Repository Analysis Report: Textualize_rich

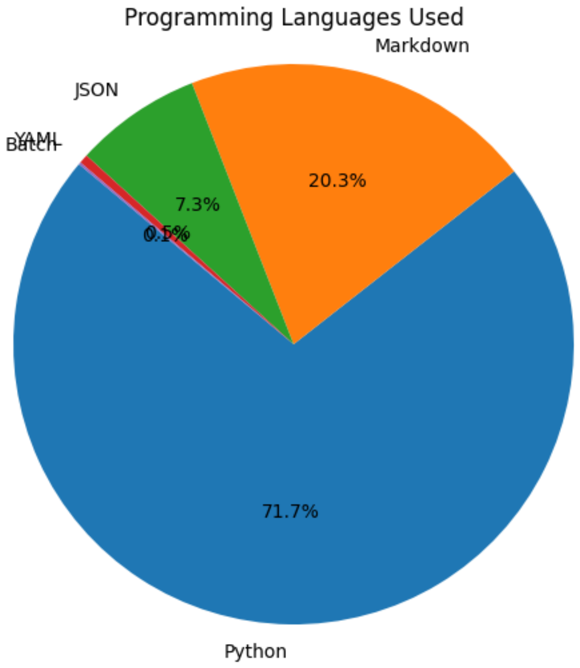
Programmer Perspective

Generated on March 22, 2025

Repository Information

Name:	Textualize_rich
Owner:	Textualize
URL:	https://github.com/Textualize/rich
Languages:	YAML, Markdown, Batch, Python, JSON
Commit Count:	4152
Contributors:	293

Programming Languages Distribution:



Introduction

This report provides a technical analysis of the repository from a programmer's perspective. It focuses on code structure, architecture, technologies used, and development practices.

The following pages contain answers to key questions relevant to this perspective, based on automated analysis of the repository content.

Question 1: What programming languages are used in this project?

Answer to: What programming languages are used in this project? Based on the repository information: Source 1: README.sv.md Name: Projekt som använder sig av Rich Content excerpt: # Projekt som använder sig av Rich Här är ett par projekt som använder Rich: - [BrancoLab/BrainRender](https://github.com/BrancoLab/BrainRender) ett python packet för visualisering av tredimensionell neuro-anatomiska data - [Ciphey/Ciphey](https://github.com/Ciphey/Ciphey) Automatiserat dekryp... Source 2: README.cn.md Name: ■■ Rich ■■■ Content excerpt: ## ■■ Rich ■■■ ■■■■■■■ Rich ■■■: - [BrancoLab/BrainRender](https://github.com/BrancoLab/BrainRender) ■■■■■■■■■■■■■■■■ python ■ - [Ciphey/Ciphey](https://github.com/Ciphey/Ciphey) ■■■■■■ - [emeryberger/scalene](https://github.com/emeryberger/scalene) ■■■■■■■■■■ Python CPU ■■■■■■ - [hedythedev... Source 3: README.es.md Name: Proyecto usando Rich Content excerpt: # Proyecto usando Rich Aquí hay algunos proyectos que usan Rich: - [BrancoLab/BrainRender](https://github.com/BrancoLab/BrainRender) un paquete de Python para la visualización de datos neuroanatómicos tridimensionales - [Ciphey/Ciphey](https://github.com/Ciphey/Ciphey) Herramienta de descifrad... Note: This is a simplified answer without using an LLM. For more comprehensive answers, please set up OpenAI API access.

Based on information from:

1. *README.sv.md (Projekt som använder sig av Rich)*
2. *README.cn.md (■■ Rich ■■■)*
3. *README.es.md (Proyecto usando Rich)*

Question 2: What is the project's architecture/structure?

Answer to: What is the project's architecture/structure? Based on the repository information: Source 1: CODE_OF_CONDUCT.md Name: Scope Content excerpt: ## 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 a... Source 2: .github/FUNDING.yml Content excerpt: # These are supported funding model platforms Source 3: CODE_OF_CONDUCT.md Name: Our Responsibilities Content excerpt: ## 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, ed... Note: This is a simplified answer without using an LLM. For more comprehensive answers, please set up OpenAI API access.

Based on information from:

1. CODE_OF_CONDUCT.md (Scope)

2. .github/FUNDING.yml

3. CODE_OF_CONDUCT.md (Our Responsibilities)

Question 3: What are the main components/modules of the project?

Answer to: What are the main components/modules of the project? Based on the repository information: Source 1: CODE_OF_CONDUCT.md Name: Scope Content excerpt: ## 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 a... Source 2: README.md Name: Textual Content excerpt: # Textual See also Rich's sister project, [Textual](https://github.com/Textualize/textual), which you can use to build sophisticated User Interfaces in the terminal. ![Textual screenshot](https://raw.githubusercontent.com/Textualize/textual/main/imgs/textual.png) Source 3: .gitignore Content excerpt: *.ipynb .pytype .DS_Store .vscode .idea/ mypy_report docs/build docs/source/_build tools/*.txt playground/ # Byte-compiled / optimized / DLL files __pycache__/ *.py[cod] *$py.class # C extensions *.so # Distribution / packaging .Python build/ develop-eggs/ dist/ downloads/ eggs/ .eggs/ lib/ lib64... Note: This is a simplified answer without using an LLM. For more comprehensive answers, please set up OpenAI API access.

Based on information from:

1. CODE_OF_CONDUCT.md (Scope)

2. README.md (Textual)

3. .gitignore

Question 4: What testing framework(s) are used?

Answer to: What testing framework(s) are used? Based on the repository information: Source 1: tests/test_windows_renderer.py Name: legacy_term_mock Content excerpt: def legacy_term_mock(): return create_autospec(LegacyWindowsTerm) Source 2: benchmarks/README.md Name: Running Benchmarks Content excerpt: ## Running Benchmarks We strongly recommend running `asv run --help` for a full list of options, but here are some common actions: * You can run the benchmarks against the `master` branch with `asv run`. * To test the most recent commit on your branch `asv run HEAD^!`. * To generate a static websi... Source 3: tests/test_control.py Name: test_control Content excerpt: def test_control(): control = Control(ControlType.BELL) assert str(control) == "\x07" Note: This is a simplified answer without using an LLM. For more comprehensive answers, please set up OpenAI API access.

Based on information from:

1. tests/test_windows_renderer.py (legacy_term_mock)

2. benchmarks/README.md (Running Benchmarks)

3. tests/test_control.py (test_control)

Question 5: What dependencies does this project have?

Answer to: What dependencies does this project have? Based on the repository information: Source 1: setup.py Content excerpt: #!/usr/bin/env python # This is a shim to hopefully allow Github to detect the package, build is done with poetry import setuptools if __name__ == "__main__": setuptools.setup(name="rich") Source 2: logs/_processing.log Content excerpt: 2025-03-22 08:57:06,950 - content_processor. - INFO - Initialized content processor for repos/Textualize_rich/ 2025-03-22 08:57:06,950 - content_processor. - INFO - Starting repository processing: repos/Textualize_rich/ 2025-03-22 08:57:06,950 - content_processor. - INFO - Processed pyproject.toml (... Source 3: .github/dependabot.yml Content excerpt: # To get started with Dependabot version updates, you'll need to specify which # package ecosystems to update and where the package manifests are located. # Please see the documentation for all configuration options: # https://help.github.com/github/administering-a-repository/configuration-options-f... Note: This is a simplified answer without using an LLM. For more comprehensive answers, please set up OpenAI API access.

Based on information from:

1. setup.py

2. logs/_processing.log

3. .github/dependabot.yml

Question 6: What is the code quality like (comments, documentation, etc.)?

Answer to: What is the code quality like (comments, documentation, etc.)? Based on the repository information: Source 1: CONTRIBUTING.md Name: Developing Content excerpt: ## Developing At this point, you're ready to start developing. Some things to consider while developing Rich code include: * Ensure new code is documented in docstrings * Avoid abbreviations in variable or class names * Aim for consistency in coding style and API design Before each [commit](https... Source 2: benchmarks/benchmarks.py Name: PrettySuite Content excerpt: class PrettySuite: def setup(self): self.console = Console(file=StringIO(), color_system="truecolor", legacy_windows=False, width=100) def time_pretty(self): pretty = Pretty(snippets.PYTHON_DICT) self.console.print(pretty) def time_pretty_i... Source 3: benchmarks/benchmarks.py Name: SyntaxWrappingSuite Content excerpt: class SyntaxWrappingSuite: def setup(self): self.console = Console(file=StringIO(), color_system="truecolor", legacy_windows=False) self.syntax = Syntax(code=snippets.PYTHON_SNIPPET, lexer="python", word_wrap=True) def time_text_th... Note: This is a simplified answer without using an LLM. For more comprehensive answers, please set up OpenAI API access.

Based on information from:

1. CONTRIBUTING.md (Developing)

2. benchmarks/benchmarks.py (PrettySuite)

3. benchmarks/benchmarks.py (SyntaxWrappingSuite)

Question 7: Are there any known bugs or issues?

Answer to: Are there any known bugs or issues? Based on the repository information: Source 1: .github/ISSUE_TEMPLATE/bug_report.md Content excerpt: --- name: Bug report about: Create a report to help us improve title: "[BUG]" labels: Needs triage assignees: "" --- - [ ] I've checked [docs](https://rich.readthedocs.io/en/latest/introduction.html) and [closed issues](https://github.com/Textualize/rich/issues?q=is%3Aissue+is%3Aclosed) for possibl... Source 2: CHANGELOG.md Name: Fixed Content excerpt: ### Fixed - Fix Rich clobbering cursor style on Windows https://github.com/Textualize/rich/pull/2339 - Fix text wrapping edge case https://github.com/Textualize/rich/pull/2296 - Allow exceptions that are raised while a Live is rendered to be displayed and/or processed https://github.com/Textualize/... Source 3: CHANGELOG.md Name: Fixed Content excerpt: ### Fixed - Fixed error message for tracebacks with broken `__str__` https://github.com/textualize/rich/issues/980 - Fixed markup edge case https://github.com/textualize/rich/issues/987 Note: This is a simplified answer without using an LLM. For more comprehensive answers, please set up OpenAI API access.

Based on information from:

1. .github/ISSUE_TEMPLATE/bug_report.md

2. CHANGELOG.md (Fixed)

3. CHANGELOG.md (Fixed)

Question 8: What is the build/deployment process?

Answer to: What is the build/deployment process? Based on the repository information: Source 1: CONTRIBUTING.md Name: Pre-Commit Content excerpt: ### Pre-Commit We strongly recommend you [install the pre-commit hooks](https://pre-commit.com/#installation) included in the repository. These automatically run some of the checks described earlier each time you run `git commit`, and over time can reduce development overhead quite considerably. Source 2: .gitignore Content excerpt: *.ipynb .pytype .DS_Store .vscode .idea/ mypy_report docs/build docs/source/_build tools/*.txt playground/ # Byte-compiled / optimized / DLL files __pycache__/ *.py[cod] *$py.class # C extensions *.so # Distribution / packaging .Python build/ develop-eggs/ dist/ downloads/ eggs/ .eggs/ lib/ lib64... Source 3: benchmarks/README.md Name: Running Benchmarks Content excerpt: ## Running Benchmarks We strongly recommend running `asv run --help` for a full list of options, but here are some common actions: * You can run the benchmarks against the `master` branch with `asv run`. * To test the most recent commit on your branch `asv run HEAD^!`. * To generate a static websi... Note: This is a simplified answer without using an LLM. For more comprehensive answers, please set up OpenAI API access.

Based on information from:

1. CONTRIBUTING.md (Pre-Commit)

2. .gitignore

3. benchmarks/README.md (Running Benchmarks)

Question 9: How is version control used in the project?

Answer to: How is version control used in the project? Based on the repository information: Source 1: CHANGELOG.md Name: Changelog Content excerpt: # Changelog All notable changes to this project will be documented in this file. The format is based on [Keep a Changelog](https://keepachangelog.com/en/1.0.0/), and this project adheres to [Semantic Versioning](https://semver.org/spec/v2.0.0.html). Source 2: CONTRIBUTING.md Name: Pre-Commit Content excerpt: ### Pre-Commit We strongly recommend you [install the pre-commit hooks](https://pre-commit.com/#installation) included in the repository. These automatically run some of the checks described earlier each time you run `git commit`, and over time can reduce development overhead quite considerably. Source 3: CODE_OF_CONDUCT.md Name: Our Responsibilities Content excerpt: ## 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, ed... Note: This is a simplified answer without using an LLM. For more comprehensive answers, please set up OpenAI API access.

Based on information from:

1. CHANGELOG.md (Changelog)

2. CONTRIBUTING.md (Pre-Commit)

3. CODE_OF_CONDUCT.md (Our Responsibilities)

Question 10: What coding standards or conventions are followed?

Answer to: What coding standards or conventions are followed? Based on the repository information: Source 1: CODE_OF_CONDUCT.md Name: Scope Content excerpt: ## 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 a... Source 2: CODE_OF_CONDUCT.md Name: Our Standards Content excerpt: ## 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 * Showin... Source 3: CODE_OF_CONDUCT.md Name: Our Responsibilities Content excerpt: ## 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, ed... Note: This is a simplified answer without using an LLM. For more comprehensive answers, please set up OpenAI API access.

Based on information from:

1. CODE_OF_CONDUCT.md (Scope)

2. CODE_OF_CONDUCT.md (Our Standards)

3. CODE_OF_CONDUCT.md (Our Responsibilities)

Conclusion

This report was generated automatically by analyzing the repository content. The analysis is based on the code, documentation, and configuration files present in the repository. For more detailed information, please refer to the repository itself or contact the development team.