

# Python Nuitka support for PyPI top 50

## About me

---

1. Name: Taofeng Li (Tommy), GitHub: [tommyli3318](#)
2. School: University of California, Irvine (United States), Sophomore/Junior in Computer Science, graduation in 2021
3. Time Zone: Pacific Standard Time (PST)
4. [Link to Resume](#)

## Code contribution

---

[Pull Request: run\\_all.py new special-comment mechanism & Urllib3Using.py](#)

[Pull Request: Standalone tests for botocore & boto3](#)

## Project information

---

1. Sub-org name: Nuitka
2. Project Abstract:

Nuitka is a Python compiler written in Python. It is a seamless replacement or extension to the Python interpreter and compiles every construct that CPython does. Nuitka works by translating Python code into a C level program which can be executed in the same way as CPython using libpython and a few C files of its own. All optimizations of Nuitka are aimed at improving performance while ensuring perfect compatibility. This project ensures Nuitka's compatibility with the top 50 PyPI packages by setting up automated testing for each package. These automated tests will serve to be very important tools for the development of Nuitka, as they will be used to assure Nuitka's compatibility with the most used Python packages every time Nuitka receives a new update.

3. Detailed description:

Create a test module for each of the top 50 PyPI packages according to [this list](#), referring to their respective documentations. Use Nuitka to compile the test module in standalone

mode with the Nuitka argument --standalone, then run the executable inside the compiled folder. If the executable runs without error, integrate the test into a reusable testing module which can be rerun anytime in the future as Nuitka gets updated. If the executable runs into errors, report the issues and implement tools for tracking down the issue (narrow down to the root of the problem). If possible, try to fix the issues if they are on Nuitka's end, or report the issues to the respective package organization if the issues are on their end. The ultimate goal of the project is to develop a reusable testing tool for the 50 most popular PyPI packages which can be used to ensure Nuitka's compatibility with each. These automated testing tools will make the future development of Nuitka easier, thus benefiting Nuitka and the Nuitka community as a whole.

#### 4. Weekly timeline

- Community Bonding (May 7-26): Get to know all my mentors, finalize and update the weekly schedule below as I see fit. Familiarize myself with Nuitka's features, ask questions if clarifications are needed. Possibly start early on the work for week 1.
- Week 1 (May 27-31): Implement testing for the following packages - botocore, six, and python-dateutil. Ask mentors for help if needed, have mentors check over the finished deliverable. (10% completion)
- Week 2 (June 3): Implement testing for the following packages - s3transfer, pyyaml, docutils, and pyasn1. (20% completion)
- Week 3 (June 10): Will probably be mostly busy with final exams this week, but will try to review all progress up to this point and ask for constructive feedback from mentors.
- Week 4 (June 17): Implement testing for the following packages - requests, jmespath, certifi, setuptools, and rsa. (30% completion)
- Week 5 (June 24): Implement testing for the following packages - awscli, futures, idna, colorama, and chardet. (40% completion)
- Week 6 (July 1): Implement testing for the following packages - wheel, simplejson, boto3, pytz, and numpy. Continue to review with mentors and apply suggestions for the all the work up until this point. (50% completion)
- Week 7 (July 8): Implement testing for the following packages - markupsafe, cryptography, awscli-cwlogs, jinja2, and cffi. (60% completion)

- Week 8 (July 15): Implement testing for the following packages - protobuf, enum34, pycparser, asn1crypto, and ipaddress. (70% completion)
- Week 9 (July 22): Implement testing for the following packages - click, attrs, pytest, future, and pbr. (80% completion)
- Week 10 (July 29): Implement testing for the following packages - decorator, pandas, pyparsing, werkzeug, and psutil. (90% completion)
- Week 11 (August 5): Implement testing for the following packages - flask, google-api-core, virtualenv, itsdangerous, and py. (~100% completion)
- Week 12 (August 12): Leaving this week for possible roadblocks and changes in schedule. Check over all of my code by running automated testing. Fix unexpected errors if any. Do any other tasks as needed.
- Final week (August 19): Let mentors review all of my work and finalize the project before submitting the official version.

## **Other commitments**

---

- Spring quarter at my university ends on June 14th, and I will be busy with finals for the week of June 10-14th
- After June 14th, my entire summer will be free and dedicated to this project