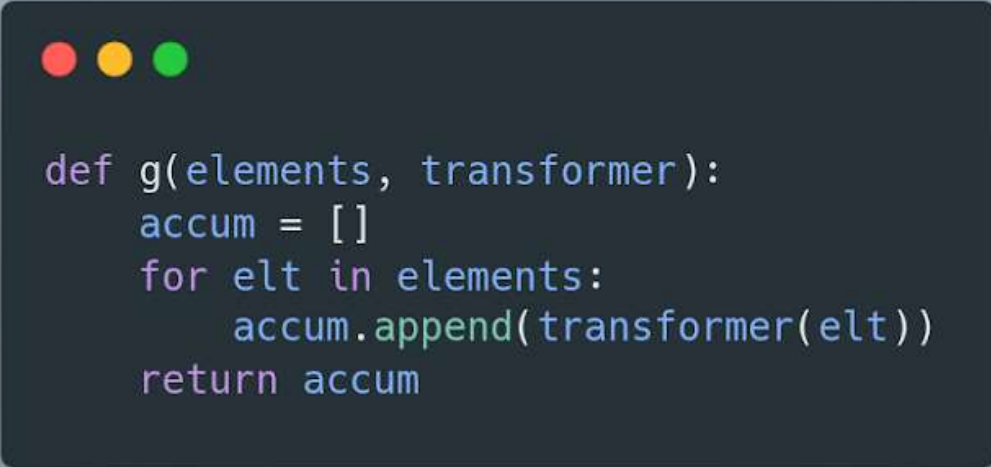
```
def candida(elements, combiner, initial_value):  
    accum = initial_value  
    for elt in elements:  
        accum = combiner(accum, elt)  
    return accum  
  
candida([1,2,34, 56], lambda a, b: a + b, 0)
```

¿Cual es el resultado de llamar a la función `candida` de la siguiente manera?

\*

1/1

- ☒ Devuelve la suma de todos los elementos de la lista.
- ☐ Devuelve el producto de todos los elementos de la lista.
- ☐ Devuelve una nueva lista con las sumas de los elementos originales



```
def g(elements, transformer):  
    accum = []  
    for elt in elements:  
        accum.append(transformer(elt))  
    return accum
```

¿Qué hace la función *g*? \*

1/1

- ☐ Recibe una lista y una función. Usa esa función para reducir la lista a un sólo elemento.
- ☒ Recibe una lista y una función. Devuelve una nueva lista con los elementos originales modificados mediante la función transformer.
- ☐ Recibe una lista y devuelve otra lista de longitud diferente.

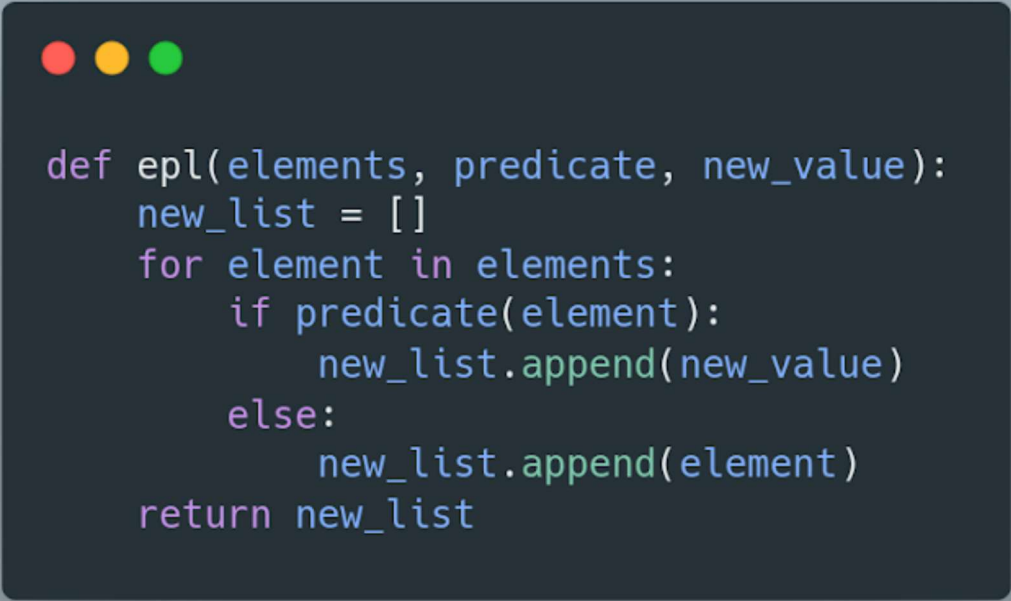
```
def g(elements, transformer):  
    accum = []  
    for elt in elements:  
        accum.append(transformer(elt))  
    return accum  
  
def lens(list_of_lists):  
    return g(list_of_lists, ????)
```

Tienes que crear la función `lens`, que recibe una lista de listas y devuelve una lista de las longitudes de cada una de esas listas. Para ello, tienes que usar la función `g` ya definida. ¿Qué habría que poner en vez de los puntos de interrogación?

\*

1/1

- ☒ `len`
- ☐ `lambda a,b : a + b`
- ☐ `map`

A dark-themed terminal window with three colored window control buttons (red, yellow, green) in the top-left corner. It contains a Python function definition for 'ep1'.

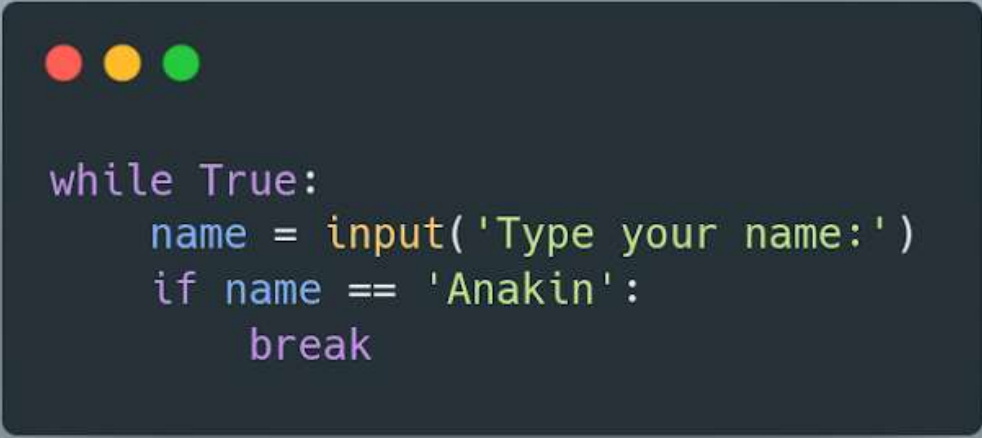
```
def ep1(elements, predicate, new_value):  
    new_list = []  
    for element in elements:  
        if predicate(element):  
            new_list.append(new_value)  
        else:  
            new_list.append(element)  
    return new_list
```

¿Qué hace esta  
función?

\*

1/1

- ☐ Itera por una lista y devuelve una nueva lista sólo con aquellos elementos que superan el test del predicado que recibe como parámetro.
- ☐ Hace lo mismo que la función reduce
- ☒ Recorre una lista y devuelve una nueva lista con los elementos que pasan el test del predicado sustituidos por new\_value.



```
while True:
    name = input('Type your name:')
    if name == 'Anakin':
        break
```

¿Qué hará este bucle  
while?

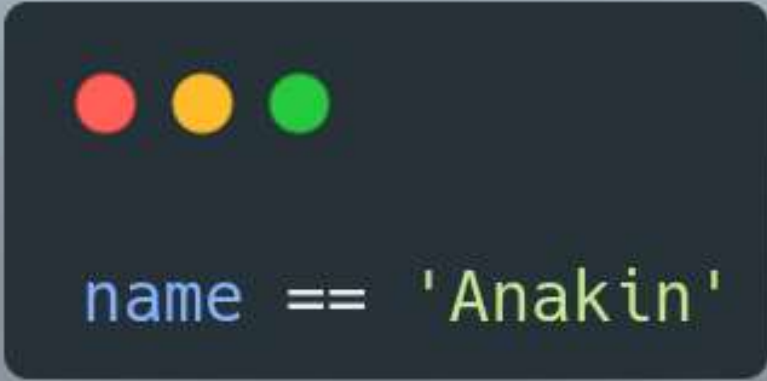
\*

1/1

- ☐ Es un bucle infinito, del cual no hay posibilidad de salir.
- ☐ True no es una expresión booleana, así que jamás se ejecutará.
- ☒ Seguirá pidiendo el nombre al usuario hasta que conteste Anakin.

Pregunta nº 13

1 de 1 puntos



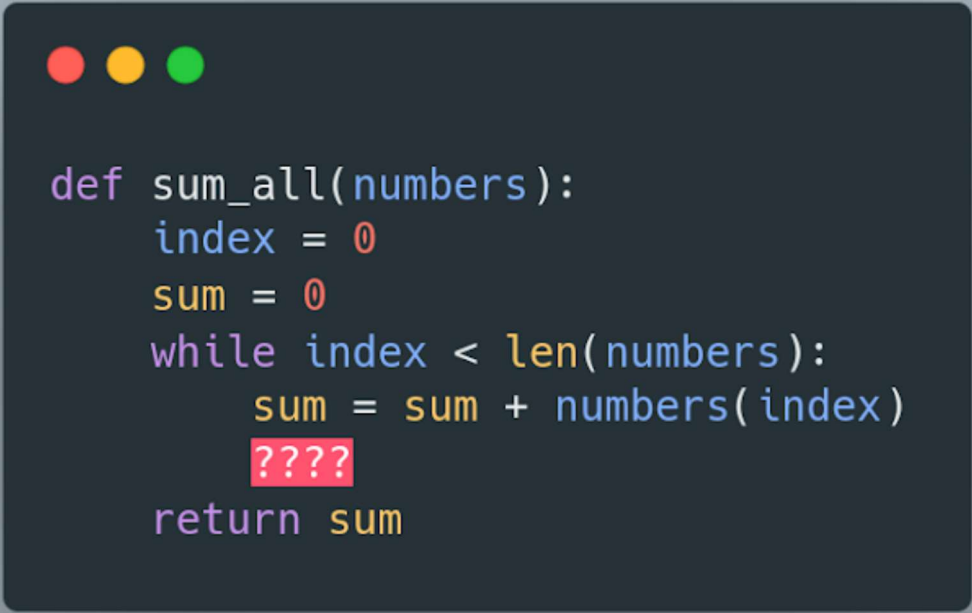
```
name == 'Anakin'
```

¿Qué hace esta línea de código?

\*

1/1

- ☐ Asigna la cadena 'Anakin' a la variable name
- ☒ Compara el valor de la variable name con la cadena 'Anakin' y devuelve un booleano.
- ☐ Compara el valor de la variable name con la cadena 'Anakin' y devuelve un número.



```
def sum_all(numbers):  
    index = 0  
    sum = 0  
    while index < len(numbers):  
        sum = sum + numbers(index)  
        ???  
    return sum
```

¿Qué le falta a esta función? Queremos crear una función que suma todos los elementos de una lista. ¿Qué habría que poner en vez de ??? para que funcione?

\*

1/1

- ☐ No falta nada.
- ☐ sum = sum + index
- ☒ index = index + 1