
rsc-on-this-day

Release 0.4.0

Displays Royal Society of Chemistry “On This Day” facts.

Dominic Davis-Foster

Jun 13, 2021

Contents

1	Installation	1
1.1	from PyPI	1
1.2	from GitHub	1
2	Usage	3
2.1	rsc-on-this-day	3
2.2	Examples	3
2.3	Adding to ~/.bashrc	4
3	API Reference	5
3.1	clear_cache	5
3.2	get_fact	5
4	Contributing	7
4.1	Coding style	7
4.2	Automated tests	7
4.3	Type Annotations	7
4.4	Build documentation locally	8
5	Downloading source code	9
5.1	Building from source	10
	Python Module Index	11
	Index	13

Installation

1.1 from PyPI

```
$ python3 -m pip install rsc_on_this_day --user
```

1.2 from GitHub

```
$ python3 -m pip install git+https://github.com/domdfcoding/rsc-on-this-day@master --user
```

Once installed, `rsc_on_this_day` can be run by typing:

```
$ rsc_on_this_day
```

If `rsc_on_this_day` is not installed in a directory in `$PATH`, you may need to add `~/.local/bin/` to your `$PATH`.

Usage

2.1 rsc-on-this-day

Display the Royal Society of Chemistry “On This Day In Chemistry” fact for the given day.

If no date is given the current date is used.

```
rsc-on-this-day [OPTIONS] [MONTH] [DAY]
```

Options

--version

Show the version and exit.

--clear-cache

Clear any cached data and exit.

-w, --width <width>

The number of characters per line of the output. Set to -1 to disable wrapping.

Default 80

Arguments

MONTH

Optional argument. Default None

DAY

Optional argument. Default None

2.2 Examples

```
$ rsc_on_this_day
```

- Display the “On This Day In Chemistry” fact for today.

```
$ rsc_on_this_day Apr 1
```

- Display the “On This Day In Chemistry” fact for April 1st.

```
$ rsc_on_this_day 12 25
```

- Display the “On This Day In Chemistry” fact for 25 December.

```
$ rsc_on_this_day --clear-cache
```

- Clear any cached data.

```
$ rsc_on_this_day October 13 --width 80
```

- Display the “On This Day In Chemistry” fact for October 13th, with at most 80 characters per line.

2.3 Adding to ~/.bashrc

`rsc-on-this-day` can be run every time you open a terminal by adding `rsc-on-this-day` to your `~/.bashrc` file. For example:

```
$ echo "rsc-on-this-day" >> ~/.bashrc
```


API Reference

Displays Royal Society of Chemistry “On This Day In Chemistry” facts in your terminal.

Functions:

<code>clear_cache()</code>	Clear any cached responses.
<code>get_fact([month, day])</code>	Returns the fact for the given date.

clear_cache()

Clear any cached responses.

Return type `int`

get_fact (*month=None, day=None*)

Returns the fact for the given date.

Parameters

- **month** (`Union[str, int, None]`) – The month, either its short name (e.g. 'Oct '), its full name (e.g. 'October ') or its number (e.g. 10). Default `None`.
- **day** (`Union[str, int, None]`) – The day of the month. Default `None`.

If `month` and `day` are both left as `None` (the default) the current date is used.

Return type `Tuple[str, str]`

Contributing

`rsc_on_this_day` uses `tox` to automate testing and packaging, and `pre-commit` to maintain code quality.

Install `pre-commit` with `pip` and install the git hook:

```
$ python -m pip install pre-commit
$ pre-commit install
```

4.1 Coding style

`formate` is used for code formatting.

It can be run manually via `pre-commit`:

```
$ pre-commit run formate -a
```

Or, to run the complete autoformatting suite:

```
$ pre-commit run -a
```

4.2 Automated tests

Tests are run with `tox` and `pytest`. To run tests for a specific Python version, such as Python 3.6:

```
$ tox -e py36
```

To run tests for all Python versions, simply run:

```
$ tox
```

4.3 Type Annotations

Type annotations are checked using `mypy`. Run `mypy` using `tox`:

```
$ tox -e mypy
```

4.4 Build documentation locally

The documentation is powered by Sphinx. A local copy of the documentation can be built with `tox`:

```
$ tox -e docs
```

Downloading source code

The `rsc_on_this_day` source code is available on GitHub, and can be accessed from the following URL: <https://github.com/domdfcoding/rsc-on-this-day>

If you have `git` installed, you can clone the repository with the following command:

```
$ git clone https://github.com/domdfcoding/rsc-on-this-day
```

```
Cloning into 'rsc-on-this-day'...
remote: Enumerating objects: 47, done.
remote: Counting objects: 100% (47/47), done.
remote: Compressing objects: 100% (41/41), done.
remote: Total 173 (delta 16), reused 17 (delta 6), pack-reused 126
Receiving objects: 100% (173/173), 126.56 KiB | 678.00 KiB/s, done.
Resolving deltas: 100% (66/66), done.
```

Alternatively, the code can be downloaded in a ‘zip’ file by clicking:

Clone or download → Download Zip

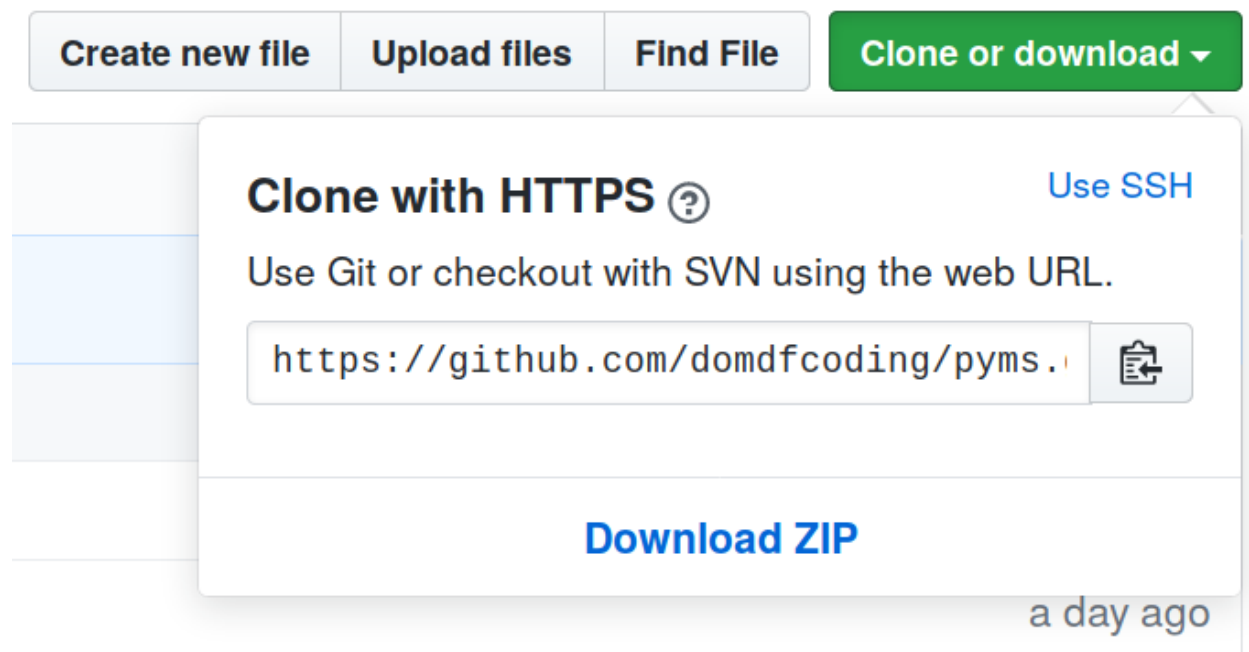


Fig. 1: Downloading a ‘zip’ file of the source code

5.1 Building from source

The recommended way to build `rsc_on_this_day` is to use `tox`:

```
$ tox -e build
```

The source and wheel distributions will be in the directory `dist`.

If you wish, you may also use `pep517.build` or another **PEP 517**-compatible build tool.

Python Module Index

r

`rsc_on_this_day`, 5

Symbols

`--clear-cache`
 `rsc-on-this-day` command line
 option, 3

`--version`
 `rsc-on-this-day` command line
 option, 3

`--width <width>`
 `rsc-on-this-day` command line
 option, 3

`-w`
 `rsc-on-this-day` command line
 option, 3

`-w`, 3
`DAY`, 3
`MONTH`, 3

C

`clear_cache()` (*in module `rsc_on_this_day`*), 5

D

`DAY`
 `rsc-on-this-day` command line
 option, 3

G

`get_fact()` (*in module `rsc_on_this_day`*), 5

M

module
 `rsc_on_this_day`, 5

`MONTH`
 `rsc-on-this-day` command line
 option, 3

P

Python Enhancement Proposals
 PEP 517, 10

R

`rsc_on_this_day`
 module, 5

`rsc-on-this-day` command line option
 `--clear-cache`, 3
 `--version`, 3
 `--width <width>`, 3