

```

pip install requests beautifulsoup4
# -----
import requests
from bs4 import BeautifulSoup
import os
import urllib.parse

def descargar_imagenes(url_pagina: str, carpeta_destino="imagenes_descargadas"):
    """
    Descarga todas las imágenes encontradas en una URL específica a una carpeta
    local.
    """
    print(f"🌐 Procesando URL: {url_pagina}")

    # 1. Crear el directorio de destino si no existe
    if not os.path.exists(carpeta_destino):
        os.makedirs(carpeta_destino)
        print(f"📁 Carpeta de destino creada: {carpeta_destino}")

    try:
        # 2. Obtener el contenido de la página web
        # Añadir un 'User-Agent' simula un navegador y ayuda a evitar ser
        bloqueado
        headers = {'User-Agent': 'Mozilla/5.0'}
        respuesta = requests.get(url_pagina, headers=headers, timeout=10)
        respuesta.raise_for_status() # Lanza una excepción para errores 4xx/5xx

    except requests.exceptions.RequestException as e:
        print(f"❌ Error al acceder a la URL: {e}")
        return

    # 3. Analizar el HTML con BeautifulSoup
    soup = BeautifulSoup(respuesta.text, 'html.parser')

    # Encontrar todas las etiquetas <img>
    etiquetas_img = soup.find_all('img')

    if not etiquetas_img:
        print("🔍 No se encontraron etiquetas <img> en la página.")
        return

    imagenes_descargadas = 0
    print(f"🔍 Encontradas {len(etiquetas_img)} posibles imágenes. Iniciando
    descarga...")

    # 4. Iterar sobre las imágenes y descargarlas
    for i, img in enumerate(etiquetas_img):
        # Obtener la URL de la imagen del atributo 'src'
        src_relativo = img.get('src')

        if not src_relativo:
            continue

        # Construir la URL absoluta:

```

```

        # A veces, el 'src' es relativo (ej: "/images/foto.jpg"), necesitamos la
URL base.
        url_imagen = urllib.parse.urljoin(url_pagina, src_relativo)

        # Generar un nombre de archivo único
        nombre_archivo =
os.path.basename(urllib.parse.urlparse(url_imagen).path)

        # Si el nombre es muy corto (a veces pasa), le damos un nombre
secuencial
        if len(nombre_archivo) < 5:
            nombre_archivo = f"imagen_{i+1}_{os.path.basename(url_imagen)}"

        ruta_completa = os.path.join(carpeta_destino, nombre_archivo)

        try:
            # Descargar el contenido de la imagen
            datos_imagen = requests.get(url_imagen, headers=headers,
stream=True, timeout=5)
            datos_imagen.raise_for_status()

            # Escribir el contenido en un archivo local (modo binario 'wb')
            with open(ruta_completa, 'wb') as f:
                for chunk in datos_imagen.iter_content(chunk_size=8192):
                    f.write(chunk)

            imagenes_descargadas += 1
            print(f"    [+] Descargada: {nombre_archivo}")

        except requests.exceptions.RequestException as e:
            # print(f"    [-] Falló la descarga de {url_imagen}: {e}")
            continue # Simplemente pasa a la siguiente imagen

    print("\n-----")
    print(f"✅ Descarga completada. Total de imágenes guardadas:
**{imagenes_descargadas}**")
    print(f"Los archivos están en la carpeta: **{carpeta_destino}**")
    print("-----")

if __name__ == "__main__":

    # 🚨 NOTA IMPORTANTE:
    # Por favor, usa una URL de prueba o una página que tengas permiso para
raspar.
    # No uses sitios web sin permiso o que puedan bloquear tu IP.

    url_de_prueba = input("Introduce la URL de la página web (ej:
https://ejemplo.com): ")

    if url_de_prueba:
        descargar_imagenes(url_de_prueba)
    else:
        print("Saliendo del programa. No se proporcionó ninguna URL.")

```