

# Añadir texto a una imagen con PIL / Pillow

marzo 23, 2016 by Recursos Python [Dejar un comentario](#)

Versión: 2.x, 3.x.

**Pillow** es una librería para el procesamiento de imágenes en Python, aunque mucho más preparada para la lectura y análisis que para la escritura. Sin embargo, permite escribir imágenes en los principales formatos y realizar algunas tareas de dibujo (líneas, formas, texto). La funcionalidad de añadir textos es en sí bastante limitada, pero en la mayoría de los casos suficiente.

Utilizaremos la siguiente imagen `marte.jpg` (¡qué belleza!).



Empezamos por importar las clases necesarias.

```
1. from PIL import Image, ImageDraw, ImageFont
```

Luego, abrir la imagen que deseamos editar y crear una instancia de `ImageDraw` para dibujar sobre la misma.

```
1. image = Image.open("marte.jpg")
2. draw = ImageDraw.Draw(image)
```

Antes de añadir un texto es necesario crear la fuente. PIL utiliza su propio formato de archivos para fuentes de texto, aunque también soporta TrueType y OpenType. Usuarios de Microsoft Windows pueden utilizar, por ejemplo, Arial.

```
1. font = ImageFont.truetype("arial.ttf", 60)
```

El primer argumento indica la ubicación del archivo de fuente. En Windows, si no se especifica una ruta, el archivo será buscado en la carpeta de fuentes del sistema (usualmente `C:\Windows\Fonts`). El segundo argumento establece el tamaño.

En sistemas basados en Unix deberá especificarse la ruta completa. Por ejemplo:

```
1. font = ImageFont.truetype("/usr/share/fonts/truetype/ttf-dejavu/DejaVuSerif.ttf", 60)
```

Una vez cargada la fuente, la función `ImageDraw.Draw.text` inserta el texto en una posición y con un color específicos.

```
1. draw.text((50, 50), "MARTE DESDE EL ESPACIO", font=font, fill="white")
```

El primer argumento establece la posición (x, y) del texto en la imagen. Las coordenadas (0, 0) corresponden a la esquina superior izquierda.

El parámetro `fill` especifica el color, que bien puede ser una tupla con el formato RGBA. Por ejemplo, (255, 255, 255, 255) es equivalente a "white". El cuarto elemento es el valor del canal *alpha* (255 es totalmente opaco, 0 totalmente transparente). La funcionalidad de agregar contorno al texto no es soportada.

Guardamos el archivo como `marte2.jpg`.

```
1. image.save("marte2.jpg")
```



Código completo:

```
1. from PIL import Image, ImageDraw, ImageFont
2.
3. image = Image.open("marte.jpg")
4. draw = ImageDraw.Draw(image)
5. # O bien /usr/share/fonts/truetype/ttf-dejavu/DejaVuSerif.ttf.
6. font = ImageFont.truetype("arial.ttf", 60)
7. draw.text((50, 50), "MARTE DESDE\nEL ESPACIO", font=font, fill="white")
8. image.save("marte2.jpg")
```

Para textos con múltiples líneas, debe utilizarse `multiline_text()`.

```
1. draw.multiline_text((50, 50), "MARTE DESDE\nEL ESPACIO", font=font,
    fill="white")
```



Adicionalmente soporta la alineación del texto vía del parámetro `align`. Debe ser especificado con los valores "left" (izquierda, por defecto), "center" (centrado) o "right" (derecha).

```
1. draw.multiline_text((50, 50), "MARTE DESDE\nEL ESPACIO", font=font,
2. fill="white", align="center")
```



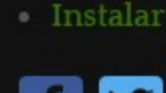
Por último, el siguiente código coloca el texto en el centro de la imagen.

```
1. text = "MARTE DESDE EL ESPACIO\n(Imagen de la NASA)"
2.
3. # Centrar texto vertical y horizontalmente.
4. lines = text.splitlines()
5. w = font.getsize(max(lines, key=lambda s: len(s)))[0]
6. h = font.getsize(text)[1] * len(lines)
7. x, y = image.size
8. x /= 2
9. x -= w / 2
10. y /= 2
11. y -= h / 2
12.
13. draw.multiline_text((x, y), text, font=font, fill="white",
14. align="center")
15. image.save("marte2.jpg")
```



## Artículos relacionados

- [Legofy - Imágenes constituidas por bloques LEGO](#)
- [Instalar PIL / Pillow y aplicar efectos visuales](#)



Entrada publicada en [Guías y Manuales](#) con las etiquetas [pil](#) [pillow](#)

[◀ El módulo estándar textwrap](#)

[Por qué no debes usar Python 2.6 ▶](#)

## Deja un comentario

Comentario

Nombre \*

Email \*

**Publicar el comentario**

Búsqueda ...



## Últimas entradas

[Enumerando colecciones con «enumerate»](#)

[hashlib - Cifrar con los algoritmos MD5 y SHA](#)

[El tipo de dato «None»](#)

[Comprensión de listas y otras colecciones](#)

[No uses triples comillas como comentarios](#)

## Comentarios recientes

Mateo Loaiza Rios en [Argumentos en funciones \(\\*args y \\*\\*kwargs\)](#)  
Recursos Python en [Ejecutar aplicación o script al iniciar Windows](#)  
Marco isidro en [Ejecutar aplicación o script al iniciar Windows](#)  
DCARD kEVEIN en [«python» no se reconoce como un comando interno o externo](#)  
Recursos Python en [Navegador web simple con PyQt 5](#)