# Trending on Hacker News and GitHub with 120 Lines of Python

From Writing to Publishing makesite.py

Sunaina Pai

PyCon UK 2018, Cardiff City Hall, Cardiff, UK

17 Sep 2018

## who am i

**Sunaina Pai**

Software Developer from Bangalore, India.

Working with Java and big data technologies during the day.

Exploring the beauty of Python and Lisp in the evening.

https://sunainapai.in/

https://github.com/sunainapai

https://twitter.com/sunainapai

## Problem

**Background**

My static website/blog generator was written in shell script.

**Problem**

Write a new static site generator in a sane language!

**Project**

Write a few Python functions to render my static website/blog.

## Scope

- No magic!
- Minimal set of necessary features.
- Generate my website/blog with this project.
- Support multiple blogs in a website.
- Generate RSS feed for each blog.
- Share the project on Hacker News with a “Show HN” post.

## Directory Structure

```
.
|-- makesite.py
|-- content
|   |-- _index.html
|   |-- about.html
|   |-- contact.html
|   |-- blog
|   |   |-- 2018-01-01-proin-quam.md
|   |   `-- 2018-01-03-sed-finibus.md
|   `-- news
|       |-- 2018-01-02-vivamus-purus.html
|       `-- 2018-01-04-mauris-tempor.html
|-- layout
|   |-- feed.xml
|   |-- item.html
|   |-- item.xml
|   |-- list.html
|   |-- page.html
|   `-- post.html
|-- static
|   `-- css
|       `-- style.css
`-- _site
```

## Layout Template Example

```
<!DOCTYPE html>
<html>
  <head>
    <title>{{ title }} - {{ subtitle }}</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width">
    <link rel="stylesheet" type="text/css" href="/css/style.css">
  </head>
  <body>
    <main>
      {{ content }}
    </main>
    <footer>
      <div>&copy; {{ current_year }} Lorem Ipsum</div>
    </footer>
  </body>
</html>
```

## Content Example

```
<!-- title: Proin Quam -->
<!-- author: Alice -->
Proin quam urna, pulvinar id ipsum ac, mattis consectetur ante. Praesent
non justo lectus. Duis egestas arcu libero, quis laoreet dolor volutpat
ut. Donec facilisis orci sit amet sem blandit elementum.

Vestibulum suscipit consectetur diam, ac posuere metus condimentum in.
Integer vehicula vitae enim id gravida.

Vestibulum ut eros vitae risus porttitor porta in eget felis. Nulla
lorem erat, mattis eget lacus eget, interdum aliquet lectus. Fusce non
felis diam.
```

## Code Walkthrough: Source Code URL

git.io/makesite.py

## Code Walkthrough: Utilities

```python
import os
import shutil
import re
import glob
import sys
import json
import datetime

def fread(filename):
    """Read file and close the file."""
    with open(filename, 'r') as f:
        return f.read()


def fwrite(filename, text):
    """Write content to file and close the file."""
    basedir = os.path.dirname(filename)
    if not os.path.isdir(basedir):
        os.makedirs(basedir)

    with open(filename, 'w') as f:
        f.write(text)
```

## Code Walkthrough: Utilities

```
def log(msg, *args):
    """Log message with specified arguments."""
    sys.stderr.write(msg.format(*args) + '\n')


def truncate(text, words=25):
    """Remove tags and truncate text to the specified number of words."""
    return ' '.join(re.sub('(?s)<.*?>', ' ', text).split()[:words])


def rfc_2822_format(date_str):
    """Convert yyyy-mm-dd date string to RFC 2822 format date string."""
    d = datetime.datetime.strptime(date_str, '%Y-%m-%d')
    return d.strftime('%a, %d %b %Y %H:%M:%S +0000')
```

## Code Walkthrough: Header Parsing

```
def read_headers(text):
    """Parse headers in text and yield (key, value, end-index) tuples."""
    for match in re.finditer('\s*<!--\s*(.+?)\s*:\s*(.+?)\s*-->\s*|.+', text):
        if not match.group(1):
            break
        yield match.group(1), match.group(2), match.end()
```

### A note on headers

Headers are key-value pairs within HTML comments at the top of a content file.

```
<!-- title: Proin Quam -->
<!-- author: Alice -->
<!-- key1: value1 -->
<!-- key2: value2 -->
Content
```

## Code Walkthrough: Content Parsing

```python
def read_content(filename):
    """Read content and metadata from file into a dictionary."""
    # Read file content.
    text = fread(filename)

    # Read metadata and save it in a dictionary.
    date_slug = os.path.basename(filename).split('.')[0]
    match = re.search('^(?:(\d\d\d\d-\d\d-\d\d)-)?(.+)$', date_slug)
    content = {
        'date': match.group(1) or '1970-01-01',
        'slug': match.group(2),
    }

    # Read headers.
    end = 0
    for key, val, end in read_headers(text):
        content[key] = val

    # Separate content from headers.
    text = text[end:]

    # ... read_content() definition continued on next slide ...
```

## Code Walkthrough: Content Parsing

```python
# ... read_content() definition continued from previous slide ...

# Convert Markdown content to HTML.
if filename.endswith(('.md', '.mkd', '.mkdn', '.mdown', '.markdown')):
    try:
        if _test == 'ImportError':
            raise ImportError('Error forced by test')
        import CommonMark
        text = CommonMark.commonmark(text)
    except ImportError as e:
        log('WARNING: Cannot render Markdown in {}: {}', filename, str(e))

# Update the dictionary with content and RFC 2822 date.
content.update({
    'content': text,
    'rfc_2822_date': rfc_2822_format(content['date'])
})

return content
```

## Code Walkthrough: Template Rendering

```
def render(template, **params):
    """Replace placeholders in template with values from params."""
    return re.sub(r'{{\s*([^}\s]+)\s*}}',
                  lambda match: str(params.get(match.group(1), match.group(0))),
                  template)
```

### A note on rendering

Key-values represented by `**params` populate placeholders with matching names in templates. For example, if `template` is

```
<head><title>{{ title }} - {{ subtitle }}</title></head>
<body>{{ content }}</body>
```

then

```
render(template, title='About', subtitle='Lorem Ipsum', content='Foo')
```

returns

```
<head><title>About - Lorem Ipsum</title></head>
<body><p>Foo</p></body>
```

## Code Walkthrough: Building Blocks

```python
def make_pages(src, dst, layout, **params):
    """Generate pages from page content."""
    items = []

    for src_path in glob.glob(src):
        content = read_content(src_path)
        page_params = dict(params, **content)

        # Populate placeholders in content if content-rendering is enabled.
        if page_params.get('render') == 'yes':
            rendered_content = render(page_params['content'], **page_params)
            page_params['content'] = rendered_content
            content['content'] = rendered_content

        items.append(content)

        dst_path = render(dst, **page_params)
        output = render(layout, **page_params)

        log('Rendering {} => {} ...', src_path, dst_path)
        fwrite(dst_path, output)

    return sorted(items, key=lambda x: x['date'], reverse=True)
```

## Code Walkthrough: Building Blocks

```python
def make_list(posts, dst, list_layout, item_layout, **params):
    """Generate list page for a blog."""
    items = []
    for post in posts:
        item_params = dict(params, **post)
        item_params['summary'] = truncate(post['content'])
        item = render(item_layout, **item_params)
        items.append(item)

    params['content'] = ''.join(items)
    dst_path = render(dst, **params)
    output = render(list_layout, **params)

    log('Rendering list => {} ...', dst_path)
    fwrite(dst_path, output)
```

## Code Walkthrough: Tying It All Together

```python
def main():
    # Create a new _site directory from scratch.
    if os.path.isdir('_site'):
        shutil.rmtree('_site')
    shutil.copytree('static', '_site')

    # Default parameters.
    params = {
        'base_path': '',
        'subtitle': 'Lorem Ipsum',
        'author': 'Admin',
        'site_url': 'http://localhost:8000',
        'current_year': datetime.datetime.now().year
    }

    # If params.json exists, load it.
    if os.path.isfile('params.json'):
        params.update(json.loads(fread('params.json')))

    # ... main() definition continued on next slide ...
```

## Code Walkthrough: Tying It All Together

```
# ... main() definition continued from previous slide ...

# Load layouts.
page_layout = fread('layout/page.html')
post_layout = fread('layout/post.html')
list_layout = fread('layout/list.html')
item_layout = fread('layout/item.html')
feed_xml = fread('layout/feed.xml')
item_xml = fread('layout/item.xml')

# Combine layouts to form final layouts.
post_layout = render(page_layout, content=post_layout)
list_layout = render(page_layout, content=list_layout)

# Create site pages.
make_pages('content/_index.html', '_site/index.html',
           page_layout, **params)
make_pages('content/[!_]*.html', '_site/{{ slug }}/index.html',
           page_layout, **params)

# ... main() definition continued on next slide ...
```

## Code Walkthrough: Tying It All Together

```python
# ... main() definition continued from previous slide ...

# Create blogs.
blog_posts = make_pages('content/blog/*.md',
                        '_site/blog/{{ slug }}/index.html',
                        post_layout, blog='blog', **params)
news_posts = make_pages('content/news/*.html',
                        '_site/news/{{ slug }}/index.html',
                        post_layout, blog='news', **params)

# Create blog list pages.
make_list(blog_posts, '_site/blog/index.html',
          list_layout, item_layout, blog='blog', title='Blog', **params)
make_list(news_posts, '_site/news/index.html',
          list_layout, item_layout, blog='news', title='News', **params)

# Create RSS feeds.
make_list(blog_posts, '_site/blog/rss.xml',
          feed_xml, item_xml, blog='blog', title='Blog', **params)
make_list(news_posts, '_site/news/rss.xml',
          feed_xml, item_xml, blog='news', title='News', **params)
```

## Code Walkthrough: Tying It All Together

```
# Test parameter to be set temporarily by unit tests.
_test = None

if __name__ == '__main__':
    main()
```

**Generating Static Website/Blog**

Running `./makesite.py` generates the static website/blog and writes it to `_site` directory.

**Local Testing**

The project comes with a convenience make serve target to generate the static website/blog and serve it locally with `SimpleHTTPServer` or `http.server` (Python 3) on port 8000.

## Project URLs

Source code: https://github.com/sunainapai/makesite

Short URL: https://git.io/makesite

Demo website/blog: https://tmug.github.io/makesite-demo/

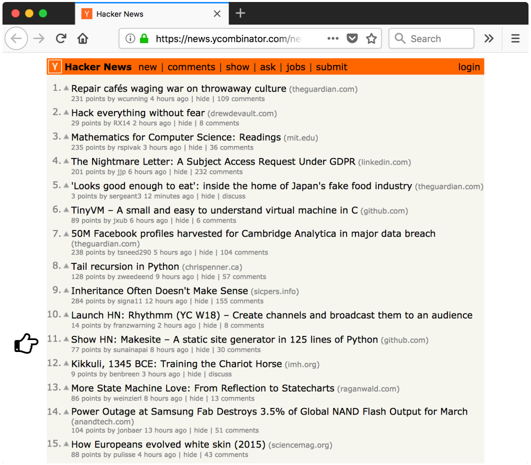
Short URL: https://git.io/makesite-demo

### Creating Short URLs

```
curl -i https://git.io \
     -F "url=https://github.com/sunainapai/makesite" \
     -F "code=makesite"

curl -i https://git.io \
     -F "url=https://tmug.github.io/makesite-demo/" \
     -F "code=makesite-demo"
```

## On Hacker News



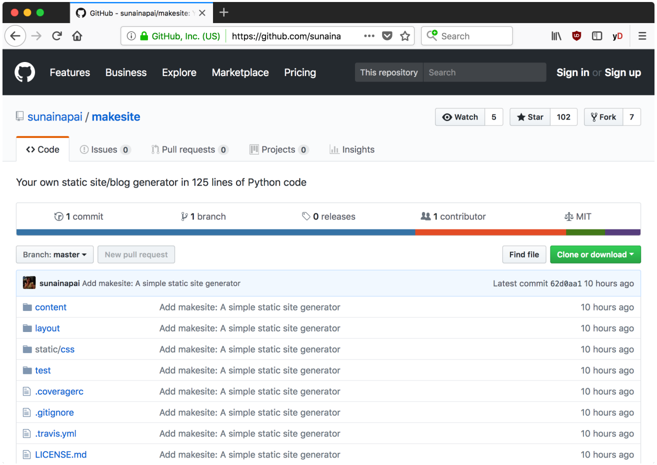
Y **Hacker News** new | comments | show | ask | jobs | submit login

1. ▲ Repair cafés waging war on throwaway culture (theguardian.com)
   231 points by wcunning 4 hours ago | hide | 109 comments
2. ▲ Hack everything without fear (drewdevault.com)
   29 points by RX14 2 hours ago | hide | 8 comments
3. ▲ Mathematics for Computer Science: Readings (mit.edu)
   235 points by rspivak 3 hours ago | hide | 36 comments
4. ▲ The Nightmare Letter: A Subject Access Request Under GDPR (linkedin.com)
   201 points by jjp 6 hours ago | hide | 232 comments
5. ▲ 'Looks good enough to eat': inside the home of Japan's fake food industry (theguardian.com)
   3 points by sergeant3 12 minutes ago | hide | discuss
6. ▲ TinyVM – A small and easy to understand virtual machine in C (github.com)
   89 points by jxub 6 hours ago | hide | 6 comments
7. ▲ 50M Facebook profiles harvested for Cambridge Analytica in major data breach (theguardian.com)
   238 points by tsneed290 5 hours ago | hide | 104 comments
8. ▲ Tail recursion in Python (chrispenner.ca)
   128 points by zweedeend 9 hours ago | hide | 57 comments
9. ▲ Inheritance Often Doesn't Make Sense (sicpers.info)
   284 points by signa11 12 hours ago | hide | 155 comments
10. ▲ Launch HN: Rhythmm (YC W18) – Create channels and broadcast them to an audience
    14 points by franzwarning 2 hours ago | hide | 8 comments
11. ▲ Show HN: Makesite – A static site generator in 125 lines of Python (github.com)
    77 points by sunainapai 8 hours ago | hide | 30 comments
12. ▲ Kikkuli, 1345 BCE: Training the Chariot Horse (imh.org)
    9 points by benbreen 3 hours ago | hide | discuss
13. ▲ More State Machine Love: From Reflection to Statecharts (raganwald.com)
    86 points by weinzierl 8 hours ago | hide | 13 comments
14. ▲ Power Outage at Samsung Fab Destroys 3.5% of Global NAND Flash Output for March (anandtech.com)
    104 points by jonbaer 13 hours ago | hide | 51 comments
15. ▲ How Europeans evolved white skin (2015) (sciencemag.org)
    88 points by pulisse 4 hours ago | hide | 43 comments

Figure: Makesite trending on the Hacker News front page on 17 Mar 2018

Hacker News “Show HN” story: https://news.ycombinator.com/item?id=16606408

## GitHub Repository



GitHub - sunainapai/makesite
GitHub, Inc. (US) https://github.com/sunaina
Search
Features Business Explore Marketplace Pricing
This repository Search
Sign in or Sign up
Watch 5
Star 102
Fork 7
Code
Issues 0
Pull requests 0
Projects 0
Insights

sunainapai / makesite

Your own static site/blog generator in 125 lines of Python code

1 commit · 1 branch · 0 releases · 1 contributor · MIT

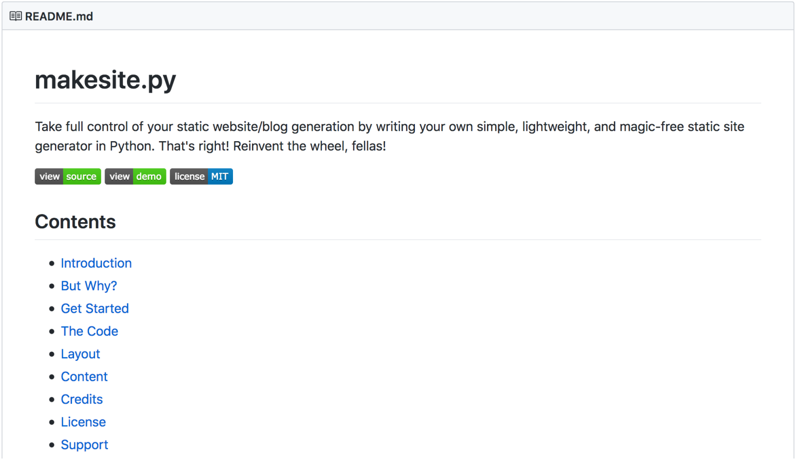
sunainapai Add makesite: A simple static site generator — Latest commit 62d0aa1 10 hours ago

| | | |
|---|---|---|
| content | Add makesite: A simple static site generator | 10 hours ago |
| layout | Add makesite: A simple static site generator | 10 hours ago |
| static/css | Add makesite: A simple static site generator | 10 hours ago |
| test | Add makesite: A simple static site generator | 10 hours ago |
| .coveragerc | Add makesite: A simple static site generator | 10 hours ago |
| .gitignore | Add makesite: A simple static site generator | 10 hours ago |
| .travis.yml | Add makesite: A simple static site generator | 10 hours ago |
| LICENSE.md | Add makesite: A simple static site generator | 10 hours ago |

Figure: Screenshot of GitHub repository taken on 17 Mar 2018

GitHub repository: https://github.com/sunainapai/makesite

## Project README



README.md

# makesite.py

Take full control of your static website/blog generation by writing your own simple, lightweight, and magic-free static site generator in Python. That's right! Reinvent the wheel, fellas!

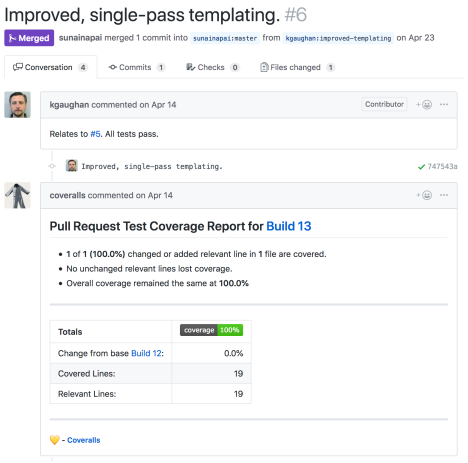
### Contents

- Introduction
- But Why?
- Get Started
- The Code
- Layout
- Content
- Credits
- License
- Support

Figure: Screenshot of project README

GitHub repository: https://github.com/sunainapai/makesite

## Travis CI and Coveralls Integration


Improved, single-pass templating. #6

Merged sunainapai merged 1 commit into `sunainapai:master` from `kgaughan:improved-templating` on Apr 23

kgaughan commented on Apr 14 Contributor

Relates to #5. All tests pass.

`Improved, single-pass templating.` ✓ 747543a

coveralls commented on Apr 14

**Pull Request Test Coverage Report for Build 13**

- **1** of **1 (100.0%)** changed or added relevant line in **1** file are covered.
- No unchanged relevant lines lost coverage.
- Overall coverage remained the same at **100.0%**

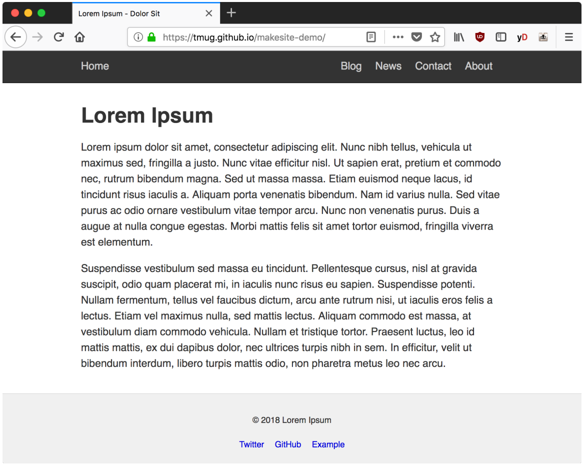
| **Totals** | coverage 100% |
|---|---|
| Change from base Build 12: | 0.0% |
| Covered Lines: | 19 |
| Relevant Lines: | 19 |

💛 - **Coveralls**

Figure: Unit testing and coverage for pull requests and commits

Pull Request URL: https://github.com/sunainapai/makesite/pull/6

## Demo Static Website



**Lorem Ipsum**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc nibh tellus, vehicula ut maximus sed, fringilla a justo. Nunc vitae efficitur nisl. Ut sapien erat, pretium et commodo nec, rutrum bibendum magna. Sed ut massa massa. Etiam euismod neque lacus, id tincidunt risus iaculis a. Aliquam porta venenatis bibendum. Nam id varius nulla. Sed vitae purus ac odio ornare vestibulum vitae tempor arcu. Nunc non venenatis purus. Duis a augue at nulla congue egestas. Morbi mattis felis sit amet tortor euismod, fringilla viverra est elementum.

Suspendisse vestibulum sed massa eu tincidunt. Pellentesque cursus, nisl at gravida suscipit, odio quam placerat mi, in iaculis nunc risus eu sapien. Suspendisse potenti. Nullam fermentum, tellus vel faucibus dictum, arcu ante rutrum nisi, ut iaculis eros felis a lectus. Etiam vel maximus nulla, sed mattis lectus. Aliquam commodo est massa, at vestibulum diam commodo vehicula. Nullam et tristique tortor. Praesent luctus, leo id mattis mattis, ex dui dapibus dolor, nec ultrices turpis nibh in sem. In efficitur, velit ut bibendum interdum, libero turpis mattis odio, non pharetra metus leo nec arcu.

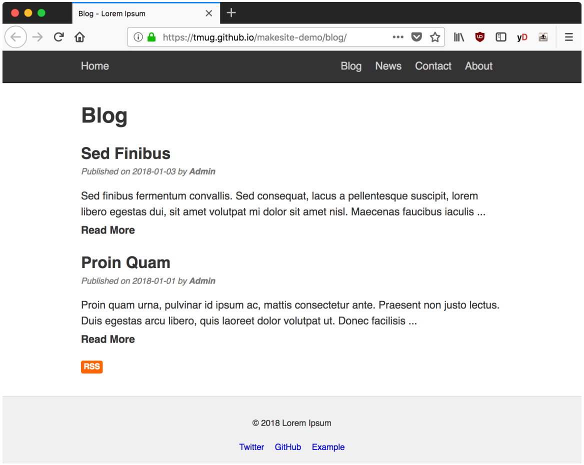
© 2018 Lorem Ipsum

Twitter GitHub Example

Figure: Screenshot of demo static website generated with makesite.py

Demo website: https://tmug.github.io/makesite-demo/

## Demo Static Blog


Blog - Lorem Ipsum

https://tmug.github.io/makesite-demo/blog/

Home Blog News Contact About

Blog

Sed Finibus

*Published on 2018-01-03 by **Admin***

Sed finibus fermentum convallis. Sed consequat, lacus a pellentesque suscipit, lorem libero egestas dui, sit amet volutpat mi dolor sit amet nisl. Maecenas faucibus iaculis ...

**Read More**

Proin Quam

*Published on 2018-01-01 by **Admin***

Proin quam urna, pulvinar id ipsum ac, mattis consectetur ante. Praesent non justo lectus. Duis egestas arcu libero, quis laoreet dolor volutpat ut. Donec facilisis ...

**Read More**

RSS

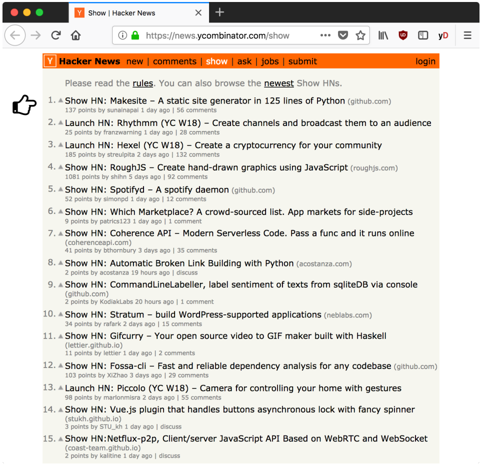
© 2018 Lorem Ipsum

Twitter GitHub Example

Figure: Screenshot of demo static blog generated with makesite.py

Demo blog: https://tmug.github.io/makesite-demo/blog/

## On “Show HN” Page


Show | Hacker News
https://news.ycombinator.com/show
Hacker News new | comments | show | ask | jobs | submit
login

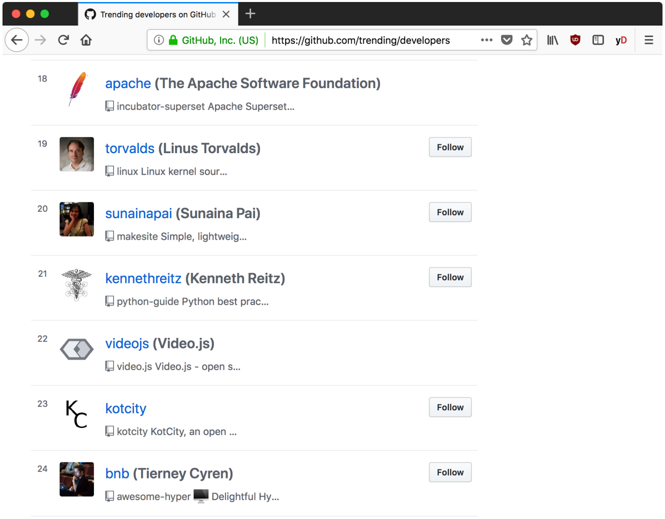
Please read the rules. You can also browse the newest Show HNs.

1. ▲ Show HN: Makesite – A static site generator in 125 lines of Python (github.com)
   137 points by sunainapai 1 day ago | 56 comments
2. ▲ Launch HN: Rhythmm (YC W18) – Create channels and broadcast them to an audience
   25 points by franzwarning 1 day ago | 28 comments
3. ▲ Launch HN: Hexel (YC W18) – Create a cryptocurrency for your community
   185 points by streulpita 2 days ago | 132 comments
4. ▲ Show HN: RoughJS – Create hand-drawn graphics using JavaScript (roughjs.com)
   1081 points by shihn 5 days ago | 92 comments
5. ▲ Show HN: Spotifyd – A spotify daemon (github.com)
   52 points by simonpd 1 day ago | 12 comments
6. ▲ Show HN: Which Marketplace? A crowd-sourced list. App markets for side-projects
   9 points by patrics123 1 day ago | 1 comment
7. ▲ Show HN: Coherence API – Modern Serverless Code. Pass a func and it runs online (coherenceapi.com)
   41 points by bthornbury 3 days ago | 35 comments
8. ▲ Show HN: Automatic Broken Link Building with Python (acostanza.com)
   2 points by acostanza 19 hours ago | discuss
9. ▲ Show HN: CommandLineLabeller, label sentiment of texts from sqliteDB via console (github.com)
   2 points by KodiakLabs 20 hours ago | 1 comment
10. ▲ Show HN: Stratum – build WordPress-supported applications (neblabs.com)
    34 points by rafark 2 days ago | 15 comments
11. ▲ Show HN: Gifcurry – Your open source video to GIF maker built with Haskell (lettier.github.io)
    11 points by lettier 1 day ago | 2 comments
12. ▲ Show HN: Fossa-cli – Fast and reliable dependency analysis for any codebase (github.com)
    103 points by XiZhao 3 days ago | 29 comments
13. ▲ Launch HN: Piccolo (YC W18) – Camera for controlling your home with gestures
    98 points by marlonmisra 2 days ago | 55 comments
14. ▲ Show HN: Vue.js plugin that handles buttons asynchronous lock with fancy spinner (stukh.github.io)
    3 points by STU_kh 1 day ago | discuss
15. ▲ Show HN:Netflux-p2p, Client/server JavaScript API Based on WebRTC and WebSocket (coast-team.github.io)
    2 points by kalitine 1 day ago | discuss

Figure: Makesite trending at the top of the “Show HN“ posts on 18 Mar 2018

Hacker News “Show HN” story: https://news.ycombinator.com/item?id=16606408

# On GitHub Trending Page



Trending developers on GitHub

GitHub, Inc. (US) https://github.com/trending/developers

18 apache (The Apache Software Foundation)
incubator-superset Apache Superset...

19 torvalds (Linus Torvalds) Follow
linux Linux kernel sour...

20 sunainapai (Sunaina Pai) Follow
makesite Simple, lightweig...

21 kennethreitz (Kenneth Reitz) Follow
python-guide Python best prac...

22 videojs (Video.js)
video.js Video.js - open s...

23 kotcity Follow
kotcity KotCity, an open ...

24 bnb (Tierney Cyren) Follow
awesome-hyper Delightful Hy...

Figure: Nice to be trending up there with Linus Torvalds! 😛

## Summary of Work

- Write code.
- Write unit tests.
- Write README.
- Publish on GitHub.
- Integrate with Travis CI and Coveralls.
- Set up an online demo.
- Announce on Hacker News.

## Results

- Got a small community of Python developers who use the project.
- Received pull requests to improve the software.
- Became a part of a very welcoming community.

Thank You!