GSoC Proposal:
Improving Python's web interpreter and more interactive exercises

Organization: Python

Sub-organization: Python Argentina (PyAr)

Project: PyZombis

Applicant's information

Hello, my name is Angela Remolina. I am a colombian 3rd-year university student. I'm currently pursuing a degree in Systems and Computing Engineering.

I have used Python for Data Science in the past. In one of my projects, we performed a sentiment analysis tool for user's reviews in an e-commerce website. The implementation included: web-scraping, large dataset manipulation, data wrangling processes and backend/frontend development. We used the Python libraries: pandas, seaborn, nltk and scikit-learn. Additionally, I have in my tech stack: Flask, MongoDB, Keras, Javascript and JSF.

You can find all my detailed information below:

- Name: Angela Remolina
- Github: angelasofiaremolinagutierrez
- University: Universidad Pontificia Bolivariana (UPB), Colombia
- Degree: Systems & Computing Engineering (2019 - 2023)
- Time zone: GMT-5 (Colombia)

Code contribution

I created a Pull Request that adds all exercises from "Lista de Exercícios I Python para Zombis.pdf" with unit tests for each exercise. I also fixed some small bugs that were happening due to indentation issues. The link to my code contribution is:

https://github.com/PyAr/PyZombis/pull/69

Proposed project

Project Abstract

[PyZombis](#) is a community course to teach Python to the spanish-speaking community. It is based on a brazilian MOOC [Python para Zumbis](#). The idea is to make an improved version of the course in spanish, with an interactive environment for the students, that allows them to visualize and try active code on the site (made possible thanks to the Runestone library). Currently PyZombis is using [Skulpt](#) as its Python's online interpreter. It has not been possible to put some of the advance PyZombis exercises interactive on the web for the students because of Skulpt limitations. It could be improved by changing the interpreter to Brython. This interpreter offers more tools as the scripting language for the Web. This is where I want to help! Also, as an extra, I can offer my help in other minor fixes in the aesthetic of the course to make it more attractive and easy-to-use for students.

Detailed description

My proposal revolves around the 2+1 following milestones:

1. Improve Python's web interpreter

PyZombis' current web interpreter, [Skulpt](#), has blocked the implementation of some complex exercises due to of its limitations with some libraries (see issue [#61](#)). These reasons being on top of the fact that Skulpt is based on Python 2.x, a version [sunsetted](#) by the Python Software Foundation (PSF) since January 1, 2020.

My proposal is then to explore the possibility of changing the interpreter to [Brython](#), which supports the latest stable releases of Python (plus main libraries). It also supports AJAX and access to the DOM elements, which will allow us to create more advanced exercises. This change would be then highly coupled with our use of the runestone library. PyZombis is using version 4.2.2 (and they are in 5.6.0), which also forces Python to not be in its latest stable release. So these changes need to be properly analyzed and carefully implemented.

While in the process I will make sure to document the tech stack and development environment, which will also guide future contributors. Finally, these upgrades would imply a change in the current CI/CD workflow as well. So I will perform an implementation of the new GitHub actions pipeline. While doing so, we could also solve some errors that are currently being raised when new open source contributors create branches or forks due to permissions (refer to Fig. 1).

```
✓  ❌ Publish generated content to GitHub Pages

  1  ▶ Run peaceiris/actions-gh-pages@v3
 15  [INFO] Usage https://github.com/peaceiris/actions-gh-pages#readme
 16  ▶ Dump inputs
 35  ▶ Setup auth token
 37  ▶ Prepare publishing assets
500  ▶ Setup Git config
506  ▶ Create a commit
514  ▼ Push the commit or tag
515  /usr/bin/git push origin gh-pages
516  remote: Permission to PyAr/PyZombis.git denied to github-actions[bot].
517  fatal: unable to access 'https://github.com/PyAr/PyZombis.git/': The requested URL returned error: 403
518  Error: Action failed with "The process '/usr/bin/git' failed with exit code 128"
```

Figure 1. Screenshot from my PR, where GitHub Actions' build step failed due to git errors. Taken from [GitHub Actions · PyAr/PyZombis@1e8865a]

2. Implementation of interactive exercises

I would like to continue with the process of expanding the core content of PyZombis (as I did it my pull request #69). There are still multiple active code exercises to implement. Also, they can grow in complexity due to the capabilities given by the previous milestone.

This is a task that is easily parallelizable among other contributors, so in addition to this, and in light of open source orchestration, I think that if we create itemized GitHub issues (and milestones) we can track our process in content completion. That way, new contributors can, easily and parallely, select and implement new exercises. All while avoiding accidental duplicated work from multiple contributors.

3. [Future Work] Revamp of the User Experience and Interfaces (UX/UI)

When looking at the original PyZombis course, we can see that the UI has a tailored and guided experience for new learners (see Fig. 2). I would like to design a frontend that also conveys a learning path for students, instead of displaying runestone's default layout (see Fig. 3). However, I am proposing this improvement of student's UX for after GSoC. I am writing it in this proposal as future work that will grow on top of milestones #1 and #2.

Figure 2. Screenshot from original course. Taken from [Python para Zumbis course]

Figure 3. Screenshot from running version. Taken from [PyZombis course]

Weekly timeline

Here is the timeline I propose week to week for the community bonding period and the coding period. This schedule can be adjusted if needed. As GSoC this year is shorter due to the pandemic, I planned for the activities to be achievable and modular (divided into the three milestones proposed above). So, we could select a subset of them and still have an enclosed project. I also added a two-week time slot, in case any of the tasks overflows in the planning. Since some activities are dependent on the discussions with mentors, that time helps to cover possible delays.

Week	Activities	Deliverables
Community Bonding (< June 7)	● Familiarize myself with the current python interpreter (Skulpt) and its relation to runestone. ● Analyze with the mentors the possible migration to Brython. ● Analyze with the mentors the possible upgrade of runestone and its compatibility with Brython.	Documentation of learnings and implications of changes.
Week 1 (June 7-13)	● Begin implementation of Brython's component.	PR with initial code structure. PRs with new web interpreter changes
Week 2 (June 14)	● Continue implementation of new interpreter.	PRs with new web interpreter changes
Week 3 (June 21)	● Continue implementation of new interpreter.	PRs with new web interpreter changes
Week 4 (June 28)	● Finish implementation of new interpreter and its junction with current active code content ● Familiarize myself with GitHub Actions' capabilities and our current workflow.	PRs with new web interpreter changes PR that handles new versionsing (if required) + its respective documentation update.
Week 5 (July 5)	● Discuss the workflow updates with mentors and implement necessary changes. ● Document the setup process as a step to step guide for new contributors.	PR that changes `python-app.yml` and its respective documentation update. Documentation instructions on the repository's README or wiki that guides new contributors to: - set up a correct local

		development environment. - build and run the project. - understand CI/CI pipeline - learn about preferred code practices and style - know the recommended tools for the project (e.g. IDE)
Week 6 (July 12)	● Analyze current status of content available. ● Start the implementation of more interactive exercises.	GitHub issues that are correctly classified (with tags and milestones) so that open source contributors can add more exercises. PRs of each new interactive exercise added.
Week 7 (July 19)	● Continue the implementation of more interactive exercises.	PRs of each new interactive exercise added.
Week 8 (July 26)	● Finish the implementation of more interactive exercises.	PRs of each new interactive exercise added.
Week 9 (August 2)	● Wrapping up with documentation of the work done.	Final documentation report in the mentors' preferred format.
Week 10 (August 9)	● Time slot in case of any unpredictable delay.	-
Week 11 - Final week (August 16)	● Time slot in case of any unpredictable delay.	-

Other commitments

The final weeks of GSoC coincide with the beginning of my academic semester. However, it is not an impediment for me. I am fully committed to the deliverables above and the first weeks of classes are not that time demanding. So I could continue my work in PyZombis and finish the project seamlessly. I have this experience as a priority.