

---

# **PhotoScript**

***Release 0.0.2***

**Rhet Turnbull**

**Aug 16, 2020**



# CONTENTS

<b>1</b>	<b>PhotoScript</b>	<b>1</b>
1.1	What is PhotoScript . . . . .	1
1.2	Installation . . . . .	1
1.3	Example . . . . .	1
1.4	See Also . . . . .	2
<b>2</b>	<b>Documentation</b>	<b>3</b>
2.1	photoscript package . . . . .	3
2.1.1	photoscript.photoscript module . . . . .	3
<b>3</b>	<b>Indices and tables</b>	<b>7</b>
	<b>Index</b>	<b>9</b>



## PHOTOSCRIPT

### 1.1 What is PhotoScript

PhotoScript provides a python wrapper around Apple Photos applescript interface. With PhotoScript you can interact with Photos using python. Runs only on MacOS. Tested on MacOS Catalina.

### 1.2 Installation

PhotoScript uses setuptools, thus simply run:

```
python3 setup.py install
```

### 1.3 Example

```
""" Simple example showing use of photoscript """

import photoscript

photoslib = photoscript.PhotosLibrary()

photoslib.activate()
print(f"Running Photos version: {photoslib.version}")

album = photoslib.album("Album1")
photos = album.photos

for photo in photos:
    print(f"{photo.title}, {photo.description}, {photo.keywords}")

new_album = photoslib.create_album("New Album")
photoslib.import_photos(["/Users/rhet/Downloads/test.jpeg"], album=new_album)

photoslib.quit()
```

## 1.4 See Also

- [osxphotos](#): Python package that provides read-only access to the Photos library including all associated meta-data.

## DOCUMENTATION

## 2.1 photoscript package

### 2.1.1 photoscript.photoscript module

```
class photoscript.photoscript.PhotosLibrary
```

```
    activate()
```

activate Photos.app

```
    album(*name, uuid=None)
```

Album instance by name or id

**Parameters**

- **name** – name of album
- **uuid** – id of album

**Returns** Album object or None if album could not be found

Raises: ValueError if both name and id passed. Must pass only name or id but not both.

```
property album_names
```

List of album names in the Photos library

**Parameters** **top\_level** – if True, returns only top-level albums otherwise also returns albums in sub-folders; default is False

```
property albums
```

list of Album objects for all albums

```
create_album(name)
```

creates a top level album

**Returns** Album object for newly created album

```
delete_album(album)
```

deletes album (but does not delete photos in the album)

**Parameters** **album** – an Album object for album to delete

```
property favorites
```

Album object for the Favorites album

```
property folder_names
```

List of folder names in the Photos library

**Parameters** **top\_level** – if True, returns only top-level folders otherwise also returns sub-folders; default is False

**property** **frontmost**

True if Photos.app is front most app otherwise False

**import\_photos** (*photo\_paths*, *album=None*, *skip\_duplicate\_check=False*)

import photos

**Parameters**

- **photos** – list of file paths to import
- **album** – optional, name of album to import into (album must exist in Photos). May also be an Album object.
- **skip\_duplicate\_check** – if True, Photos will not check for duplicates on import, default is False

**property** **name**

name of Photos.app

**photos** (*search=None*, *uuid=None*)

List of Photo objects for items in the library Note: for a large library, calling photos() may run a *very* long time (minutes)

**Parameters**

- **search** – optional text string to search for (returns matching items)
- **uuid** – list of UUIDs to get
- **may pass search or uuid but not both** (*you*) –

**Returns** List of Photo objects or [] if no photos found

Raises: ValueError if both search and uuid are passed

**quit** ()

quit Photos.app

**property** **selection**

List of Photo objects for currently selected photos or [] if no selection

**property** **version**

version of Photos.app

**class** photoscript.photoscript.**Album** (*uuid*)

**add** (*photos*)

add photos from the library to album

**Parameters** **photos** – list of Photo objects to add to album

**export** (*path*, *original=True*, *edited=True*, *timeout=120*)

Export photos in album to path

**Parameters**

- **path** – path to export to
- **original** – if True exports original photo
- **edited** – if True, exports edited version, if one exists
- **timeout** – number of seconds to timeout waiting for Photos to respond



**Returns** list of names of exported photos

```
import_photos (photo_paths, skip_duplicate_check=False)  
import photos
```

**Parameters**

- **photos** – list of file paths to import
- **skip\_duplicate\_check** – if True, Photos will not check for duplicates on import, default is False

**property name**  
name of album

**property parent\_id**  
parent container id

**property photos**  
list of Photo objects for photos contained in album

**property title**  
title of album (alias for Album.name)

**property uuid**  
UUID of Album

```
class photoscript.photoscript.Photo (uuid)
```

**property date**  
date of photo as timezone-naive datetime.datetime object

**property description**  
description of photo

**duplicate** ()  
duplicates the photo and returns Photo object for the duplicate

**export** (*path*, *original=True*, *edited=True*, *timeout=120*)  
Export photo

**Parameters**

- **path** – path to export to
- **original** – if True exports original photo
- **edited** – if True, exports edited version, if one exists
- **timeout** – number of seconds to timeout waiting for Photos to respond

**Returns** name of exported photo

**property filename**  
filename of photo

**property keywords**  
list of keywords for photo

**property name**  
name of photo

**property title**  
title of photo (alias for name)

**property uuid**  
UUID of Photo

## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`



## INDEX

### A

`activate()` (*photoscript.photoscript.PhotosLibrary method*), 3  
`add()` (*photoscript.photoscript.Album method*), 4  
`Album` (*class in photoscript.photoscript*), 4  
`album()` (*photoscript.photoscript.PhotosLibrary method*), 3  
`album_names()` (*photoscript.photoscript.PhotosLibrary property*), 3  
`albums()` (*photoscript.photoscript.PhotosLibrary property*), 3

### C

`create_album()` (*photoscript.photoscript.PhotosLibrary method*), 3

### D

`date()` (*photoscript.photoscript.Photo property*), 5  
`delete_album()` (*photoscript.photoscript.PhotosLibrary method*), 3  
`description()` (*photoscript.photoscript.Photo property*), 5  
`duplicate()` (*photoscript.photoscript.Photo method*), 5

### E

`export()` (*photoscript.photoscript.Album method*), 4  
`export()` (*photoscript.photoscript.Photo method*), 5

### F

`favorites()` (*photoscript.photoscript.PhotosLibrary property*), 3  
`filename()` (*photoscript.photoscript.Photo property*), 5  
`folder_names()` (*photoscript.photoscript.PhotosLibrary property*), 3  
`frontmost()` (*photoscript.photoscript.PhotosLibrary property*), 4

### I

`import_photos()` (*photoscript.photoscript.Album method*), 5  
`import_photos()` (*photoscript.photoscript.PhotosLibrary method*), 4

### K

`keywords()` (*photoscript.photoscript.Photo property*), 5

### N

`name()` (*photoscript.photoscript.Album property*), 5  
`name()` (*photoscript.photoscript.Photo property*), 5  
`name()` (*photoscript.photoscript.PhotosLibrary property*), 4

### P

`parent_id()` (*photoscript.photoscript.Album property*), 5  
`Photo` (*class in photoscript.photoscript*), 5  
`photos()` (*photoscript.photoscript.Album property*), 5  
`photos()` (*photoscript.photoscript.PhotosLibrary method*), 4  
`PhotosLibrary` (*class in photoscript.photoscript*), 3

### Q

`quit()` (*photoscript.photoscript.PhotosLibrary method*), 4

### S

`selection()` (*photoscript.photoscript.PhotosLibrary property*), 4

### T

`title()` (*photoscript.photoscript.Album property*), 5  
`title()` (*photoscript.photoscript.Photo property*), 5

### U

`uuid()` (*photoscript.photoscript.Album property*), 5  
`uuid()` (*photoscript.photoscript.Photo property*), 5

## V

`version()` (*photoscript.photoscript.PhotosLibrary*  
*property*), [4](#)