anicode Documentation

Release 0.0.2

Stephen Bunn

Oct 04, 2019

Contents

A quick unicode search tool

Documentation

1.1 Getting Started

This tools is something I created for myself for quickly searching for some unicode characters without having to open my browser everytime.

1.1.1 Installation

Anicode is available on PyPi. So you can easily install the anicode package with the following command:

```
pip install --user anicode
```

This will install the Anicode package as well as give you access to the embeded command line utility!

Note: If you do not have access to the anicode command line utility after using the above command, make sure that the `~/.local/bin/` directory is included in your `$PATH` environment varaible.

Important: Anicode requires Python 3.5+! If you haven't yet started using Python 3, you should definitely start since **many** projects are fully dropping support for Python 2.7.

1.1.2 Command Line

Using Anicode from the command line is really simple. After installing Anicode from pip, the `anicode` tool should be accessible from the shell.

You should be able to search for unicode characters using the following command:

```
anicode search "query"
```

After searching, the tool will prompt you to select the character you want.

Note: This tool only searches for proper unicode character names. Fuzzy unicode character matches are not yet supported.

1.2 Changelog

All notable changes to anicode will be documented in this file.

The format is based on Keep a Changelog and this project adheres to Semantic Versioning.

1.2.1 *unreleased*

1.2.2 0.0.2 (*2019-10-04*)

- fixed lxml import for setup.py

1.2.3 0.0.1 (*2018-01-15*)

- fixed usage documentation

1.2.4 0.0.0 (*2018-01-15*)

this is the first pre-alpha release, so the only other prior change history is the git commit log

- added support for basic unicode searches
- **added support for the following sources**
    - unicode_table
- added command line interface

CHAPTER 2

Reference

2.1 Contributing

When contributing to this repository, please first discuss the change you wish to make via an issue to the owners of this repository before submitting a pull request.

Important: **We have an enforced style guide and a code of conduct.** Please follow them in all your interactions with this project.

2.1.1 Style Guide

- We stictly follow [PEP8](#) and utilize [Sphinx](#) docstrings on **all** classes and functions.
- **We employee [flake8](#) as our linter with exceptions to the following rules:**
    - D203
    - F401
    - E123
- Linting and test environments are configured via `tox.ini`.
- An `.editorconfig` file is included in this repository which dictates whitespace, indentation, and file encoding rules.
- Although `requirements.txt` and `requirements_dev.txt` do exist, [Pipenv](#) is utilized as the primary virtual environment and package manager for this project.
- We strictly utilize [Semantic Versioning](#) as our version specification.

2.1.2 Issues

Issues should follow the included `ISSUE_TEMPLATE` found in `.github/ISSUE_TEMPLATE.md`.

- **Issues should contain the following sections:**

    - Expected Behavior

    - Current Behavior

    - Possible Solution

    - Steps to Reproduce (for bugs)

    - Context

    - Your Environment

These sections help the developers greatly by providing a large understanding of the context of the bug or requested feature without having to launch a full fleged discussion inside of the issue.

2.1.3 Pull Requests

Pull requests should follow the included `PULL_REQUEST_TEMPLATE` found in `.github/PULL_REQUEST_TEMPLATE.md`.

- Pull requests should always be from a **topic/feature/bugfix** (left side) branch. *Pull requests from master branches will not be merged.*

- Pull requests should not fail our requested style guidelines or linting checks.

2.1.4 Code of Conduct

Our code of conduct is taken directly from the Contributor Covenant since it directly hits all of the points we find necessary to address.

Our Pledge

In the interest of fostering an open and welcoming environment, we as contributors and maintainers pledge to making participation in our project and our community a harassment-free experience for everyone, regardless of age, body size, disability, ethnicity, gender identity and expression, level of experience, education, socio-economic status, nationality, personal appearance, race, religion, or sexual identity and orientation.

Our Standards

Examples of behavior that contributes to creating a positive environment include:

- Using welcoming and inclusive language

- Being respectful of differing viewpoints and experiences

- Gracefully accepting constructive criticism

- Focusing on what is best for the community

- Showing empathy towards other community members

Examples of unacceptable behavior by participants include:

- The use of sexualized language or imagery and unwelcome sexual attention or advances

- Trolling, insulting/derogatory comments, and personal or political attacks

- Public or private harassment

- Publishing others' private information, such as a physical or electronic address, without explicit permission

- Other conduct which could reasonably be considered inappropriate in a professional setting

Our Responsibilities

Project maintainers are responsible for clarifying the standards of acceptable behavior and are expected to take appropriate and fair corrective action in response to any instances of unacceptable behavior.

Project maintainers have the right and responsibility to remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that are not aligned to this Code of Conduct, or to ban temporarily or permanently any contributor for other behaviors that they deem inappropriate, threatening, offensive, or harmful.

Scope

This Code of Conduct applies both within project spaces and in public spaces when an individual is representing the project or its community. Examples of representing a project or community include using an official project e-mail address, posting via an official social media account, or acting as an appointed representative at an online or offline event. Representation of a project may be further defined and clarified by project maintainers.

Enforcement

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported by contacting the project team at stephen@bunn.io. All complaints will be reviewed and investigated and will result in a response that is deemed necessary and appropriate to the circumstances. The project team is obligated to maintain confidentiality with regard to the reporter of an incident. Further details of specific enforcement policies may be posted separately.

Project maintainers who do not follow or enforce the Code of Conduct in good faith may face temporary or permanent repercussions as determined by other members of the project's leadership.

Attribution

This Code of Conduct is adapted from the Contributor Covenant, version 1.4, available at https://www.contributor-covenant.org/version/1/4/code-of-conduct.html

2.2 Command Line

Here is a autodoc built refernce of the current state of anicode.

2.3 Anicode Package

This is the base Anicode package.

2.3.1 anicode

2.3.2 anicode.result

The default anicode result object.

2.3.3 anicode.sources

Below are a list of currently included unicode sources utilized by `Anicode`.

unicode_table

2.4 Indices

- genindex
- modindex
- search