

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/>)

Solucionador de Sudoku

diciembre 26, 2016 (<https://recursospython.com/codigos-de-fuente/solucionador-de-sudoku/>) by Recursos Python (<https://recursospython.com/author/admin/>)  (<https://recursospython.com/codigos-de-fuente/solucionador-de-sudoku/#comments>) Dejar un comentario (<https://recursospython.com/codigos-de-fuente/solucionador-de-sudoku/#respond>)

Descarga: [sudokusolver.zip](https://www.recursospython.com/wp-content/uploads/2016/12/sudokusolver.zip) (<https://www.recursospython.com/wp-content/uploads/2016/12/sudokusolver.zip>).

El siguiente código de alrededor de 150 líneas es un simple solucionador del juego Sudoku

(<https://es.wikipedia.org/wiki/Sudoku>) escrito en Python 3.

Basándose en las reglas del juego, el pequeño programa determina un conjunto de números posibles para cada uno de los casilleros vacíos, y luego aplica fuerza bruta (seleccionar números aleatoriamente hasta obtener el resultado correcto). La fuerza bruta es sin duda uno de los métodos más simples para resolver este tipo de juegos matemáticos, pero no por eso poco eficaz. No será un complejo modelo de *Machine Learning* (https://es.wikipedia.org/wiki/Aprendizaje_autom%C3%A1tico) pero ha logrado pasar todas las pruebas: desde las grillas más simples a las más avanzadas. El tiempo de resolución del script es generalmente de unos pocos segundos.



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/>)

```
1.  #!/usr/bin/env python
2.  # -*- coding: utf-8 -*-
3.
4.  from random import choice
```



```

5.
6.
7. def number_in_row(grid, row, number):
8.     """
9.     Chequear si un número se encuentra en la fila
    especificada.
10.    """
11.    return number in grid[row]
12.
13.
14. def number_in_col(grid, col, number):
15.     """
16.     Chequear si un número se encuentra en la columna
    especificada.
17.    """
18.    return number in (row[col] for row in grid)
19.
20.
21. def number_in_box(grid, row, col, number):
22.     """
23.     Chequear si un número se encuentra en la caja a la que
    corresponde la posición especificada.
24.    """
25.
26.    # Obtener la caja a la que pertenece el número.
27.    box_row, box_col = box_by_pos(row, col)
28.    # Construir una lista con los números en la caja.
29.    numbers_in_box = unpack(
30.        row[box_col*3:box_col*3 + 3]
31.        for row in grid[box_row*3:box_row*3 + 3]
32.    )
33.    return number in numbers_in_box
34.
35.
36. def reduce(n):
37.     """
38.     Reducir la posición 9x9 a 3x3.
39.    """
40.    n /= 3
41.    if n == 0 or n != int(n):
42.        n += 1
43.    return int(n)
44.
45.
46. def box_by_pos(row, col):
47.     # Trabajar temporalmente con base 1.
48.     row += 1
49.     col += 1
50.
51.     # Obtener base 0 nuevamente.
52.     return reduce(row) - 1, reduce(col) - 1
53.
54.
55. def unpack(iterable):
56.     """
57.     >>> list(unpack([[1, 2], [3, 4]]))
58.     [1, 2, 3, 4]
59.     """
60.     for item in iterable:
61.         yield from item
62.
63.
64. def get_possible_numbers(grid, row, col):
65.     """
66.     Retorna números posibles para una determinada posición.
67.    """
68.     for number in range(1, 10):

```



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

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

Comentarios recientes

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

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

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

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

```

69.         if (not number_in_row(grid, row, number) and
70.             not number_in_col(grid, col, number) and
71.             not number_in_box(grid, row, col, number)):
72.             yield number
73.
74.
75. def main():
76.     while True:
77.         # Los ceros representan casilleros vacíos.
78.         grid = [
79.             [0, 0, 0, 8, 0, 3, 0, 0, 0],
80.             [0, 1, 0, 9, 0, 0, 7, 0, 8],
81.             [0, 0, 0, 0, 4, 0, 0, 0, 9],
82.             [2, 0, 0, 0, 0, 4, 0, 6, 0],
83.             [4, 0, 7, 0, 0, 0, 9, 0, 5],
84.             [0, 5, 0, 7, 0, 0, 0, 0, 1],
85.             [7, 0, 0, 0, 3, 0, 0, 0, 0],
86.             [3, 0, 6, 0, 0, 8, 0, 7, 0],
87.             [0, 0, 0, 2, 0, 7, 0, 0, 0],
88.         ]
89.
90.         s = \
91.         """\
92.         +-----+
93.         | {} {} {} | {} {} {} | {} {} {} |
94.         | {} {} {} | {} {} {} | {} {} {} |
95.         | {} {} {} | {} {} {} | {} {} {} |
96.         +-----+
97.         | {} {} {} | {} {} {} | {} {} {} |
98.         | {} {} {} | {} {} {} | {} {} {} |
99.         | {} {} {} | {} {} {} | {} {} {} |
100.        +-----+
101.        | {} {} {} | {} {} {} | {} {} {} |
102.        | {} {} {} | {} {} {} | {} {} {} |
103.        | {} {} {} | {} {} {} | {} {} {} |
104.        +-----+
105.        """
106.
107.        while True:
108.            possible_numbers = {
109.                (row, col): None for row in range(9) for
110.                col in range(9)
111.            }
112.
113.            # Obtener una lista de números posibles para
114.            # cada una de
115.            # las posiciones vacías.
116.            for row in range(9):
117.                for col in range(9):
118.                    number = grid[row][col]
119.                    if number == 0:
120.                        options = list(
121.                            get_possible_numbers(grid, row,
122.                                col)
123.                        )
124.                        if options:
125.                            possible_numbers[(row, col)] =
126.                                options
127.
128.            # Remover valores vacíos y ordenar por la
129.            # cantidad de
130.            # posibilidades.
131.            possible_numbers = sorted(
132.                (
133.                    (k, v)
134.                    for (k, v) in possible_numbers.items()
135.                    if v is not None

```

[en Tcl/Tk \(tkinter\).
\(https://recursospython.com/gui-y-manuales/lista-desplegable-combobox-en-tkinter/#comment-2434\)](https://recursospython.com/gui-y-manuales/lista-desplegable-combobox-en-tkinter/#comment-2434)
 Herná en [Lista desplegable \(Combobox\) en Tcl/Tk \(tkinter\).
\(https://recursospython.com/gui-y-manuales/lista-desplegable-combobox-en-tkinter/#comment-2424\)](https://recursospython.com/gui-y-manuales/lista-desplegable-combobox-en-tkinter/#comment-2424)

```

131.         ),
132.         key=lambda kv: len(kv[1])
133.     )
134.
135.     if possible_numbers:
136.         # Obtener el primer item.
137.         (row, col), numbers = possible_numbers[0]
138.         # Fuerza bruta: obtener un número aleatorio
de la
139.         # lista de posibiilidades hasta que la
grilla esté
140.         # completa.
141.         grid[row][col] = choice(numbers)
142.     else:
143.         break
144.
145.     # Chequear si la fuerza bruta dió resultado: si no
hay más
146.     # ceros en la grilla entonces el Sudoku está
resuelto.
147.     if 0 not in unpack(grid):
148.         print(s.format(*unpack(grid)))
149.         break
150.
151.
152. if __name__ == "__main__":
153.     main()

```

Traté de mantener el código lo más funcional, comentado y legible posible. La grilla por defecto representa el siguiente juego de Sudoku:

			8		3			
	1		9			7		8
				4				9
2					4		6	
4		7				9		5
	5		7					1
7				3				
3		6			8		7	
			2		7			

(<https://www.rekursopython.com/wp-content/uploads/2016/12/sudoku.png>)

Y la solución provista por el programa es:

5	4	9	:	8	7	3	:	2	1	6	:
6	1	3	:	9	2	5	:	7	4	8	:
8	7	2	:	1	4	6	:	3	5	9	:
2	3	1	:	5	9	4	:	8	6	7	:
4	6	7	:	3	8	1	:	9	2	5	:
9	5	8	:	7	6	2	:	4	3	1	:
7	2	5	:	6	3	9	:	1	8	4	:
3	9	6	:	4	1	8	:	5	7	2	:
1	8	4	:	2	5	7	:	6	9	3	:

(<https://www.recursospython.com/wp-content/uploads/2016/12/sudoku-solucion.png>)

Para resolver tus propias partidas de Sudoku, simplemente debes editar los elementos de la lista `grid`.

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 `fuerza-bruta` (<https://recursospython.com/tag/fuerza-bruta/>)
`sudoku` (<https://recursospython.com/tag/sudoku/>)

◀ [Formulario de contacto vía CGI](#)

(<https://recursospython.com/codigos-de-fuente/formulario-de-contacto-via-cgi/>)

[Diferencia entre variables en Python y otros lenguajes](#) ▶

(<https://recursospython.com/guias-y-manuales/diferencia-entre-variables-en-python-y-otros-lenguajes/>)

Deja una respuesta

Comentario *

Nombre *

Email *



Publicar el comentario

© 2013 - 2023

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

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

