

Inicio (<https://recursospython.com/>)

Códigos de fuente (<https://www.recursospython.com/category/codigos-de-fuente/>)

Guías y manuales (<https://www.recursospython.com/category/guias-y-manuales/>)


Foro (<https://foro.recursospython.com/>) Micro (<https://micro.recursospython.com/>)

Tutorial (<https://tutorial.recursospython.com/>) Newsletter (<https://recursospython.com/newsletter/>)

Consultoría (<https://recursospython.com/consultoria/>)

Contacto (<https://recursospython.com/contacto/>) Donar ❤️ (<https://recursospython.com/donar/>)

Verificar ortografía de una página web

marzo 8, 2016 (<https://recursospython.com/codigos-de-fuente/verificar-ortografia-de-una-pagina-web/>) by Recursos Python (<https://recursospython.com/author/admin/>)  (<https://recursospython.com/codigos-de-fuente/verificar-ortografia-de-una-pagina-web/#comments>) Dejar un comentario (<https://recursospython.com/codigos-de-fuente/verificar-ortografia-de-una-pagina-web/#respond>)

Versión: 3.x.

Descarga: [urlsc.zip \(<https://www.recursospython.com/wp-content/uploads/2016/03/urlsc.zip>\)](https://www.recursospython.com/wp-content/uploads/2016/03/urlsc.zip).

El siguiente código permite chequear la ortografía de una página web dada una URL. El programa lee el contenido de la dirección solicitada utilizando la función estándar `urllib.request.urlopen` (<https://docs.python.org/3/library/urllib.request.html#urllib.request.urlopen>) extrae el texto con [Beautiful Soup 4](http://www.crummy.com/software/BeautifulSoup/) (<http://www.crummy.com/software/BeautifulSoup/>) y verifica cada una de las palabras vía [Hunspell](https://www.recursospython.com/guias-y-manuales/hunspell-correector-ortografico/) (<https://www.recursospython.com/guias-y-manuales/hunspell-correector-ortografico/>).

```
1. #!/usr/bin/env python
2. # -*- coding: utf-8 -*-
3. #
4. # Copyright 2016 Recursos Python -
   # www.recursospython.com
5. #
6. #
```



Últimas entradas

[Reproducir inyección SQL en sqlite3 y PyMySQL \(<https://recursospython.com/guias-y-manuales/reproducir-inyeccion-sql-en-sqlite3-y-pymysql/>\)](#)

[Bloc de notas simple con Tk \(tkinter\) \(<https://recursospython.com/codigos-de-fuente/bloc-de-notas-simple-con-tkinter/>\)](#)

[Examinar archivo o carpeta en Tk \(tkinter\) \(<https://recursospython.com/guias-y-manuales/examinar-archivo-o-carpeta-en-tk-tkinter/>\)](#)

```

7.
8. from string import ascii_lowercase
9. from urllib.request import urlopen
10.
11. from bs4 import BeautifulSoup
12. from hunspell import HunSpell
13.
14.
15. def chars_filter(s, valid_chars):
16.     return "".join(c for c in s if c in valid_chars)
17.
18.
19. def main():
20.     url = "https://www.rekursospython.com/guias-y-
manuales/por-que-existe-python-3/"
21.
22.     # Obtener contenido de la página.
23.     with urlopen(url) as r:
24.         content = r.read()
25.
26.     # Analizar el código HTML.
27.     soup = BeautifulSoup(content, "html.parser")
28.     # Etiqueta HTML de donde extraer el texto.
29.     root = soup.article
30.
31.     # Remover código innecesario.
32.     for tag in root.find_all(["script", "style", "code",
"pre"]):
33.         tag.decompose()
34.
35.     # Extraer texto, remover saltos de línea y convertir
36.     # a minúsculas para agilizar la búsqueda.
37.     text = root.get_text().replace("\n", " ").lower()
38.     # Remover caracteres innecesario (puntuación y demás).
39.     text = chars_filter(text, ascii_lowercase + "áéíóüñ ")
40.
41.     # Crear el diccionario y agregar las palabras
necesarias.
42.     dic = HunSpell("es_ANY.dic", "es_ANY.aff")
43.     dic.add("python")
44.
45.     unknown_words = {}
46.
47.     # Buscar palabras que no se encuentren en el
diccionario.
48.     for word in text.split(" "):
49.         if word:
50.             # Ignorar letras sueltas.
51.             if len(word) > 1 and not dic.spell(word):
52.                 if word in unknown_words:
53.                     unknown_words[word] += 1
54.                 else:
55.                     unknown_words[word] = 1
56.
57.     print(len(unknown_words), "palabras desconocidas.")
58.
59.     # Ordenar alfabéticamente e imprimir sugerencias.
60.     for word in sorted(unknown_words):
61.         print("{} ({}).".format(word, unknown_words[word]))
62.         suggest = dic.suggest(word)
63.         if suggest:
64.             print("{}Quiso decir {}?".format(
65.                 word, ".join(s.decode("utf-8") for s in
suggest)))
66.
67.
68. if __name__ == "__main__":

```

[Múltiples configuraciones \(desarrollo/producción\) en Django \(https://recursospython.com/guias-y-manuales/multiples-configuraciones-desarrollo-produccion-en-django/\)](#)

[Buscar el archivo de mayor tamaño en una ruta \(https://recursospython.com/comandos-de-fuente/buscar-el-archivo-de-mayor-tamano-en-una-ruta/\)](#)

Comentarios recientes

Recursos Python en [Generar código QR \(https://recursospython.com/guias-y-manuales/generar-codigo-qr/#comment-2586\)](#)

Joaquín en [Generar código QR \(https://recursospython.com/guias-y-manuales/generar-codigo-qr/#comment-2584\)](#)

Recursos Python en [pickle – Serialización de objetos \(https://recursospython.com/guias-y-manuales/pickle-serializacion-de-objetos/#comment-2435\)](#)

Recursos Python en [Lista desplegable \(Combobox\)](#)

Con la URL por defecto del artículo [Por qué existe Python 3](https://www.rekursospython.com/guias-y-manuales/por-que-existe-python-3/) (<https://www.rekursospython.com/guias-y-manuales/por-que-existe-python-3/>) el script imprime las siguientes sugerencias.

18 palabras desconocidas.

agradecidamente (1).

¿Quiso decir agradecida mente, agradecida-mente, desgraciadamente, agradecimiento, eternecidamente, encarecidamente?

ascii (1).

¿Quiso decir ascitis?

brett (1).

¿Quiso decir brete, Bretó?

bug (1).

¿Quiso decir bu, bus, bum, Buga?

by (1).

¿Quiso decir bu, y, bey, ay, be, bs, bv?

cannon (1).

¿Quiso decir canon, can non, can-non, cantonan, canoa, cano?

exists (1).

¿Quiso decir exista, existas, existes, existís, existe, existo, existí, existir?

pep (3).

¿Quiso decir peo, pe, pp, pepa, pepe, pea, pee, rep, pop, pes, pez, peí, peé?

poisición (1).

¿Quiso decir posición, inquisición?

puget (1).

¿Quiso decir puñete?

puppy (1).

¿Quiso decir Pupuya?

renumeramos (1).

¿Quiso decir remuneramos, enumeramos, re numeramos, re-numeramos, numeramos, enumeraros, enumeraos, enumerativos?

semánticamente (1).

¿Quiso decir semántica mente, semántica-mente, sistemáticamente, matemáticamente, esquemáticamente, cuánticamente?

solucionables (1).

¿Quiso decir solucionales, solucionarles, solucionadles, soluciona bles, soluciona-bles, solucionares, solucionas, soluciones, revolucionales?

sound (1).

¿Quiso decir undoso?

string (1).

¿Quiso decir astringir?

unicode (8).

¿Quiso decir unicornio?

why (1).

¿Quiso decir whisky?

[en Tcl/Tk \(tkinter\)](#)

(<https://recursospython.com/guias-y-manuales/lista-desplegable-combobox-en-tkinter/#comment-2434>)

Herná en [Lista](#)

[desplegable \(Combobox\)](#)

[en Tcl/Tk \(tkinter\)](#)

(<https://recursospython.com/guias-y-manuales/lista-desplegable-combobox-en-tkinter/#comment-2424>)

La función `get_text` de BeautifulSoup no es del todo precisa, por lo que puede retornar textos pertenecientes a comentarios HTML, código JavaScript o CSS, entre otros datos no deseados. Por esta razón es propicio indicar, siempre que sea posible, el bloque de código HTML en donde se encuentra el texto.

```
1. # Etiqueta HTML de donde extraer el texto.
2. root = soup.article
```

Otros ejemplos:

```
1. # Extraer el texto de todo el documento HTML.
2. root = soup
```

```
1. # Más específico.
2. root = soup.body.div
```

Además, probablemente querrás excluir algunas etiquetas de la búsqueda de texto.

```
1. # Remover código innecesario.
2. for tag in root.find_all(["script", "style", "code",
3. "pre"]):
    tag.decompose()
```

En general, el código JavaScript y CSS (`<script>` y `<style>`). En el caso particular de Recursos Python, añadido `<code>` y `<pre>` para excluir el código Python que incluyen la mayoría de los artículos.

Artículos relacionados

- [Tareas en segundo plano con Tcl/Tk \(tkinter\)](https://recursospython.com/guias-y-manuales/tareas-en-segundo-plano-con-tcl-tk-tkinter/)
(<https://recursospython.com/guias-y-manuales/tareas-en-segundo-plano-con-tcl-tk-tkinter/>)
- [Tareas en segundo plano con PyQt/PySide](https://recursospython.com/guias-y-manuales/tareas-en-segundo-plano-con-pyqt/)
(<https://recursospython.com/guias-y-manuales/tareas-en-segundo-plano-con-pyqt/>)
- [Hunspell – Corrector ortográfico](https://recursospython.com/guias-y-manuales/hunspell-corrector-ortografico/)
(<https://recursospython.com/guias-y-manuales/hunspell-corrector-ortografico/>)
- [Descargar archivos vía HTTP con urllib y urllib2](https://recursospython.com/codigos-de-fuente/descargar-archivos-urllib/)
(<https://recursospython.com/codigos-de-fuente/descargar-archivos-urllib/>)

Donar 

¿Te gusta nuestro contenido? ¡Ayúdanos a [seguir creciendo con una donación \(/donar\)](#)!



Entrada publicada en
Códigos de fuente (<https://recursospython.com/category/codigos-de-fuente/>) con las
etiquetas [beautifulsoup](https://recursospython.com/tag/beautifulsoup/) (<https://recursospython.com/tag/beautifulsoup/>)
[hunspell](https://recursospython.com/tag/hunspell/) (<https://recursospython.com/tag/hunspell/>)
[urllib](https://recursospython.com/tag/urllib/) (<https://recursospython.com/tag/urllib/>)

◀ [Por qué existe Python 3 \(https://recursospython.com/guias-y-manuales/por-que-existe-python-3/\)](https://recursospython.com/guias-y-manuales/por-que-existe-python-3/)

[El módulo estándar textwrap \(https://recursospython.com/guias-y-manuales/modulo-estandar-textwrap/\)](https://recursospython.com/guias-y-manuales/modulo-estandar-textwrap/) ▶

Deja una respuesta

Comentario *

Nombre *

Email *

Publicar el comentario

© 2013 - 2023

¡Suscríbete a nuestra newsletter! (<https://www.recursospython.com/newsletter/>)

[Políticas de Uso y Privacidad \(https://www.recursospython.com/politicas-de-uso-y-privacidad/\)](https://www.recursospython.com/politicas-de-uso-y-privacidad/)

En inglés: [Python Assets \(https://pythonassets.com/\)](https://pythonassets.com/)



(<https://creativecommons.org/licenses/by-nc/3.0/deed.es>)

