

✓ ¡Felicitaciones! ¡Aprobaste!

Calificación recibida 100 % Para Aprobar 80 % o más

[Ir al siguiente elemento](#)

 Estás viendo una versión traducida automáticamente de esta evaluación

Puedes volver a ver el contenido en su idioma original si lo prefieres. No perderás el progreso que hayas conseguido si cambias el idioma.

Desestimar ✕

1. Rellene los espacios en blanco para completar la función `is_palindromo`. Esta función comprueba si una cadena dada es un palíndromo. Un palíndromo es una cadena que contiene las mismas letras en el mismo orden, tanto si la palabra se lee de izquierda a derecha como de derecha a izquierda. Ejemplos de palíndromos son palabras como kayak y radar, y frases como "Ni par ni impar". La función debe ignorar los espacios en blanco y las mayúsculas al comprobar si la cadena dada es un palíndromo. Complete esta función para que devuelva `True` si la cadena pasada es un palíndromo. `False` en caso contrario.

1 / 1 punto

```

1 def is_palindrome(input_string):
2     # Two variables are initialized as string data types using empty
3     # quotes: "reverse_string" to hold the "input_string" in reverse
4     # order and "new_string" to hold the "input_string" minus the
5     # spaces between words, if any are found.
6     new_string = ""
7     reverse_string = ""
8
9     # Complete the for loop to iterate through each letter of the
10    # "input_string"
11    for letter in input_string:
12
13        # The if-statement checks if the "letter" is not a space.
14        if letter != " ":
15
16            # If True, add the "letter" to the end of "new_string" and
17            # to the front of "reverse_string". If False (if a space
18            # is detected), no action is needed. Exit the if-block.
19            new_string += letter.lower()
20            reverse_string = letter.lower() + reverse_string
21
22    # Complete the if-statement to compare the "new_string" to the
23    # "reverse_string". Remember that Python is case-sensitive when
24    # creating the string comparison code.
25    if new_string == reverse_string:
26
27        # If True, the "input_string" contains a palindrome.
28        return True
29
30    # Otherwise, return False.
31    return False
32
33
34 print(is_palindrome("Never Odd or Even")) # Should be True
35 print(is_palindrome("abc")) # Should be False
36 print(is_palindrome("kavak")) # Should be True

```

**Eigenspace**

Rectangles

True  
False  
True

✔ **Correcto**

¡Woohool! Te estás convirtiendo rápidamente en el experto en cadenas de Python!

2. Utilizando el método `format`, rellene los huecos de la función `convert_distance` para que devuelva la frase "X millas equivalen a Y km", teniendo Y sólo 1 decimal. Por ejemplo, `convert_distance(12)` debería devolver "12 millas equivalen a 19.2 km".

1 / 1 punto

```
1 def convert_distance(miles):
2     km = miles * 1.6
3     result = "{} miles equals {:.1f} km".format(miles, km)
4     return result
5
6
7 print(convert_distance(12)) # Should be: 12 miles equals 19.2 km
8 print(convert_distance(5.5)) # Should be: 5.5 miles equals 8.8 km
9 print(convert_distance(11)) # Should be: 11 miles equals 17.6 km
```

## Ejecutar

Restablecer

```
12 miles equals 19.2 km
5.5 miles equals 8.8 km
11 miles equals 17.6 km
```

✔ **Correcto**

¡Enhorabuena! Le está cogiendo el truco a formatear cadenas,  
¡hurra!

3. Si tenemos una variable de cadena denominada Tiempo = "Lluvia", ¿cuál de las siguientes opciones imprimirá la subcadena "Lluvia" o todos los caracteres anteriores a la "f"?

1 / 1 punto

☒ `print(El tiempo[:4])`

☐ `print(El tiempo[4:])`

☐ `print(Tiempo[1:4])`

☐ `print(Tiempo[: "f"])`

✔ **Correcto**

Correcto. Formateado de esta manera, se imprimirá la subcadena que precede al carácter "f", que está indexado por 4

4. Rellene los huecos de la función `nametag` para que utilice el método `format` y devuelva `first_name` y la primera inicial de `last_name` seguidos de un punto. Por ejemplo, `nametag("Jane", "Smith")` debería devolver "Jane S"

1 / 1 punto

```
1 def nametag(first_name, last_name):
2     return("{} {}".format(first_name, last_name[0]))
3
4
5 print(nametag("Jane", "Smith"))
6 # Should display "Jane S."
7 print(nametag("Francesco", "Rinaldi"))
8 # Should display "Francesco R."
9 print(nametag("Jean-Luc", "Grand-Pierre"))
10 # Should display "Jean-Luc G."
```

Ejecutar

Restablecer

Jane S.  
Francesco R.  
Jean-Luc G.

✔ **Correcto**

¡Gran trabajo! Ha recordado la expresión de formato para limitar cuántos caracteres de una cadena se muestran.

5. La función `replace_ending` sustituye una subcadena especificada al final de una sentencia dada por una nueva subcadena. Si la subcadena especificada no aparece al final de la sentencia dada, no se realiza ninguna acción y se devuelve la sentencia original. Si hay más de una aparición de la subcadena especificada en la sentencia, sólo se sustituye la subcadena del final de la sentencia. Por ejemplo, `replace_ending("abcabc", "abc", "xyz")` debería devolver `abcxyz`, no `xyzxyz` o `xyzabc`. La comparación de cadenas distingue entre mayúsculas y minúsculas, por lo que `replace_ending("abcabc", "ABC", "xyz")` debería devolver `abcabc` (sin cambios).

1 / 1 punto

```

1 def replace_ending(sentence, old, new):
2     # Check if the old substring is at the end of the sentence
3     if sentence.endswith(old):
4         # Using i as the slicing index, combine the part
5         # of the sentence up to the matched string at the
6         # end with the new string
7         i = len(sentence) - len(old)
8         new_sentence = sentence[:i] + new
9         return new_sentence
10
11
12     # Return the original sentence if there is no match
13     return sentence
14
15 print(replace_ending("It's raining cats and cats", "cats", "dogs"))
16 # Should display "It's raining cats and dogs"
17 print(replace_ending("She sells seashells by the seashore", "seashells", "donuts"))
18 # Should display "She sells seashells by the seashore"
19 print(replace_ending("The weather is nice in May", "may", "april"))
20 # Should display "The weather is nice in May"
21 print(replace_ending("The weather is nice in May", "May", "April"))
22 # Should display "The weather is nice in April"

```

### Ejecutar

Restablecer

It's raining cats and dogs  
She sells seashells by the seashore  
The weather is nice in May  
The weather is nice in April

✔ **Correcto**

¡Sobresaliente! Mira todas las cosas que puedes hacer con estos comandos de cadena!