

---

# **gedcompy Documentation**

***Release 0.1***

**Rory McCann**

February 12, 2017



<b>1</b>	<b>Reading GEDCOM files</b>	<b>1</b>
<b>2</b>	<b>Writing GEDCOM files</b>	<b>3</b>
<b>3</b>	<b>gedcompy</b>	<b>5</b>
3.1	Example Usage . . . . .	5
3.2	Contributing . . . . .	5
<b>4</b>	<b>Indices and tables</b>	<b>7</b>



---

## Reading GEDCOM files

---

The function `gedcom.parse()` reads a file, string, or file-like object and returns a `gedcom.GedcomFile`.

`gedcom.parse(obj)`

Parse and return this object, if it's a file.

If it's a filename, it calls `parse_filename()`, for file-like objects, `parse_fp`, for strings, calls `parse_string`.

**Parameters** `obj` – filename, open file-like object or string contents of GEDCOM file

**Returns** `GedcomFile`



---

## Writing GEDCOM files

---

*gedcom.GedcomFile.save()* saves a `gedcom.GedcomFile` to a specified filename, or file-like object.

`GedcomFile.save(fileout)`

Saves the contents of this GEDCOM file to specified filename or file-like object.

**Parameters** `fileout` – Filename or open file-like object to save this to.

**Raises** `Exception` – if the filename exists





---

## gedcompy

---

Python library to parse and work with **GEDCOM** (geneology/family tree) files.

It's goal is to support GEDCOM v5.5 ([specification here](#)).

This is released under the GNU General Public Licence version 3 (or at your option, a later version). See the file *LICENCE* for more.

### 3.1 Example Usage

```
>>> import gedcom
>>> gedcomfile = gedcom.parse("myfamilytree.ged")
>>> for person in gedcomfile.individuals:
...     firstname, lastname = person.name
...     print "{0} {1} is in the file".format(firstname, lastname)
```

### 3.2 Contributing

Run all unittests with *tox*.



---

## Indices and tables

---

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



## P

`parse()` (in module `gedcom`), 1

## S

`save()` (`gedcom.GedcomFile` method), 3