Tu calificación: 83,33 %

Tu calificación más reciente: **83,33 %** • Tu calificación más alta: **83,33 %** • Para aprobar necesitas al menos un 80 %. Guardamos tu puntaje más alto.

Próximo artículo

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.
Mostrar la versión en Inglés
Desestimar

1. La función `check_web_address()` comprueba si el texto pasado cumple los requisitos para ser una dirección web de nivel superior, es decir, si contiene caracteres alfanuméricos (que incluyen letras, números y guiones bajos), así como puntos, guiones y un signo más, seguidos de un punto y un dominio de nivel superior de sólo caracteres como ".com", ".info", ".edu", etc. Rellene la expresión regular para ello, utilizando caracteres de escape, comodines, calificadores de repetición, caracteres de principio y fin de línea y clases de caracteres.

1 / 1 punto

```
import re
def check_web_address(text):
  pattern = r'^(www\.)?[a-zA-Z0-9_-]+\.[a-zA-Z]{2,}$'
  result = re.search(pattern, text)
  return result != None

print(check_web_address("gmail.com")) # True
print(check_web_address("www@google")) # False
print(check_web_address("www.Coursera.org")) # True
print(check_web_address("web-address.com/homepage")) # False
print(check_web_address("My_Favorite-Blog.US")) # True

```

Ejecutar
Restablecer

Correcto
¡Muy bien! Ninguna dirección web falsa se te escapará

2. La función `check_time()` comprueba el formato de hora de un reloj de 12 horas, como sigue: la hora está entre 1 y 12, sin cero inicial, seguida de dos puntos, luego los minutos entre 00 y 59, luego un espacio opcional, y luego AM o PM, en mayúsculas o minúsculas. Rellene la expresión regular para hacerlo. ¿Cuántos de los conceptos que acabas de aprender puedes utilizar aquí?

1 / 1 punto

```
import re
def check_time(text):
  pattern = r'^(1[0-2]|[1-9]):[0-5][0-9]\s?(AM|PM|am|pm)$'
  result = re.search(pattern, text)
  return result != None

print(check_time("12:45pm")) # True
print(check_time("9:59 AM")) # True
print(check_time("6:60am")) # False
print(check_time("five o'clock")) # False
print(check_time("6:02 am")) # True
print(check_time("6:02km")) # False
```

Ejecutar
Restablecer

Correcto
¡Lo has clavado! ¡Es "hora" de celebrar!

3. La función `contains_acronym()` comprueba la presencia en el texto de 2 o más caracteres o dígitos rodeados de paréntesis, con al menos el primer carácter en mayúscula (si es una letra), y devuelve `True` si se cumple la condición, o `False` en caso contrario. Por ejemplo, "La mensajería instantánea (MI) es un conjunto de tecnologías de comunicación utilizadas para la comunicación basada en texto" debería devolver `True` ya que (MI) satisface las condiciones de coincidencia" Introduzca la expresión regular en esta función:

1 / 1 punto

```
import re
def contains_acronym(text):
  pattern = r'\([A-Za-z0-9]+\)'
  result = re.search(pattern, text)
  return result != None

print(contains_acronym("Instant messaging (IM) is a set of communication technologies used for text-based communication")) # True
print(contains_acronym("American Standard Code for Information Interchange (ASCII) is a character encoding standard for electronic
print(contains_acronym("Please do NOT enter without permission!")) # False
print(contains_acronym("PostScript is a fourth-generation programming language (4GL)")) # True
print(contains_acronym("Have fun using a self-contained underwater breathing apparatus (Scuba)!")) # True
```

Ejecutar
Restablecer

Correcto
¡Buen trabajo! Eso sí que es Responsabilidad Personal
La Excelencia (PRIDE)

4. ¿Qué indica `r` antes de la cadena de patrones en `re.search(r"Py.*n", sample.txt)`?

1 / 1 punto

- (•) Cadenas en bruto
- ( ) Regex
- ( ) Repita
- ( ) Resultado

Correcto
¡RIGHT! las cadenas "crudas" sólo significan que el intérprete de Python no intentará interpretar ningún carácter especial y, en su lugar, pasará la cadena a la función tal cual.

5. ¿Qué hace el carácter más [+] en regex?

1 / 1 punto

- ( ) Coincide con caracteres de signo más
- (•) coincide con una o más apariciones del carácter que le precede
- ( ) Coincide con el final de una cadena
- ( ) Coincide con el carácter anterior a [+] sólo si hay más de uno

Correcto
Impresionante El carácter más [+], coincide con una o más ocurrencias del carácter que le precede.

6. Un becario ha implementado un comprobador de códigos postales, pero sólo funciona con códigos postales de cinco dígitos. Su tarea consiste en actualizar el verificador para que incluya los nueve dígitos del código postal; los cinco primeros dígitos y los cuatro opcionales después del guión. El código postal debe ir precedido de al menos un espacio y no puede estar al principio del texto. Actualice la expresión regular.

0 / 1 punto

```
import re
def check_zip_code (text):
  result = re.search(r"\b\d{5}(-\d{4})?\b", text)
  return result != None

print(check_zip_code("The zip codes for New York are 10001 thru 11104.")) # True
print(check_zip_code("90210 is a TV show")) # True
print(check_zip_code("Their address is: 123 Main Street, Anytown, AZ 85258-0001.")) # True
print(check_zip_code("The Parliament of Canada is at 111 Wellington St, Ottawa, ON K1A0A9.")) # False
```

Ejecutar
Restablecer

Incorrecto
No exactamente. ¿Está comprobando exactamente 5 dígitos, y que caracteres pueden preceder y seguir al código postal?