

Livemark

[express everything in Markdown]

DATA JOURNALISM

SOFTWARE EDUCATION

PYTHON DEVELOPMENT

Evgeny Karev

evgeny.karev@okfn.org

livemark.frictionlessdata.io

FOSDEM 2022

The screenshot shows the GitHub repository for Livemark. The main content area displays the README, which includes a navigation menu on the left, a search bar at the bottom, and a list of links on the right. The README text describes Livemark as a static site generator that extends Markdown with interactive charts, tables, scripts, and more. It mentions that the document is written and published in Livemark notation and provides a link to the Jinja templates. The 'Logic' section explains that Livemark uses the Jinja templating language and the Frictionless Framework for processing data. A code block shows a Jinja template snippet for extracting data from a CSV file. The 'Table' section includes a link to a tutorial on Handsontable. The right sidebar shows the repository's star count (45) and a list of links to external resources like Frictionless, Discord, Twitter, Report, Fork, and Edit.

Livemark Written in Livemark (2021-08-23 16:27) Star 45

build [passito](#) coverage [93%](#) [pypi](#) [10,451](#) [codebase](#) [github](#) [support](#) [discord](#)

This document is completely [written](#) and [published](#) in Livemark notation.

Livemark is a static site generator that extends Markdown with interactive charts, tables, scripts, and more.

Logic

<https://jinja.palletsprojects.com/en/3.0.x/templates/>

Livemark process your document using the Jinja templating language. Inside templates, you can use [Frictionless Framework](#) as a [frictionless](#) variable to work with tabular data. It's a high-level preprocessing so you can combine Logic with other syntax, such as [Table](#) or [Chart](#):

```
{% for car in frictionless.extract('data/cars.csv', layout={'LimitRows': 5}) %}
- {{ car.brand }} {{ car.model }}: ${{ car.price }}
{% endfor %}
```

- Volkswagen Vento: \$785
- Hyundai Verna: \$774
- Skoda Rapid: \$756
- Suzuki Ciaz: \$725
- Renault Scala: \$724

Table

<https://handsontable.com/docs/9.0.0/tutorial-introduction.html>

Livemark supports CSV table rendering using Handsontable, which you can see in the example below (replace the single quotes with

Frictionless [↗](#)
Discord [↗](#)
Twitter [↗](#)
Report [↗](#)
Fork [↗](#)
Edit [↗](#)



Open Knowledge
Foundation

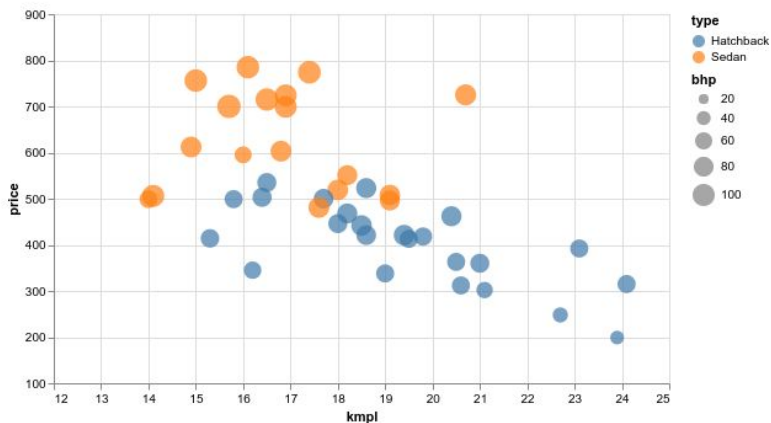
@okfn

<https://okfn.org>

Example [show me the code!]

```
'''yaml chart
data:
  url: data/cars.csv
mark: circle
selection:
  brush:
    type: interval
# other options are omitted
width: 500
height: 300
'''
```

CHART



```
'''yaml table
data: data/cars.csv
width: 600
order:
  - [3, 'desc']
columns:
  - data: type
  - data: brand
  - data: model
  - data: price
  - data: kmpl
  - data: bhp
'''
```

TABLE

Show entries

Search:

type	brand	model	price	kmpl	bhp
Sedan	Volkswagen	Vento	785	16.1	104
Sedan	Hyundai	Verna	774	17.4	106
Sedan	Skoda	Rapid	756	15.0	104
Sedan	Suzuki	Ciaz	725	20.7	91
Sedan	Renault	Scala	724	16.9	98
Sedan	Suzuki	SX4	715	16.5	103
Sedan	Fiat	Linea	700	15.7	112
Sedan	Nissan	Sunny	699	16.9	98
Sedan	Fiat	Linea Classic	612	14.9	89
Sedan	Toyota	Etios	603	16.8	89

Showing 1 to 10 of 42 entries

Previous 2 3 4 5 Next

```
{% for car in frictionless
- {{ car.brand }} {{ car.
{% endfor %}
```

- Volkswagen Vento: \$785
- Hyundai Verna: \$774
- Skoda Rapid: \$756
- Suzuki Ciaz: \$725
- Renault Scala: \$724

LOGIC

Principles [philosophy and ideas behind Livemark]

- **Markdown -> HTML**
- **For Python community (with fast build)**
- **Express everything in Markdown (using standards):**
VEGA/JINJA/HTML/CSS/JQUERY/BS/BASH/ETC
- **Extreme and simple extensibility (with plugins)**
- **Back to good old WWW (decentralized!)**

Data Journalism [data collection + visualization]

COVID-19 Tracker

World

Regions

Countries

Excess Mortality

Rolling Immunity

Data Collection

Contribute

Forum

Status

Deaths

New Cases

Vaccinations

Locations

About

COVID-19 Tracker

Written in Livemark 
(2021-10-26 06:03)

[build](#) [passing](#) [codebase](#) [github](#) [support](#) [discord](#)

Last data update: **2021-10-25** (this site is being updated on a nightly basis).

Coronavirus disease 2019 (COVID-19), also known as **COVID** and the coronavirus, is a contagious disease caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). The data dashboard has been created by the Frictionless Data team and it's open for contributing for anyone who is interested. The main COVID-19 data resources we use for this project are [Our World in Data](#) core data and [Excess Mortality](#) research.

Status

The main three core metrics we track: total deaths, total cases, and all the vaccination shots taken (worldwide):

Deaths

5.0M

Cases

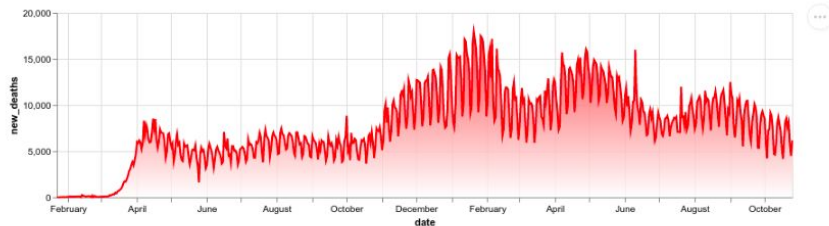
244.1M

Vaccinations

6.9B

Deaths

Here is a timeline of the worldwide deaths caused by the COVID-19 pandemic:



Search...

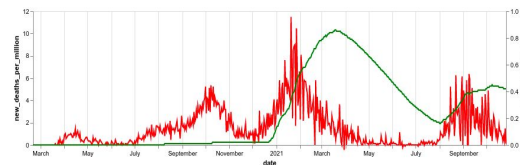
Deaths

Here is an interactive map showing the count of total deaths per country:



Rolling Immunity

Here is deaths/rolling immunity chart. Read more about [rolling immunity](#):



Locations

Here is a summary of the key pandemic data by region:

Show entries

Location	Deaths	Deaths/1M	Cases	Cases/1M	Shots	Shots/100
Europe	1292229	1725.358	63815558	85205.223	838213821	111.92
South America	1168007	2689.648	38272164	88131.884	499206162	114.96
Asia	1166861	249.412	78851114	16854.129	4648742315	99.37
North America	1116099	1870.825	54827177	91902.275	666627322	111.74
Africa	217339	158.239	8477472	6172.228	188208162	13.7
Oceania	2706	62.61	250045	5785.406	43977740	101.75

Showing 1 to 6 of 6 entries

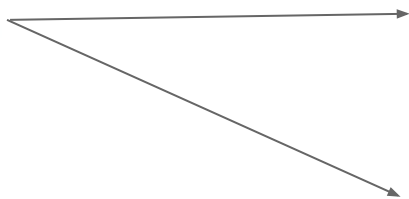
Previous Next

Software Education [live examples + book style]

Getting Started

Let's print "Hello World" in Python:

```
```python script  
print("Hello World")
```
```



Getting Started

Let's print "Hello World" in Python:

```
print("Hello World")
```

```
Hello World
```

Python Development [task runner + docs portal]

Livemark

Introduction

- Getting Started >
- Livemark Markdown
- Livemark Components
- Plugin System
- Universe
- Blog

Logic

- Table
- Chart
- Script
- Markup
- Content
- Sidebar
- Scroll

Livemark

Written in Livemark [🔗](#)
(2021-08-23 16:27)

[build](#) [passing](#) [coverage](#) [93%](#) [pypi](#) [v0.40.0](#) [codebase](#) [github](#) [support](#) [discord](#)

This document is completely [written](#) and [published](#) in Livemark notation

Livemark is a static site generator that extends Markdown with interactive charts, tables, scripts, and more.

Logic

<https://jinja.palletsprojects.com/en/3.0.x/templates/>

Livemark process your document using the Jinja templating language. Inside templates, you can use [Frictionless Framework](#) as a `frictionless` variable to work with tabular data. It's a high-level preprocessing so you can combine Logic with other syntax, such as Table or Chart:

```
{% for car in frictionless.extract('data/cars.csv', layout={"limitRows": 5}) %}
- {{ car.brand }} {{ car.model }}: ${ car.price }}
{% endfor %}
```

- Volkswagen Vento: \$785
- Hyundai Verna: \$774
- Skoda Rapid: \$756
- Suzuki Ciaz: \$725
- Renault Scala: \$724

Table

<https://handsontable.com/docs/9.0.0/tutorial-introduction.html>

Livemark supports CSV table rendering using Handsontable, which you can see in the example below (replace the single quotes with

Projects

Written in Livemark [🔗](#)
(2021-10-26 22:29)

covid-tracker

A livemark tracking COVID-19 disease pandemic
[Github](#) [🔗](#)



data-packages

A livemark listing data packages hosted on Github
[Github](#) [🔗](#)



[Livemark](#)
[Data Journalism](#)
[Meet Livemark](#)
[Using Markdown](#)
[Markdown](#)

chart

Current

Current Livemark Version:

v0.85.0 | [PyPi](#)

Features [built-in plugins we already have]

- **Markdown:** Task, Logic, Table, Chart, Script, Markup, Map, Audio, File, Image, Remark, Package, Reference, Notebook, Report, Resource, Schema, Video, Pipeline
- **Appearance:** Brand, Notes, Rating, About, Display, Mobile, Source, News
- **Navigation:** Pages, Topics, Links, Search, Signs, Infinity, Pagination
- **Blogging:** blog
- **Others:** Cleanup, Counter, Github, Prepare, Redirect

Customization [config and plugins]

```
brand: livemark.yaml
  text: Livemark Project
about:
  text: A Livemark project template
github:
  user: frictionlessdata
  repo: livemark-project
topics:
  selector: h2
site:
  favicon: assets/favicon.ico
  styles:
    - style.css
  scripts:
    - script.js
pages:
  items:
    - path: index
    - path: pages/data
    - path: pages/contrib
```

```
from livemark import Plugin
```

```
class CustomPlugin(Plugin):
```

```
    identity = "custom"
```

```
    # Process
```

```
    def process_markup(self, markup):
```

```
        markup.add_markup("<span>(template)<span>", target="h1")
```

plugin.py

output

Livemark Project (template)

build passing

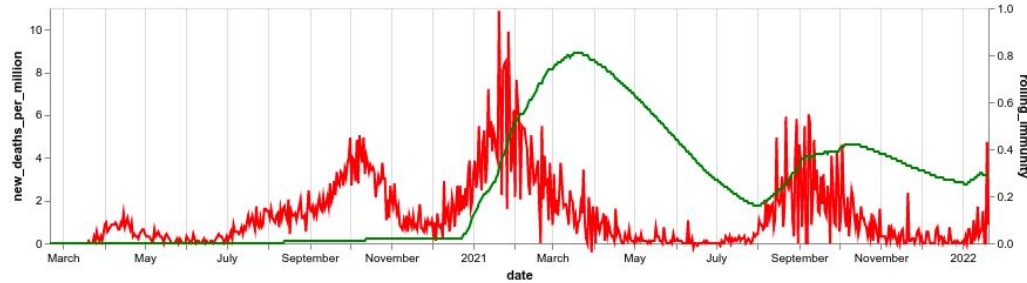
codebase github

support discord

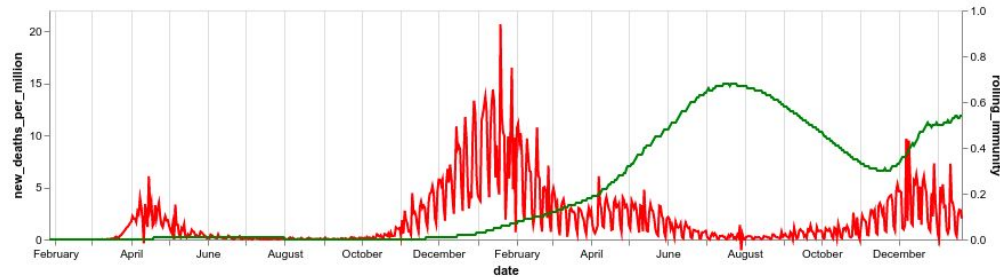


Rolling Immunity [insight from data]

Israel



Germany



| Location | Immunity | Deaths/1M | Cases/1M |
|-------------|----------|-----------|----------|
| Cuba | 0.87 | 0.265 | 290.599 |
| South Korea | 0.82 | 0.674 | 99.414 |
| Australia | 0.64 | 2.249 | 3099.76 |
| Peru | 0.64 | | |
| Denmark | 0.61 | 2.384 | 5319.696 |
| New Zealand | 0.59 | 0.0 | 12.661 |
| Vietnam | 0.58 | 1.595 | 173.692 |
| Greece | 0.57 | 8.733 | 1783.781 |
| Austria | 0.55 | | 1751.965 |
| Chile | 0.55 | 0.929 | 498.615 |

Location Immunity Deaths/1M Cases/1M

Work in Progress [feedback is welcome!]

- Livemark is approaching v1 release in the middle of 2022
- Currently, any feedback and ideas are priceless
- Thanks for your time!