

# Python API for visualisation of sets

Nathan John Crowley

Department of Computer Science

BSc Computer Science

Dr Rosane Minghim

University College Cork

April 2022

# Contents

<b>Abstract</b>	<b>2</b>
<b>1.Introduction</b>	<b>3</b>
<b>2.Expectations</b>	<b>4</b>
<b>3.Organisation and Structure</b>	<b>6</b>
3.1.    Main sections	.
3.1.1.    Introduction	.
3.1.2.    Analysis	.
3.1.3.    Design	.
3.1.4.    Implementation	.
3.1.5.    Evaluation	.
3.1.6.    Conclusion	.
3.1.7.    Support elements	.
3.1.8.    Back matter	.
<b>4.Figures, Tables and Code</b>	.
4.1.    code	.
4.2.    figures	.
4.3.    tables	.
4.4.    some useful resources	.
<b>5.Appendix</b>	.
5.1.    Declaration	.
5.2.    References	.

# Abstract

The project involves creating a Python API/ library to be used by students or academic professionals with the aim of visualising sets, using InteractiVenn's approach to Set Visualisation, to allow a greater understanding of given sets, and their unions and intersections. The Python API developed should allow users to input up to six sets of data to be visualised as an interactive Venn Diagram with functionality that:

- 1) Allow users to visualise the union and intersections of the given complex input sets.
- 2) Allow users to hover over set unions and present the elements of a given set.

The developed Python API should take into account that the users may have little or no technical skills, this should direct the design towards being as easy as possible to use for the target audience. The documentation of the Python API must have clear and concise information on its functionality, along with concrete examples displaying to the user the operations that can be performed.

The project includes running and evaluating the developed Python API by itself as well as against an established API, *UpSet – Visualising Intersecting Sets*, to highlight the advantages and disadvantages of each API given a number of examples.

- **crisply summarises the project in a single paragraph**
- **used to assess if a document is worthy of reading**
- ***appears on its own page***
- ***positioned verticalled on the page beginning roughly  $\frac{1}{3}$  down from the top***
- ***labelled with the word "Abstract" (bold face)***
- ***between 50-200 words***

# 1 Introduction

**“The Introduction should provide a brief non-technical overview of the project. It should indicate clearly what the goal was and what was achieved and also how, in broad terms, the project was conducted. It may include a thumbnail sketch of the main concepts or technologies used, but without any of the technical details.**

**The idea is to present a summary of the work that is accessible to as wide an audience as possible. It should try to enthuse the reader to read the rest of the report.”**

---

**“set the scene - paint a picture - basically why and how?.**

**Why?**

- Importance, relevance and interest to (the most important & relevant of):-
  - humanity, science, computer science, department, yourself.

**How?**

- this part of the introduction should be brief (< 1 page) - only give only a general outline of approach to give direction to the thesis and whet the reader's appetite - don't write the entire thesis in the intro - else it's finished”
-

## 2 Expectations

The project should commence with research into which technologies should be used and what API's are best for outputting interactive models to the user through Python. The design of the Python API should have consideration for users coming from any Operating System, with minimal importing of dependencies needed, allowing for users of any technical skill level to operate.

Technologies that need to be decided on:

- Which technology for displaying interactive outputs through Python.
- How the documentation of the Python API will be presented to the user.
- Which file types the user will be inputting and relevant error handling measures.

Once the technologies have been chosen, the implementation may begin. This will involve the development of the Python API using the chosen technologies from the initial definition phase of the project. The implementation involves:

- 1) Creating a directory to store the Python library.
- 2) Creating a virtual environment for the directory.
- 3) Creating the content of the Python library.
- 4) Building the Python library.

Development of the documentation, to allow the users to understand the functionality of the API, may commence. Once the implementation and documentation have been completed to a satisfactory level, the final stage of the project may begin.

The final stage of the project will involve testing the Python API extensively, to discard unforeseen errors or missing functionality. The Python API will be evaluated by itself to assure it satisfies the requirements of the project. The Python API will also be evaluated against the existing *UpSet Visualisation* API to compare the two projects. The documentation will be finalised and completed. The project closure will involve public and peer testing of the developed Python API by the target audience.

**“Most projects involve the development of some piece of software, though there are other possibilities as listed below. Expectations and the assessment yardsticks applied vary between different types of projects, reflecting the underlying diversity of the projects themselves. Clearly what may be appropriate in the case of a software development project may be pointless and irrelevant in the case of an algorithm design project.**

**Software development project** The most common category of project involves the design and implementation of some substantial piece of software or perhaps a hardware-software system.

**Research project (empirical)** Other projects typically involve the experimental evaluation of some idea (algorithm, heuristic etc.). This may involve the bench-marking of some software package or a rigorous comparison of a number of competing algorithms or protocols.

**Research project (investigative)** Projects in this category might involve the development of a new algorithm for some problem or an enhancement of an existing algorithm.”

### 3 Organisation and Structure

**“This section specifies the standard organisation of a project report. It focuses mainly on software development projects but also touches the other types. Subsection 4.1 discusses the heart of the report and the sections into which should be subdivided. Subsection 4.2 specifies the format and sequencing of the other elements (title page, table of contents and so on).”**

#### 1. Main Sections

##### 1.1. Introduction

**“Set the scene - paint a picture - basically why and how?.**

**Why?:**

- **Importance, relevance and interest to (the most important & relevant of):**
  - **humanity, science, computer science, department, yourself.**

**How?:**

- **this part of the introduction should be brief (< 1 page) - only give only a a general outline of approach to give direction to the thesis and whet the reader’s appetite - don’t write the entire thesis in the intro - else it’s finished”**

##### 1.2. Analysis

***“How comprehensive is the student’s literature review? Is the review analytic rather than descriptive? Are the strengths and weaknesses of existing work identified?”***

##### 1.3. Design

**“Is the research methodology the right one? Are the experiments well designed? Are the experimental materials (e.g. prototype systems, etc.) well designed? Can the hypotheses be verified?”**

##### 1.4. Implementation

**“Were the experimental materials properly constructed? Were the experiments properly conducted?”**

##### 1.5. Evaluation

**\*\*\* have a documentation for API use \*\*\***

**My API vs UpSet:**

- **advantages**
- **disadvantages**
- **suitable utility for each**

**“**

- **Collect and run the API's of the two tools on collected data.**
- **Evaluate the new API by itself and in comparison with UpSet.**

**“**

**“Are the results well reported (e.g. using tables, graphs, etc.)? Are the results analysed deeply and properly explained? Is there both quantitative and qualitative analysis of the results?”**

- 1.6. Conclusion
- 1.7. Support elements
- 1.8. Back matter

## 4 Figures and Code

1. Code
2. Figures
3. Tables
4. Some useful resources



## 5 Acknowledgements

This report and research would not have been possible without the guidance and support of my supervisor Dr Rosane Minghim. Dr.Minghim's supervision has been critical from the beginning of the project to the conclusive report. Her knowledge and management showcased in bi-weekly meetings, manifested in meeting reports as well as a continuous logbook of the project's progress, helped keep the project and myself on track to achieve the goals set for the project.

I would like to thank Frank Boehme, the administrator of the Final Year Project process for the BSc Computer Science degree program. Mr.Boehme's lecture at the beginning of the academic calendar assisted in many aspects of the report, such as report structure, key dates and preferred language, along with the information necessary for submission in a timely manner.

**“You must acknowledge the sources the any published material that you make use of in your report. Failure to do so can have very adverse con-sequences [10]. The details of each source (author name, title of the work in question, publisher and so on) should be recorded in the References section at the back of the report.”**

### **1. Some useful resources**

**“For sound advice on (non-technical) writing in general, Strunk and White's slim classic Elements of Style [12], a stalwart on US campuses for generations, is well worth reading.**

**For a reliable source on spelling and related matters, you should refer to The Oxford English Dictionary [1]. There are copies in the library and the on-line version is quite useful. For guidance on punctuation and grammar, the authoritative Fowler's Dictionary of Modern English Usage [9] is a good source. There is also a (limited) on-line version.**

**The IEEE Computer Society has a useful style guide [11]. It is aimed at authors intending to publish in the organization's technical journals and offers guidance on technical (specifically computer-related) writing.”**

## 6 Appendix

**“All students are bound by UCC’s Plagiarism Policy [2] and are expected to familiarize themselves with the provisions (and penalties) of that policy. The principles set out below interpret this policy in the context of final year projects in Computer Science.**

**The overarching principle is that the presentation of the project, encompassing both the project report and the underlying body of technical work, should be accurate and honest in describing what has been accomplished and in specifying the precise individual contribution of the examinee.”**

## 7 Declaration

This is to certify that the work I am submitting is my own and has been done by me solely and not in consultation with anyone else. Neither I nor anyone else have submitted this work for assessment, either at University College Cork or elsewhere. I have read and understood University College Cork's exam regulations, plagiarism policy and Code of Honour. I understand that breaches of this declaration are serious issues and can incur penalties.

---

Signature of Student

- ***signed declaration that the work submitted is the fruits of your own individual effort.***
- ***report not accepted without***

# References

**“Include an entry that includes the publication details of the work in question in the References section:**

**[1] Jeremy Butterfield, ed. Fowler’s Dictionary of Modern English Usage. 4th. Oxford University Press, 2015. isbn: 978-0199661350.**

**[2] University College Cork. UCC Plagiarism Policy. url: [www.ucc.ie/en/exams/procedures-regulations/](http://www.ucc.ie/en/exams/procedures-regulations/).**

**[3] Thomas H. Cormen et al. Introduction to Algorithms. 2nd. McGraw-Hill Higher Education, 2001. isbn: 0070131511.**

**[4] Charles Darwin. On the origin of species. (Available online: [www.gutenberg.org/ebooks/22764](http://www.gutenberg.org/ebooks/22764)). London: John Murray, 1859.**

**[5] Marc van Dongen. L A T E X and Friends. url: [csweb.ucc.ie/~dongen/LAF/LAF.html](http://csweb.ucc.ie/~dongen/LAF/LAF.html).**