2025 Python Packaging Survey is now live! Take the survey now

Type '/' to search projects

Help Docs Sponsors Log in Register

# supermemo2 3.0.1

Latest version

```
pip install supermemo2
```

Released: Jun 22, 2024

Implemented the SM-2 algorithm for spaced repetition learning.

2025 Python Packaging Survey is now live! Take the survey now

Project description

Release history

Download files

**Verified details**

*These details have been verified by PyPI*

**Maintainers**

alankan886

**Unverified details**

*These details have **not** been verified by PyPI*

**Project links**

Homepage

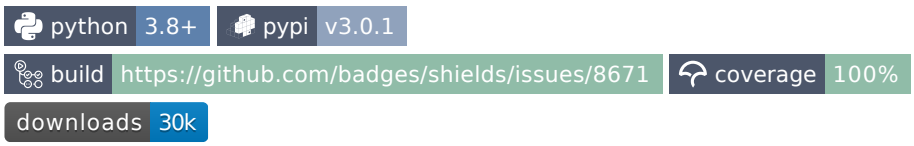
**Meta**

- **License:** MIT License (MIT)
- **Author:** Alan Kan
- spaced-repetition , SM-2 , SuperMemo , Python

**Classifiers**

**Development Status**

# SuperMemo2




A package that implemented the spaced repetition algorithm SM-2 for you to quickly calculate your next review date for whatever you are learning.

📌 **Note:** The algorithm SM-2 doesn't equal to the computer implementation SuperMemo2. In fact, the 3 earliest implementations (SuperMemo1, SuperMemo2 and SuperMemo3) all used algorithm SM-2. I didn't notice that when I first published the package on PyPI, and I can't change the package name.

📦 PyPI page

## Table of Contents



## Motivation

The goal was to have an efficient way to calculate the next review date for studying/learning. Removes the burden of remembering the algorithm, equations, and math from the users.

2025 Python Packaging Survey is now live! Take the survey now ×

**Intended Audience**
- Education

**License**
- OSI Approved :: MIT License

**Operating System**
- OS Independent

**Programming Language**
- Python
- Python :: 3

Report project as malware

## Package Install

Install and upate the package using pip:

```
pip install -U supermemo2
```

## To Play Around with the Code

Download the code:

```
git clone https://github.com/alankan886/SuperMemo2.git
```

Install dependencies to run the code:

```
pip install -r requirements.txt
```

supermemo2 supports Python 3.8+

## A Simple Example

```
from supermemo2 import first_review, review

# first review
# using quality=4 as an example, read below for what each value fr
# review date would default to datetime.utcnow() (UTC timezone) if
first_review = first_review(4, "2024-06-22")
# first_review prints { "easiness": 2.36, "interval": 1, "repetiti

# second review
second_review = review(4, first_review["easiness"], first_review["
# or just unpack the first review dictionary
second_review = review(4, **first_review)
# second_review prints similar to example above.
```

💬 2025 Python Packaging Survey is now live! Take the survey now ✕

📣 Calculates the review date of the task following the SM-2 algorithm.
📣 The first_review method to calculate the review date at ease without having to know the initial values.

## What is SM-2?

🎥 If you are curious of what spaced repetition is, check this short video out.

📌 A longer but interactive article on spaced repetition learning.

📎 The SM-2 Algorithm

### What are the "values"?

The values are the:

- Quality: The quality of recalling the answer from a scale of 0 to 5.
  - 5: perfect response.
  - 4: correct response after a hesitation.
  - 3: correct response recalled with serious difficulty.
  - 2: incorrect response; where the correct one seemed easy to recall.
  - 1: incorrect response; the correct one remembered.
  - 0: complete blackout.
- Easiness: The easiness factor, a multipler that affects the size of the interval, determine by the quality of the recall.
- Interval: The gap/space between your next review.
- Repetitions: The count of correct response (quality >= 3) you have in a row.

## Code Reference

**first_review**( quality, review_datetime=None**)**

2025 Python Packaging Survey is now live! Take the survey now

dictionary containing the new values.

**Parameters:**

- quality (int) - the recall quality of the review.
- review_datetime (str or datetime.datetime) - optional parameter, the datetime in ISO format up to seconds in UTC timezone of the review.

**Returns:** dictionary containing values like quality, easiness, interval, repetitions and review_datetime.

**Return Type:** Dict

**Usage:**

```
from supermemo2 import first_review
# using default datetime.utcnow() if you just reviewed it
first_review(3)

# providing string date in Year-Month-Day format
first_review(3, "2024-06-22")

# providing date object date
from datetime import datetime
d = datetime(2024, 1, 1)
first_review(3, d)
```

**review(** quality, easiness, interval, repetitions, review_datetime=None **)**

Calcualtes the next review date based on previous values, and returns a dictionary containing the new values.

**Parameters:**

- quality (int) - the recall quality of the review.
- easiness (float) - the easiness determines the interval.
- interval (int) - the interval between the latest review date and the next review date.
- repetitions (int) - the count of consecutive reviews with quality

2025 Python Packaging Survey is now live! Take the survey now

the datetime in ISO format up to seconds in UTC timezone of the review.

**Returns:** dictionary containing values like quality, easiness, interval, repetitions and review_datetime.

**Return Type:** Dict

**Usage:**

```
from supermemo2 import first_review, review
# using previous values from first_review call
r = first_review(3)

# using default datetime.utcnow() if you just reviewed it
review(3, r["easiness"], r["interval"], r["repetitions"])

# providing review_datetime from previous review
review(3, r["easiness"], r["interval"], r["repetitions"], r["revie

# providing string review_datetime
review(3, r["easiness"], r["interval"], r["repetitions"], "2024-01

# providing datetime object review_datetime
from datetime import datetime
d = datetime(2024, 1, 1)
review(3, r["easiness"], r["interval"], r["repetitions"], d)
```

## Testing

Assuming you dowloaded the code and installed requirements.

### Run the tests

```
pytest tests/
```

### Check test coverages

```
pytest --cov
```

2025 Python Packaging Survey is now live! Take the survey now

## Changelog

3.0.1 (2024-06-22): Minor changes, Update recommended

- Forgot to update some code and tests from review_date to review_datetime, the returned dictionary was review_date instead review_datetime.

3.0.0 (2024-06-22): Major changes/rebuild, Update recommended

- Rewrote the code to remove the class structure, simplfying the code and usability.
- Update to provide datetime instead of just date, more specific with when to review.

2.0.0 (2021-03-28): Major changes/rebuild, Update recommended

- Rebuilt and simplfied the package.

1.0.3 (2021-01-30): Minor bug fix, Update recommended

- Re-evaluate the default date argument to first_review() on each call.

1.0.2 (2021-01-18): Major and Minor bug fix, Update recommended

- Add required attrs package version to setup.py.
- Allow users to access SMTwo model.
- Fix E-Factor calculation when q < 3.

1.0.1 (2021-01-02): Fix tests, update README and add Github actions, Update not required

- Add missing assertions to test_api.py.
- Update README badges and fix format.
- Add Github actions to run tests against Python versions 3.6 to 3.9 in different OS, and upload coverage to Codecov.

2025 Python Packaging Survey is now live! Take the survey now

- Provide API methods to quickly access the SMTwo class.
- Develop 100% coverage integration and unit tests in a TDD manner.
- Write new documentation.

0.1.0 (2020-07-14): Add tests, Update not required

- Add passing unit tests with a coverage of 100%.

0.0.4 (2020-07-10): Minor bug fix, Update recommended

- Fix interval calculation error when q < 3.

0.0.3 (2020-07-06): Documentation Update, Update not required

- Add new section about SM-2 in documentation, and fix some formats in README.

0.0.2 (2020-07-05): Refactor feature, Update recommended

- Refactor the supermemo2 algorithm code into a simpler structure, and remove unnecessary methods in the class.

0.0.1 (2020-07-02): Feature release

- Initial Release

## Credits

1. pytest
2. The SM-2 Algorithm

2025 Python Packaging Survey is now live! Take the survey now

Installing packages
Uploading packages
User guide
Project name retention
FAQs

PyPI Blog
Infrastructure dashboard
Statistics
Logos & trademarks
Our sponsors

## Contributing to PyPI

Bugs and feedback
Contribute on GitHub
Translate PyPI
Sponsor PyPI
Development credits

## Using PyPI

Terms of Service
Report security issue
Code of conduct
Privacy Notice
Acceptable Use Policy

Status: All Systems Operational

Developed and maintained by the Python community, for the Python community.
Donate today!

"PyPI", "Python Package Index", and the blocks logos are registered trademarks of the Python Software Foundation.

© 2025 Python Software Foundation
Site map

**Switch to desktop version**

English español français 日本語 português (Brasil) українська Ελληνικά Deutsch 中文 (简体) 中文 (繁體) русский עברית Esperanto 한국어