## Contents

1 Installation 3  
1.1 PyPI ................................................. 3  
1.2 Archive ............................................. 3  

2 Compatibility 5  

3 Usage 7  
3.1 Command line ......................................... 7  
3.2 Python Script ........................................... 7  

4 Tag Descriptions 9  

5 Processing Options 11  
5.1 Faster Processing ...................................... 11  
5.2 Stop at a Given Tag ................................... 11  
5.3 Strict Processing ...................................... 11  

Easy to use Python module to extract Exif metadata from tiff and jpeg files.

Originally written by Gene Cash & Thierry Bousch.
1.1 PyPI

The recommended process is to install the PyPI package, as it allows easily staying up to date:

```
$ pip install exifread
```

See the pip documentation for more info.

1.2 Archive

Download an archive from the project’s releases page.

Extract and enjoy.
EXIF.py is tested on the following Python versions:

- 2.6
- 2.7
- 3.2
- 3.3
- 3.4
CHAPTER 3

Usage

3.1 Command line

Some examples:

$ EXIF.py image1.jpg
$ EXIF.py image1.jpg image2.tiff
$ find ~/Pictures -name "*.jpg" -name "*.tiff" | xargs EXIF.py

Show command line options:

$ EXIF.py

3.2 Python Script

import exifread
# Open image file for reading (binary mode)
f = open(path_name, 'rb')

# Return Exif tags
tags = exifread.process_file(f)

Note: To use this library in your project as a Git submodule, you should:
from <submodule_folder> import exifread

Returned tags will be a dictionary mapping names of Exif tags to their values in the file named by path_name. You
can process the tags as you wish. In particular, you can iterate through all the tags with:

for tag in tags.keys():
    if tag not in ('JPEGThumbnail', 'TIFFThumbnail', 'Filename', 'EXIF MakerNote'):
        print "Key: $s, value $s" % (tag, tags[tag])

An if statement is used to avoid printing out a few of the tags that tend to be long or boring.
The tags dictionary will include keys for all of the usual Exif tags, and will also include keys for Makernotes used by
some cameras, for which we have a good specification.

Note that the dictionary keys are the IFD name followed by the tag name. For example:

'EXIF DateTimeOriginal', 'Image Orientation', 'MakerNote FocusMode'
Tag Descriptions

Tags are divided into these main categories:

- **Image**: information related to the main image (IFD0 of the Exif data).
- **Thumbnail**: information related to the thumbnail image, if present (IFD1 of the Exif data).
- **EXIF**: Exif information (sub-IFD).
- **GPS**: GPS information (sub-IFD).
- **Interoperability**: Interoperability information (sub-IFD).
- **MakerNote**: Manufacturer specific information. There are no official published references for these tags.
CHAPTER 5

Processing Options

These options can be used both in command line mode and within a script.

5.1 Faster Processing

Don’t process makernote tags, don’t extract the thumbnail image (if any).
Pass the -q or --quick command line arguments, or as:

```python
tags = exifread.process_file(f, details=False)
```

5.2 Stop at a Given Tag

To stop processing the file after a specified tag is retrieved.
Pass the -t TAG or --stop-tag TAG argument, or as:

```python
tags = exifread.process_file(f, stop_tag='TAG')
```

where TAG is a valid tag name, ex ‘DateTimeOriginal’.

The two above options are useful to speed up processing of large numbers of files.

5.3 Strict Processing

Return an error on invalid tags instead of silently ignoring.
Pass the -s or --strict argument, or as:

```python
tags = exifread.process_file(f, strict=True)
```