# Optilearn.image.web_image_downloader

08 November 2024 13:56

## web_image_downloader

**Description**:
The web_image_downloader function is designed to scrape and download images from a given webpage. It allows users to specify the number of images to download, the file format for saving, and customize how images are chosen (e.g., by indices, excluding certain images, or printing download status). Additionally, the function supports scraping multiple pages of a website and identifying next-page navigation buttons.

**Parameters:**

- **file_path** (str):
  The directory path where images will be saved. Ensure that the path uses forward slashes (e.g., 'C:/downloaded_images/'), and append a trailing slash at the end of the path to avoid errors.
- **url** (str):
  The URL of the webpage to scrape images from. The function will attempt to extract images from this page.
- **img_tag** (str, optional, default='img'):
  The HTML tag used to identify the images. By default, it is set to 'img', which is typically used for image elements on web pages. You may need to customize this if the images are nested in other tags.
- **tag_sc** (str, optional, default='src'):
  The attribute of the image tag that contains the image URL. Typically, this is 'src', but in some cases, it may be 'srcset' or another attribute depending on the website's image loading structure.
- **n_image** (int, optional, default=20):
  The maximum number of images to scrape and download from the page. If you want to download more or fewer images, adjust this parameter accordingly.
- **extention** (str, optional, default='.jpg'):
  The file extension for the saved images. Ensure that it includes a leading dot (e.g., '.jpg' or '.png'). If only 'jpg' or 'png' is provided, it may result in an error.
- **state** (int, optional, default=0):
  The starting index for naming the downloaded images. This is useful if you are repeatedly downloading images to the same directory and want to avoid overwriting files. If you're starting fresh, the default value of 0 is typically used.
- **keep** (str or list, optional, default='all'):
  A list specifying which images to keep, using their indices. If set to 'all', all images from the page are kept. You can also provide a specific list of indices to select particular images (e.g., keep=[0, 2, 5, 8]).
- **exception** (list, optional, default=None):
  A list of indices specifying images to exclude from downloading. If you want to avoid downloading specific images, provide their indices here (e.g., exception=[1, 3, 7]).
- **img_download_status** (bool, optional, default=False):
  If set to True, the function will print the download status for each image, indicating whether the image was successfully downloaded.
- **page_count** (int, optional, default=1):
  Specifies how many pages of the website to scrape. By default, only the first page is scraped. You can set this parameter to scrape additional pages (e.g., page_count=3 to scrape the first three pages).
- **next_button_class** (str, optional, default=None):
  The class name of the "Next" button or link used for pagination. This is useful when scraping multiple pages and needing to click the "Next" button. If the webpage uses a pagination button with a specific class, you should pass this class name here.
- **next_button_tag** (str, optional, default='a'):
  The HTML tag used for the pagination button or link. The default value is 'a', which corresponds to anchor tags commonly used for navigation. If the website uses a different tag (e.g., <div>), you should modify this parameter.

**Returns:**

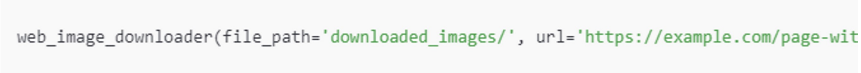
- **tuple**:
  The function returns a tuple containing:
  - The number of images successfully downloaded.
  - A success message indicating the number of images saved.

**Notes:**

- The function scrapes images from the provided webpage based on the specified HTML tag and attribute (img_tag and tag_sc).
- If page_count is greater than 1, the function will scrape multiple pages of the website and continue clicking the "Next" button until all pages have been processed.
- If the img_download_status parameter is set to True, a message will be printed for each image indicating whether it was successfully downloaded or not.
- The keep parameter lets you filter which images to download, while exception helps you exclude specific images from downloading.
- The state parameter is useful when repeatedly downloading images to prevent overwriting files, especially if using the same directory for multiple downloads.

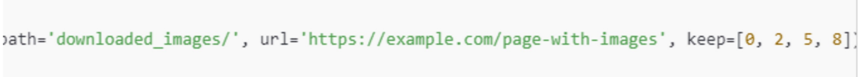
**Examples:**

1. **Download all images from the given URL**:



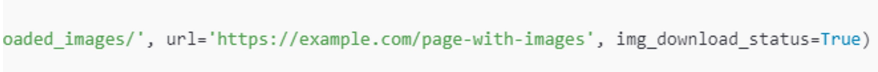
```
web_image_downloader(file_path='downloaded_images/', url='https://example.com/page-wit
```

2. **Download specific images using their indices**:



```
)ath='downloaded_images/', url='https://example.com/page-with-images', keep=[0, 2, 5, 8])
```

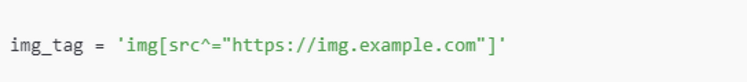
3. **Download images excluding specific indices**:



```
oaded_images/', url='https://example.com/page-with-images', img_download_status=True)
```

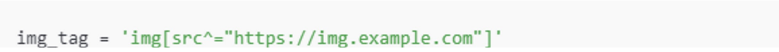
**Tutorials:**

- **Query-1**: *How to get the proper image tag?*
  **Answer**:
  Right-click on any image on the web page and click "Inspect". Locate the <img> tag, and find the src or srcset attribute. This will contain the URL of the image. You can use this URL to identify the appropriate tag and attribute for scraping images.
  **Example**:
  If the src attribute for an image is https://img.example.com/image.jpg, then use:



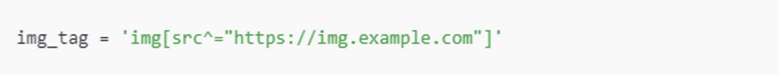
```
img_tag = 'img[src^="https://img.example.com"]'
```

- **Query-2**: *How to write the image tag?*
  **Answer**:
  You should write the img_tag as follows:



```
img_tag = 'img[src^="https://img.example.com"]'
```

**Answer**:
You should write the img_tag as follows:



```
img_tag = 'img[src^="https://img.example.com"]'
```

- **Query-3**: *What should be the ideal value of the tag_sc parameter?*
  **Answer**:
  The tag_sc can usually be 'src' or 'srcset', but the exact value depends on how the images are loaded on the webpage. Experiment with both values to see which one works best for the target website.
- **Query-4**: *What does the state parameter value refer to?*
  **Answer**:
  The state parameter specifies the starting index for naming the downloaded images. This ensures that images are not overwritten when downloading to the same directory multiple times. If unsure, set it to 0 for a fresh start.
- **Query-5**: *How to get the next_button_class?*
  **Answer**:
  Right-click on the "Next" button or link on the webpage and select "Inspect". Locate the class name within the corresponding tag (usually <a>). This is the value you should provide for the next_button_class.