

Computational Reproducibility:

A how-to guide based on
The Turing Way

Martina G. Vilas (she/her)

@martinagvilas



1. what is *The Turing Way*?
2. guide to computational reproducibility

1. what is *The Turing Way*?

2. guide to computational reproducibility



Book:
the-turing-way.netlify.app/

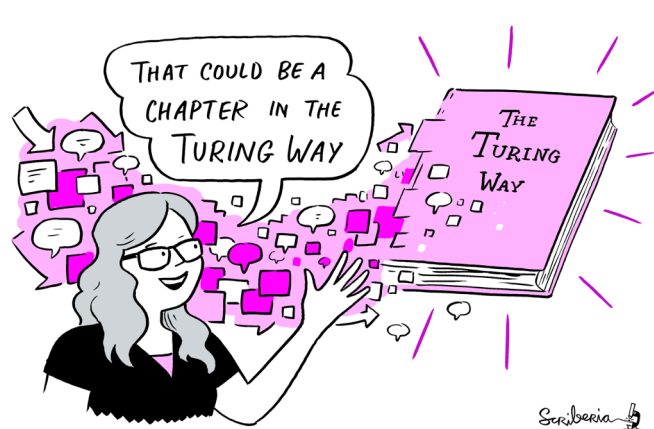
GitHub:
github.com/alan-turing-institute/the-turing-way

Twitter:
twitter.com/turingway

Email:
theturingway@gmail.com

CC-BY 4.0, The Turing Way

A Book



An Open Source Project



A Community



A Culture of Collaboration



The Turing Way



Kirstie Whitaker

Program Lead for Tools, Practices and
Systems at the Allan Turing Institute

The Turing Way

- online book
- moonshot goal:
make reproducible
research “too easy
not to do”

1. Introduction
2. Reproducibility
3. Open Research
4. Version Control
5. Collaborating on GitHub/GitLab
6. Research Data Management
7. Reproducible Environments
8. Testing
9. Reviewing
10. Continuous Integration
11. Reproducible Research with Make
12. Risk Assessment

Welcome to the Turing Way

The Turing Way is a lightly opinionated guide to reproducible data science.

Our goal is to provide all the information that researchers need at the start of their projects to ensure that they are easy to reproduce at the end.

This also means making sure PhD students, postdocs, PIs and funding teams know which parts of the “responsibility of reproducibility” they can affect, and what they should do to nudge data science to being more efficient, effective and understandable.

A bit more background

Reproducible research is necessary to ensure that scientific work can be trusted. Funders and publishers are beginning to require that publications include access to the underlying data and the analysis code. The goal is to ensure that all results can be independently verified and built upon in future work. This is sometimes easier said than done. Sharing these research outputs means understanding data management, library sciences, software development, and continuous integration techniques: skills that are not widely taught or expected of academic researchers and data scientists.


The Turing Way is a handbook to support students, their supervisors, funders and journal editors

reproducible research

same analysis steps on
the same dataset
produces same answer

		Data	
		Same	Different
Analysis	Same	Reproducible	Replicable
	Different	Robust	Generalisable

The Turing Way chapters on reproducibility



The Turing Way

Search this book...

Welcome

[Guide for Reproducible Research](#)


- Overview
- Open Research
- Version Control
- Licensing
- Research Data Management
- Reproducible Environments
- BinderHub
- Code quality
- Code Testing
- Code Reviewing Process
- Continuous Integration
- Reproducible Research with Make
- Research Compendia
- Credit for Reproducible Research
- Risk Assessment
- Case Studies

Guide for Reproducible Research

This guide covers topics related to skills, tools and best practices for research reproducibility.

The Turing Way defines reproducibility in data research as data and code being available to fully rerun the analysis.

There are several definitions of reproducibility in use, and we discuss these in more detail in the [Definitions of Reproducibility](#) section of this chapter. While it is absolutely fine for us each to use different words, it will be useful for you to know how The Turing Way defines reproducibility to avoid misunderstandings when reading the rest of the handbook.



<https://the-turing-way.netlify.app>

@martinagvilas, @turingway #TuringWay



The Turing Way

🔍 Search this book...

Welcome

Guide for Reproducible Research

Guide for Project Design

Guide for Communication

Guide for Collaboration

Guide for Ethical Research

Community Handbook

Afterword

Visit our [GitHub Repository](#)

This book is powered by [Jupyter Book](#)

Welcome

The Turing Way is an open source community-driven guide to reproducible, ethical, inclusive and collaborative data science.

Our goal is to provide all the information that data scientists in academia, industry, government and in the third sector need at the start of their projects to ensure that they are easy to reproduce and reuse at the end.

The book started as a guide for reproducibility, covering version control, testing, and continuous integration. But technical skills are just one aspect of making data science research “open for all”.

In February 2020, *The Turing Way* expanded to a series of books covering reproducible research, project design, communication, collaboration, and ethical research.



<https://the-turing-way.netlify.app>

@martinagvilas, @turingway #TuringWay



Book:
the-turing-way.netlify.app/

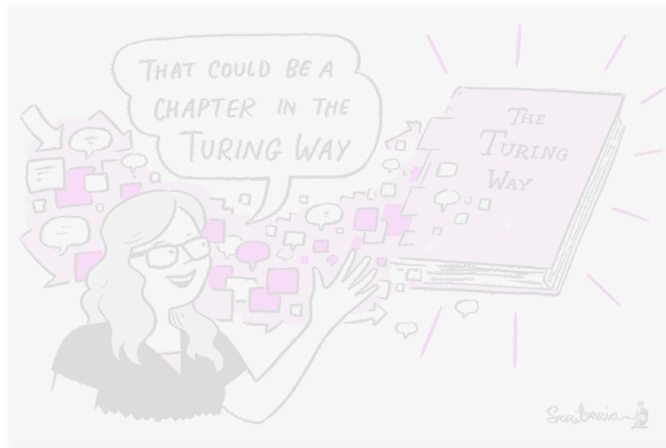
GitHub:
github.com/alan-turing-institute/the-turing-way

Twitter:
twitter.com/turingway

Email:
theturingway@gmail.com

CC-BY 4.0, *The Turing Way*

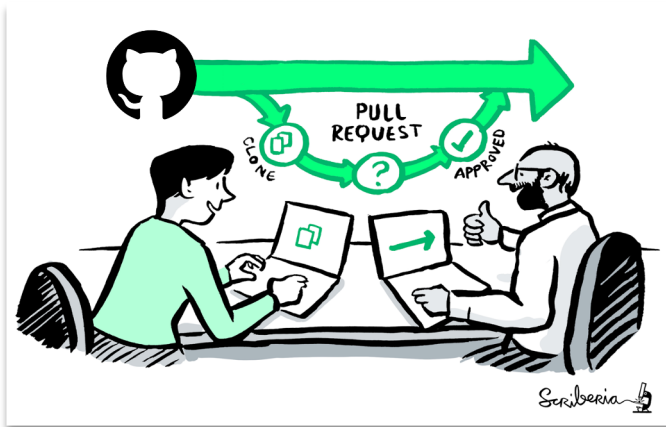
A Book



A Community



An Open Source Project

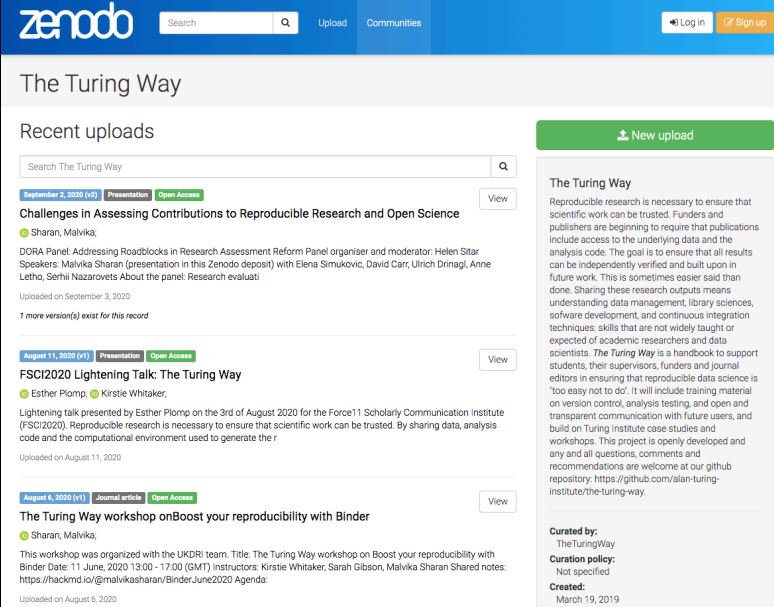


A Culture of Collaboration



an open source project

- everyone can freely read, reuse, distribute, modify and help develop
- the project belongs to *The Turing Way* community (CC-BY license)



The screenshot shows the Zenodo website interface for the 'The Turing Way' community. The header includes the Zenodo logo, a search bar, and links for 'Upload' and 'Communities'. On the right, there are links for 'Log in' and 'Sign up'. The main content area is titled 'The Turing Way' and features a 'Recent uploads' section. This section contains three entries, each with a date, version, and type (Presentation or Journal article), followed by a 'View' button. The first entry is 'Challenges in Assessing Contributions to Reproducible Research and Open Science' by Sharan, Malvika, dated September 2, 2020. The second entry is 'FSCI2020 Lightning Talk: The Turing Way' by Esther Plomp and Kirstie Whitaker, dated August 11, 2020. The third entry is 'The Turing Way workshop onBoost your reproducibility with Binder' by Sharan, Malvika, dated August 6, 2020. On the right side of the page, there is a 'New upload' button and a section titled 'The Turing Way' which provides a detailed description of the project's goals and the importance of reproducible research.

zenodo Search Upload Communities Log in Sign up

The Turing Way

Recent uploads

Search The Turing Way

September 2, 2020 (v2) Presentation Open Access View

Challenges in Assessing Contributions to Reproducible Research and Open Science

Sharan, Malvika

DORA Panel: Addressing Roadblocks in Research Assessment Reform Panel organiser and moderator: Helen Star. Speakers: Malvika Sharan (presentation in this Zenodo deposit) with Elena Simukovic, David Carr, Ulrich Drinag, Anne Letho, Serhii Nazarovets About the panel: Research evaluati

Uploaded on September 3, 2020

1 more version(s) exist for this record

August 11, 2020 (v1) Presentation Open Access View

FSCI2020 Lightning Talk: The Turing Way

Esther Plomp, Kirstie Whitaker

Lightening talk presented by Esther Plomp on the 3rd of August 2020 for the Force11 Scholarly Communication Institute (FSCI2020). Reproducible research is necessary to ensure that scientific work can be trusted. By sharing data, analysis code and the computational environment used to generate the r

Uploaded on August 11, 2020

August 6, 2020 (v1) Journal article Open Access View

The Turing Way workshop onBoost your reproducibility with Binder

Sharan, Malvika

This workshop was organized with the UKDR team. Title: The Turing Way workshop on Boost your reproducibility with Binder Date: 11 June, 2020 13:00 - 17:00 (GMT) Instructors: Kirstie Whitaker, Sarah Gibson, Malvika Sharan Shared notes: <https://hackmd.io/@malvikasharan/BinderJune2020> Agenda:

Uploaded on August 6, 2020

Curated by: TheTuringWay

Curator policy: Not specified

Created: March 19, 2019

<https://zenodo.org/communities/the-turing-way>
@martinagvilas, @turingway #TuringWay

an open source project

content hosted
on GitHub

The screenshot shows the GitHub repository page for `alan-turing-institute / the-turing-way`. The repository has 38 watchers, 623 stars, and 202 forks. It includes tabs for Code, Issues (232), Pull requests (55), Actions, Projects (2), Wiki, Security, and Insights. The main content area shows a file tree with folders like `.github`, `book`, `communications`, `conferences`, `open-life-science-mentoring`, `project_management`, `templates`, `tests`, and `workshops`, along with files like `.all-contributorsrc`, `.gitignore`, `CODE_OF_CONDUCT.md`, and `CONTRIBUTING.md`. The right sidebar contains an 'About' section, a 'Releases' section with a 'Latest' tag for `v0.0.4`, and a 'Packages' section.

alan-turing-institute / the-turing-way

Unwatch 38 Unstar 623 Fork 202

<> Code Issues 232 Pull requests 55 Actions Projects 2 Wiki Security Insights

master 86 branches 4 tags Go to file Add file Code

malvikasharan Merge pull request #1279 from alan-turing-institute/malvika... 455d5d8 4 days ago 5,105 commits

.github	Merge pull request #985 from alan-turing-institute/refine-tests	4 months ago
book	Merge pull request #1279 from alan-turing-institute/malvikasharan-r...	4 days ago
communications	Update README.md	2 months ago
conferences	Update README.md	2 months ago
open-life-science-mentoring	Updated OLS-2 Ethics README.md	11 days ago
project_management	split acknowledgement file into two subchapters	3 months ago
templates	Updating Github templates	17 months ago
tests	Update tests/no-bad-latin.py	11 days ago
workshops	minor update	last month
.all-contributorsrc	docs: update .all-contributorsrc	2 months ago
.gitignore	Merge pull request #985 from alan-turing-institute/refine-tests	4 months ago
CODE_OF_CONDUCT.md	Merge pull request #1130 from srishti-nema/add-label	4 months ago
CONTRIBUTING.md	Update CONTRIBUTING.md	2 months ago

About

Host repository for The Turing Way: a how to guide for reproducible data science

the-turing-way.netlify.app

hut23 hut23-270 hut23-396

Readme

View license

Releases 4

v0.0.4 new chapter: Conti... Latest on 28 May 2019

+ 3 releases

Packages

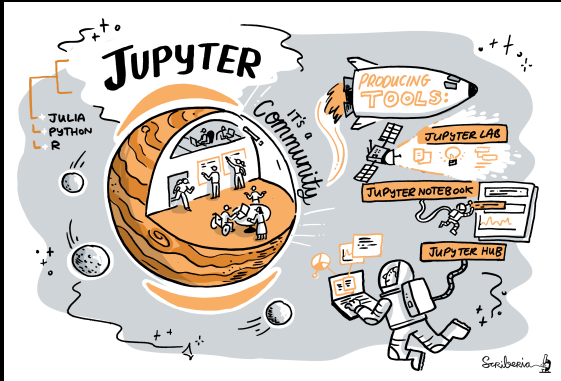
No packages published

Publish your first package

<https://github.com/alan-turing-institute/the-turing-way>
@martinagvilas, @turingway #TuringWay

an open source project

built with other Open
Source tools



jupyter {book}

Search this book...

GET STARTED

- Overview and installation
- Build your book
- Publish your book online
- Configure book settings
- Table of Contents structure
- Types of content source files

WRITE BOOK CONTENT

- MyST Markdown Overview
- Special content blocks
- References and citations
- Math and Equations
- Images and Figures
- Control the page layout
- Execute and cache your pages
- Formatting code outputs

MAKE YOUR BOOK INTERACTIVE

- Launch buttons for interactivity
- Hide or remove content
- Interactive data visualizations
- Commenting and Annotation

Books with Jupyter

Jupyter Book is an open source project for building beautiful, publication-quality books and documents from computational material.

Here are some of the features of Jupyter Book:

- ✓ **Write publication-quality content in markdown**
You can write in either Jupyter markdown, or an extended flavor of markdown with [publishing features](#). This includes support for rich syntax such as [citations](#) and [cross-references](#), [math](#) and [equations](#), and [figures](#).
- ✓ **Write content in Jupyter Notebooks**
This allows you to include your code and outputs in your book. You can also write notebooks [entirely in markdown](#) to execute when you build your book.
- ✓ **Execute and cache your book's content**
For `.ipynb` and markdown notebooks, execute code and insert the latest outputs into your book. In addition, [cache](#) and [re-use](#) outputs to be used later.
- ✓ **Insert notebook outputs into your content**
Generate outputs as you build your documentation, and insert them in-line with your content across pages.
- ✓ **Add interactivity to your book**
You can [toggle cell visibility](#), include [interactive outputs](#) from Jupyter, and [connect with online services](#) like Binder.
- ✓ **Generate a variety of outputs**
This includes single- and multi-page websites, as well as [PDF outputs](#).
- ✓ **Build books with a simple command-line interface**
You can quickly generate your books with one command, like so: `jupyter-book build mybook/`

This website is built with Jupyter Book! You can browse its contents to the left to see what is possible.

Get involved with Jupyter Book!

Jupyter Book is an open community that welcomes your feedback, input, and contributions!

[Open an issue](#) to provide feedback and new ideas, and to help others.

Contents

- Get started
- A Small Example Project
- Under the hood - the components of Jupyter Book
- Contribute to Jupyter Book
- Acknowledgements

<https://jupyterbook.org/>
@martinagvilas, @turingway #TuringWay

an open source project

2020 jupyterCON

Module 1: Welcome to the tutorial

Creating a Jupyter Book with The Turing Way

by Malvika Sharan
@malvikasharan



2020 jupyterCON

module 6:

Jupyter Book CI/CD

by Sarah Gibson
@sgibson91




2020 jupyterCON

module 3:

Jupyter Book set-up

by Martina Vilas
@martinavilas



jupyter 3-setup-jupyterbook Last Checkpoint: 09/20/2020 (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help

Not Trusted Python 3

Creating a Jupyter Book with The Turing Way

Module 3: Creating a Jupyter Book with The Turing Way

Learning Objective:

- Explain what files exist in the repository that we will use for the hands on session in this module (if you haven't already, please download the data required for this tutorial described in [module 1](#))
- Explore the important/minimal components for creating a Jupyter Book
- Build the first minimal version of the Jupyter Book locally using example files from *The Turing Way*

VIDEO

Introduction to Jupyter Book

Welcome! In this Jupyter Notebook we will introduce the basic commands to generate your first Jupyter Book.

In the previous module, we briefly looked into the awesome and very detailed [documentation](#) of Jupyter Book, and its [GitHub repository](#).

Jupyter Book has a [command-line interface](#), so in this tutorial we will show you how to build your book using iPython's special syntax that lets you execute shell commands from a Jupyter Notebook. In this example we will do so by prefixing `!` in each cell.

TIP: If you are unfamiliar with executing shell commands from Jupyter Notebooks, read this [tutorial by Jake VanderPlas](#).

Creating the content of your book based on *The Turing Way*

In order to build our Jupyter Book, we first need to create a folder where all the content of your book exists. This is also where Jupyter Book will create the `html` files in order to host the book online.

As demonstrated for *The Turing Way*, we will store all the content of our book in a folder named `book` located in the main repository (it doesn't need to be named this way for Jupyter Book to work). Let's create it:

```
In [ ]: 1 mkdir ../book/
```

<https://github.com/martinagvilas/tutorial-jupyterbook-with-turing-way/>
@martinagvilas, @turingway #TuringWay



Book:
the-turing-way.netlify.app/

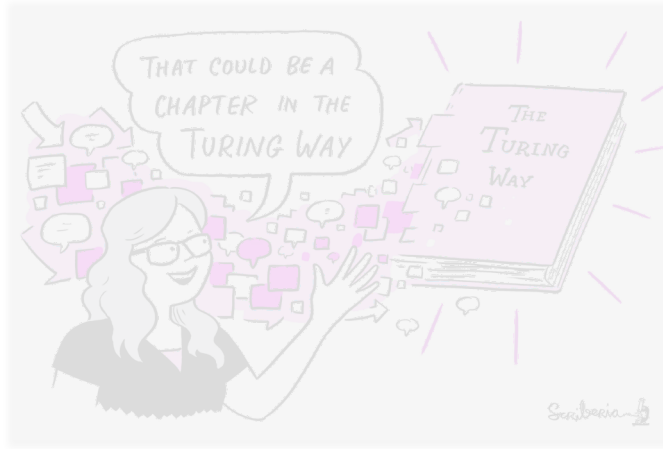
GitHub:
github.com/alan-turing-institute/the-turing-way

Twitter:
twitter.com/turingway

Email:
theturingway@gmail.com

CC-BY 4.0, *The Turing Way*

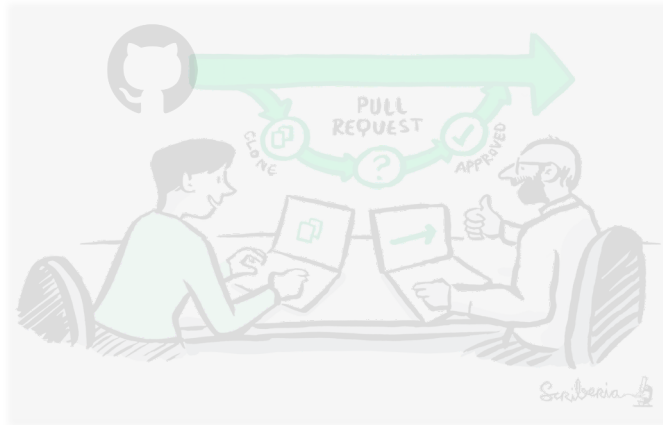
A Book



A Community



An Open Source Project



A Culture of Collaboration

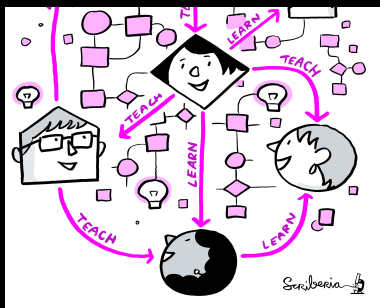


📁 .github	Remove prettier configuration
📁 book	minor update
📁 communications	Fix typos
📁 conferences	Add KW formatting pedantry
📁 project_management	Update online-collaboration-cafe.md
📁 templates	Updating Github templates
📁 tests	Add "et cetera" as a deprecated Latinism
📁 workshops	Remove mis-pasted text

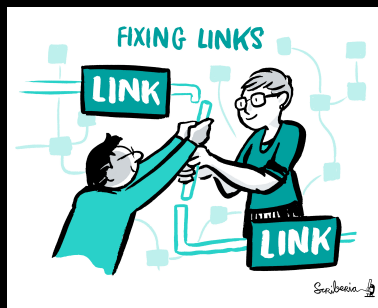


📄 .all-contributorsrc	Merge pull request #991 from alan-turing-institute/all-contributors/a...	5 days ago
📄 .gitignore	ignore pptx in workshop folder	9 months ago
📄 .travis.yml	add html-proof file again	last month
📄 <u>CODE_OF_CONDUCT.md</u>	her -> their	6 months ago
📄 <u>CONTRIBUTING.md</u>	Update CONTRIBUTING.md	2 months ago
📄 GOVERNANCE.md	Read through months later	5 months ago
📄 LICENSE.md	Fix typo in licence	2 months ago
📄 <u>README.md</u>	Merge pull request #991 from alan-turing-institute/all-contributors/a...	5 days ago
📄 book_skeleton.md	Update book_skeleton.md	13 months ago
📄 <u>contributors.md</u>	Add myself to contributors.md	11 months ago
📄 tips_and_tricks_survey.md	Update tips_and_tricks_survey.md	14 months ago
📄 ways_of_working.md	Adjust team contact section	5 months ago

Contribution Pathways



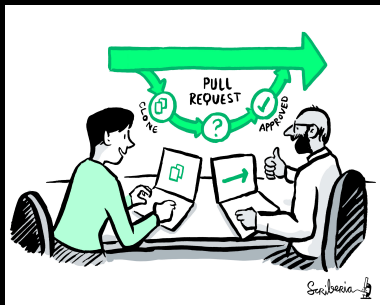
- Read/share resources



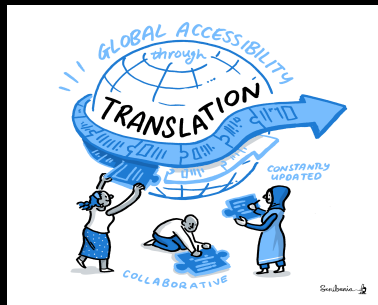
- Fix bugs/typos/errors



- Engage informally



- Review pull requests



- Help translate



- Improve our culture

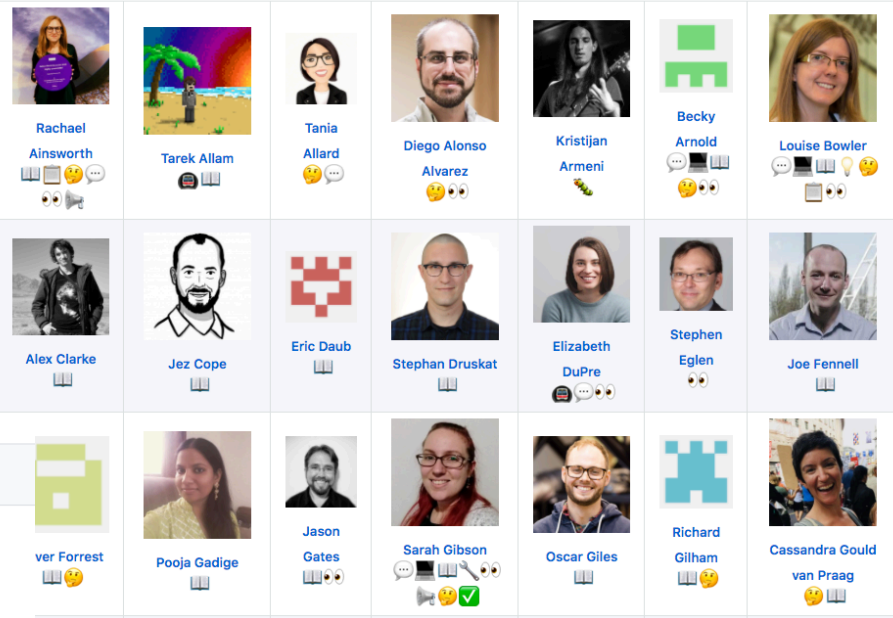
acknowledgment

every contribution gets
acknowledged



Contributors

Thanks goes to these wonderful people ([emoji key](#)):





Book:
the-turing-way.netlify.app/

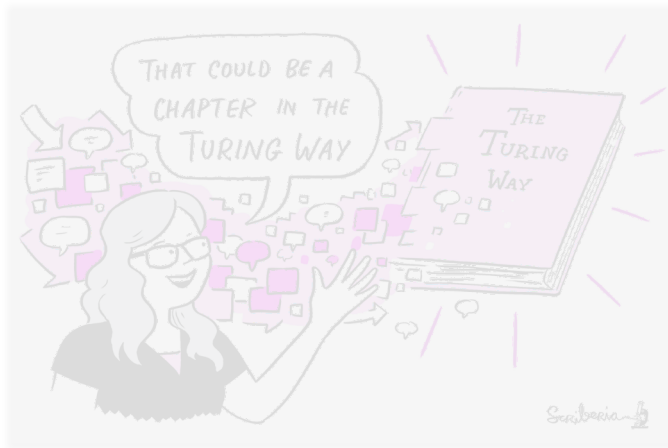
GitHub:
github.com/alan-turing-institute/the-turing-way

Twitter:
twitter.com/turingway

Email:
theturingway@gmail.com

CC-BY 4.0, *The Turing Way*

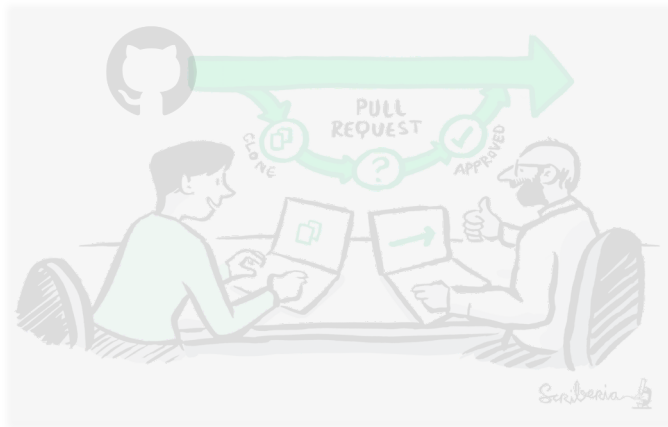
A Book



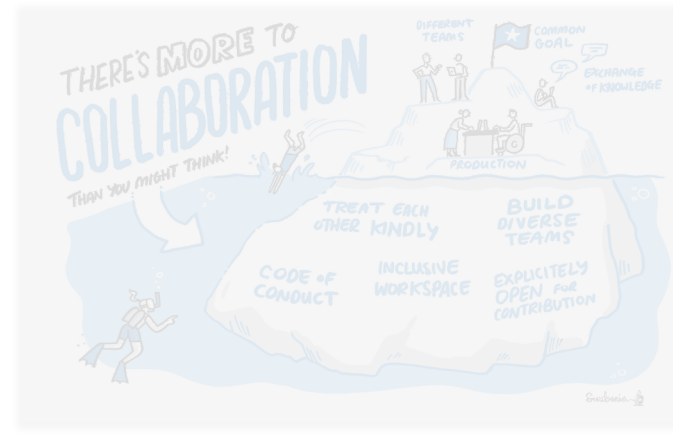
A Community



An Open Source Project

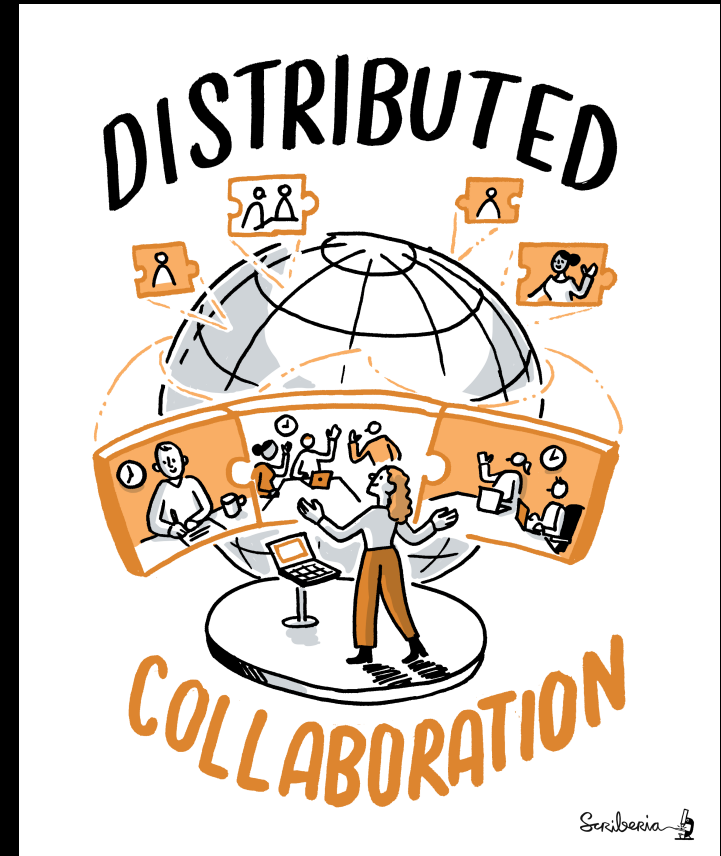


A Culture of Collaboration



open community

- anyone can join and contribute
- the direction and the goals are determined collaboratively



mentored contributions



Collaboration Cafés
& Co-working Calls



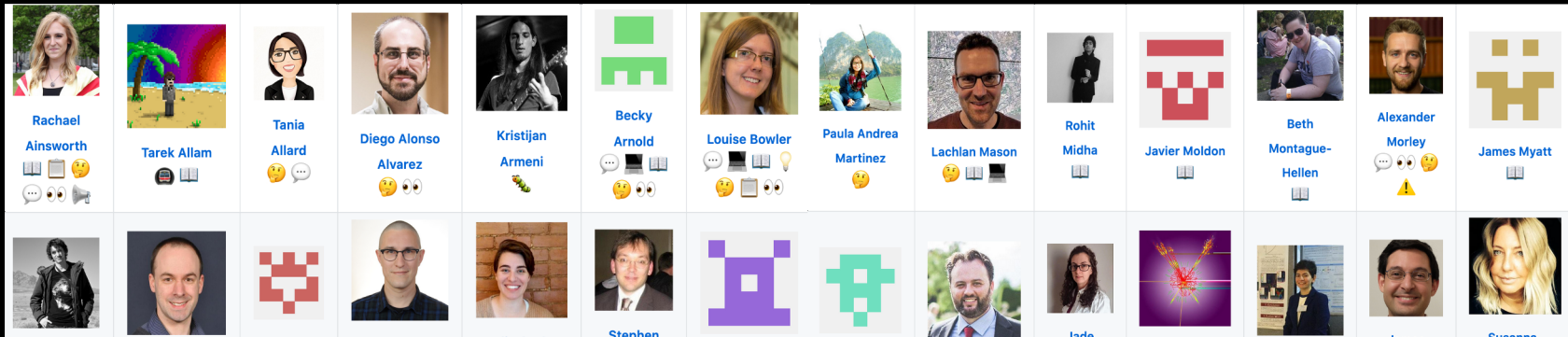
Book Dash Events

mentored contributions

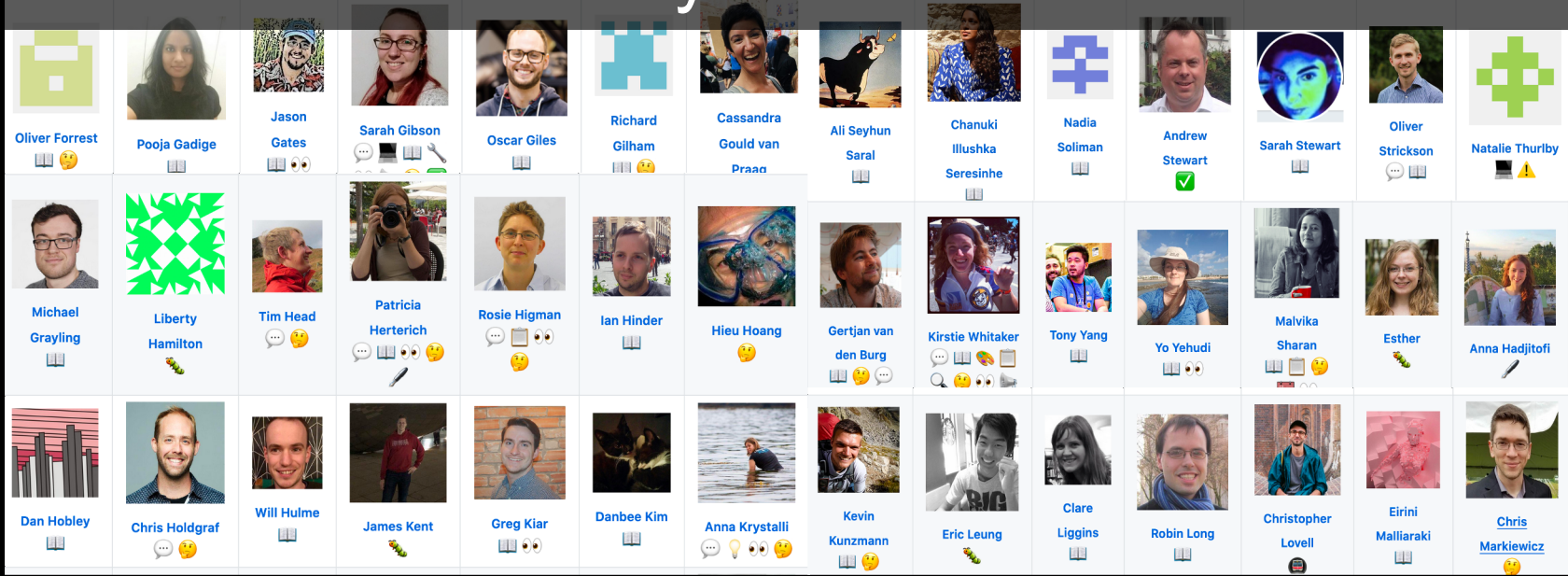
Malvika Sharan

Community manager of
The Turing Way





Join our community of contributors & friends!



Wall of all Contributors

tagged by emojis

- eventOrganizing
- financial
- fundingFinding
- ideas
- review
- security
- tool
- translation
- test
- tutorial
- talk

1. what is *The Turing Way*?

2. guide to computational reproducibility

The Turing Way chapters on reproducibility

Guide for Reproducible Research

- Overview
- Open Research
- Version Control
- Licensing
- Research Data Management
- Reproducible Environments
- BinderHub
- Code quality
- Code Testing
- Code Reviewing Process
- Continuous Integration
- Reproducible Research with Make
- Research Compendia
- Credit for Reproducible Research
- Risk Assessment
- Case Studies



The Turing Way

Q Search this book...

Welcome

Guide for Reproducible Research

- Overview
- Open Research
- Version Control
- Licensing
- Research Data Management
- Reproducible Environments
- BinderHub
- Code quality
- Code Testing
- Code Reviewing Process
- Continuous Integration
- Reproducible Research with Make
- Research Compendia
- Credit for Reproducible Research
- Risk Assessment
- Case Studies



Guide for Reproducible Research

This guide covers topics related to skills, tools and best practices for research reproducibility.

The Turing Way defines reproducibility in data research as data and code being available to fully rerun the analysis.

There are several definitions of reproducibility in use, and we discuss these in more detail in the [Definitions of Reproducibility](#) section of this chapter. While it is absolutely fine for us each to use different words, it will be useful for you to know how *The Turing Way* defines *reproducibility* to avoid misunderstandings when reading the rest of the handbook.



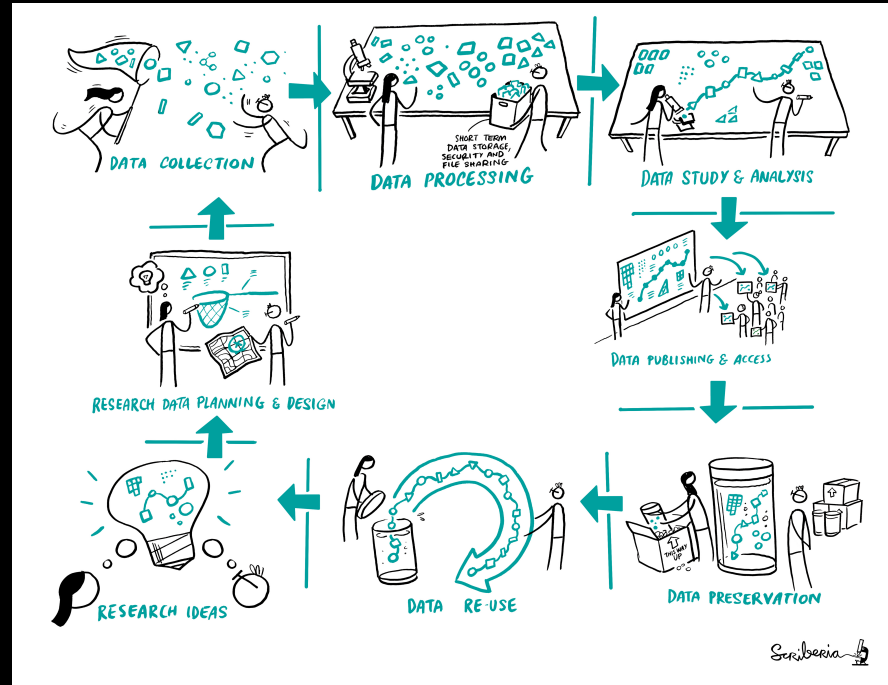
<https://the-turing-way.netlify.app>

@martinagvilas, @turingway #TuringWay

The Turing Way chapters on reproducibility

Guide for Reproducible Research

- Overview
- Open Research
- Version Control
- Licensing
- Research Data Management
- Reproducible Environments
- BinderHub
- Code quality
- Code Testing
- Code Reviewing Process
- Continuous Integration
- Reproducible Research with Make
- Research Compendia
- Credit for Reproducible Research
- Risk Assessment
- Case Studies



take home message

sharing your code and
data isn't enough



computational reproducibility

- Track changes to your code (version control)
- Write clean, understandable and error free code
- Save and share your computational environment
- Make your code open source

computational reproducibility

- **Track changes to your code (version control)**
- Write clean, understandable and error free code
- Save and share your computational environment
- Make your code open source

version control

- records changes to a file or set of files over time
- provides access to any specific version

version control



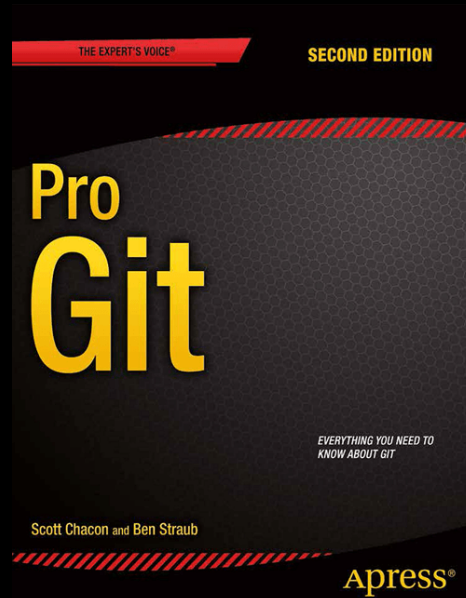
- changes are recorded using **snapshots**
- **distributed** version control system

version control



- Web and Desktop App GUI interface
- most Open Source software hosted here

version control



<https://git-scm.com/>

@martinagvilas, @turingway #TuringWay

computational reproducibility

- Track changes to your code (version control)
- **Write clean, understandable and error free code**
- Save and share your computational environment
- Make your code open source

code style guide

- set of conventions on how to format code
- e.g.
 - ✓ Indentation
 - ✓ Comments
 - ✓ Imports
 - ✓ Naming

code style guide

PEP 8

e.g. imports

```
# Correct:  
import os  
import sys
```

```
# Wrong:  
import sys, os
```

code style guide

PEP 8

e.g. spaces

```
# Correct:  
  
i = i + 1  
  
submitted += 1  
  
x = x*2 - 1  
  
hypot2 = x*x + y*y  
  
c = (a+b) * (a-b)
```

```
# Wrong:  
  
i=i+1  
  
submitted +=1  
  
x = x * 2 - 1  
  
hypot2 = x * x + y * y  
  
c = (a + b) * (a - b)
```

code style guide

Style Guide Enforcement tools

code style guide

Style Guide Enforcement tools → **flake8**

```
!flake8 example_files/bad_file.py
```

```
example_files/bad_file.py:1:80: E501 line too long (80 > 79 characters)  
example_files/bad_file.py:3:2: E225 missing whitespace around operator
```

code style guide

Automatic formatting tools

<https://github.com/psf/black>

@martinagvilas, @turingway #TuringWay

code style guide

Automatic formatting tools → **Black**

```
# in:  
  
j = [1,  
     2,  
     3  
]  
  
# out:  
  
j = [1, 2, 3]
```

code review

- have another programmer look over your code and assess it

code review

types

→ synchronous

- lab meetings presentations

→ asynchronous

- GitHub

code testing

“You should not skip writing tests because you are short on time, you should write tests because you are short on time”

code testing

“You should not skip writing tests because you are short on time, you should write tests because you are short on time”

- you probably are already doing it ...

```
data = np.random.randint(0, 10, size=(4, 6))  
print(data)
```

```
[[3 4 4 3 3 8]  
 [5 7 5 6 8 8]  
 [0 0 8 2 9 8]  
 [4 4 1 8 7 4]]
```

assert

```
expected_n_rows = 3  
assert data.shape[0] == expected_n_rows, "shape mismatch"
```

```
-----  
AssertionError                                Traceback (most recent call last)  
<ipython-input-3-c9f3f4600ddd> in <module>  
      1 expected_n_rows = 3  
----> 2 assert data.shape[0] == expected_n_rows, "shape mismatch"  
  
AssertionError: shape mismatch
```

assert

```
expected_n_rows = 3
assert data.shape[0] == expected_n_rows, "shape mismatch"
```

```
expected_n_rows = 3
real_n_rows = data.shape[0]
assert real_n_rows == expected_n_rows, (
    f"shape mismatch, data has {real_n_rows} rows, expected {expected_n_rows} rows"
)
```

```
-----
AssertionError                                Traceback (most recent call last)
<ipython-input-22-1d999f81fff0> in <module>
      2 real_n_rows = data.shape[0]
      3 assert real_n_rows == expected_n_rows, (
----> 4     f"shape mismatch, data has {real_n_rows} rows, expected {expected_n_rows} rows"
      5 )
```

```
AssertionError: shape mismatch, data has 4 rows, expected 3 rows
```

unit testing

```
def take_fifth_power(x):  
    result = x * x * x * x * x  
    return result
```



```
def test_take_fifth_power():  
    assert take_fifth_power(1.5) == 7.59375
```



pytest

Testing framework

- detailed info of assert statements

```
$ pytest
===== test session starts =====
platform linux -- Python 3.x.y, pytest-6.x.y, py-1.x.y, pluggy-0.x.y
cachedir: $PYTHON_PREFIX/.pytest_cache
rootdir: $REGENDOC_TMPDIR
collected 1 item

test_sample.py F [100%]

===== FAILURES =====
_____ test_answer _____

    def test_answer():
>       assert inc(3) == 5
E       assert 4 == 5
E       + where 4 = inc(3)

test_sample.py:6: AssertionError
===== short test summary info =====
FAILED test_sample.py::test_answer - assert 4 == 5
===== 1 failed in 0.12s =====
```

pytest

Testing framework

- detailed info of assert statements
- auto-discovery of tests

```
setup.py
mypkg/
  __init__.py
  app.py
  view.py
tests/
  test_app.py
  test_view.py
  ...
```


other types of testing

<https://the-turing-way.netlify.app/reproducible-research/testing/testing-types.html>

@martinagvilas, @turingway #TuringWay

other types of testing

Smoke test

- initial checks designed to ensure very basic functionality

other types of testing

Smoke test

- initial checks designed to ensure very basic functionality

Integration test

- individual units are combined and tested as a group

other types of testing

Smoke test

- initial checks designed to ensure very basic functionality

Integration test

- individual units are combined and tested as a group

etc..!


computational reproducibility

- Track changes to your code (version control)
- Write clean, understandable and error free code
- **Save and share your computational environment**
- Make your code open source

what is a computational environment?

- hardware (e.g. CPU)
- software
 - operating system
 - programming languages
 - packages

what is a computational environment?

- hardware (e.g. CPU)
 - software
 - operating system
 - programming languages
 - packages
- 
- their versions
and their
configuration

what is a computational environment?

- hardware (e.g. CPU)

- software

- operating system

- programming languages

- packages

their versions
and their
configuration

and their
interaction

reproducible computational environments

		Interaction style	
		Graphical	Command line
What is reproduced?	Software and versions	Binder	Conda
	Entire system	Virtual machine	Containers

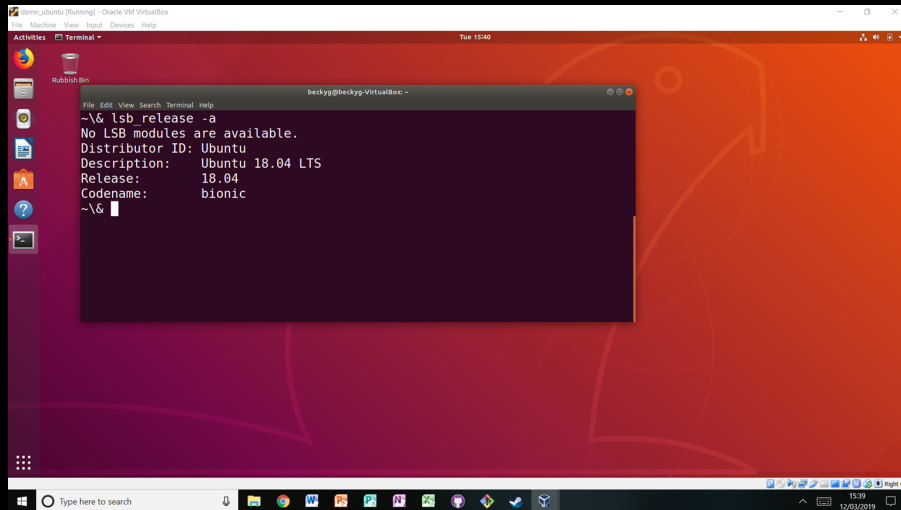
conda



- environment management system
- package and version management system

virtual machines

package a whole
computer as an app that
can be run



<https://the-turing-way.netlify.app/reproducible-research/renv/renv-vm.html>

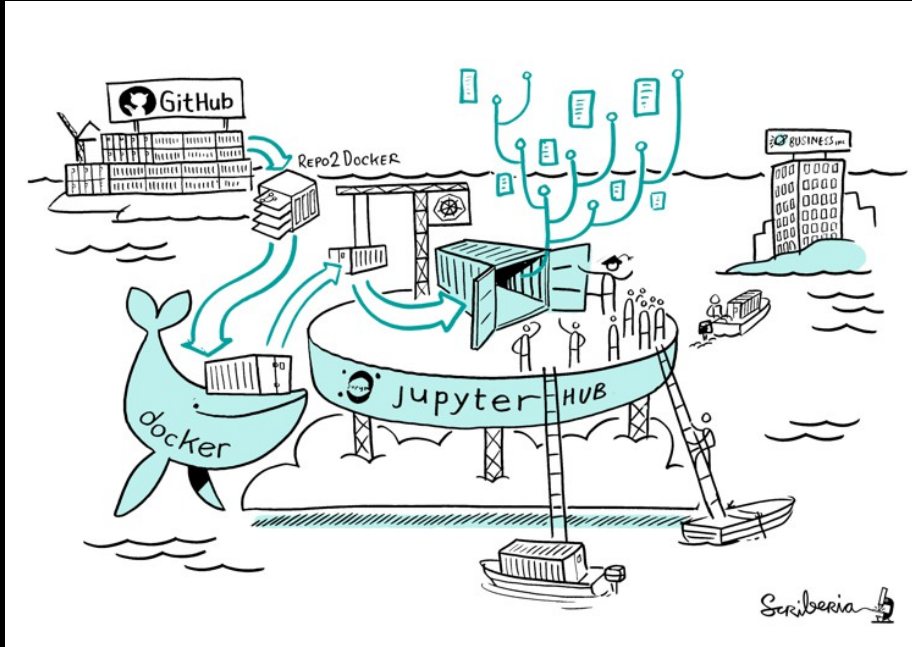
@martinagvilas, @turingway #TuringWay

containers



- behaves like a virtual machine
- more lightweight -> only contains the individual components needed to operate the project

Binder



Turn a Git repo into a collection of interactive notebooks

Have a repository full of Jupyter notebooks? With Binder, open those notebooks in an executable environment, making your code immediately reproducible by anyone, anywhere.

New to Binder? Get started with a [Zero-to-Binder tutorial](#) in [Julia](#), [Python](#) or [R](#).

Build and launch a repository

GitHub repository name or URL


GitHub

Git ref (branch, tag, or commit)

HEAD

Path to a notebook file (optional)

Copy the URL below and share your Binder with others:

Copy the text below, then paste into your README to show a binder badge:  [launch binder](#)

<https://the-turing-way.netlify.app/reproducible-research/renv/renv-binder.html>

@martinagvilas, @turingway #TuringWay

computational reproducibility

- Track changes to your code (version control)
- Write clean, understandable and error free code
- Save and share your computational environment
- **Make your code open source**

open source

- anybody can view, use, modify, and distribute the software for any purpose

licensing

	Copyleft		Permissive	Proprietary
	Strong	Weak		
Use for anything	Yes	Yes	Yes	Sometimes
Private changes	Yes	Yes	Yes	Rarely
Distribute original	Same license, with source	Same license, with source	Same license, also binary-only ¹	Rarely
Distribute modified	Same license, with source	Same license, with source ²	Any license, also binary-only	Rarely
Distribute combined	Same license, with source	Any license, binary additions	Any license, also binary-only	Rarely

¹Under any license for the MIT license ²Relicensing LGPL to GPL is allowed

Free			Proprietary
Copyleft		Permissive	
Strong	Weak		
GPL ¹ CDDL ²	LGPL ³ MPL ⁴	BSD ⁵ MIT ⁶ Apache	
			Research Only: No copying, No modification

¹GPL: GNU General Public License ²CDDL: Common Development and Distribution License ³LGPL: GNU Lesser General Public License ⁴MPL: Mozilla Public License ⁵BSD: Berkley Software Distribution ⁶MIT: Massachusetts Institute of Technology

<https://the-turing-way.netlify.app/reproducible-research/licensing/licensing-software.html>

@martinagvilas, @turingway #TuringWay

community files

The screenshot shows the GitHub interface for the file 'README.md' in the repository 'the-turing-way'. At the top, it indicates the file is on the 'master' branch. Below this, a commit message 'allcontributors docs: update README.md' is shown, along with the commit hash 'baed753' and the time '17 hours ago'. A row of 16 contributor avatars is displayed. The file statistics show '386 lines (346 sloc)' and '82.2 KB'. The file content starts with the title 'The Turing Way' followed by a series of links and buttons: 'read the book', 'receive our newsletter', 'chat on github', 'DOI 10.5281/zenodo.3233853', and a button 'TuringWay I want to contribute!'. Below these, there is a link to 'all contributors' with a count of '208'. The main text of the README describes the project as a guide to reproducible data science and provides a link to the project's website.

README.md

- project name and main features
- installation instructions
- how to run associated tests
- list of authors/contributors
- contact information
- links to related material

<https://the-turing-way.netlify.app/reproducible-research/open/open-source.html>

@martinagvilas, @turingway #TuringWay

community files

CONTRIBUTING.md

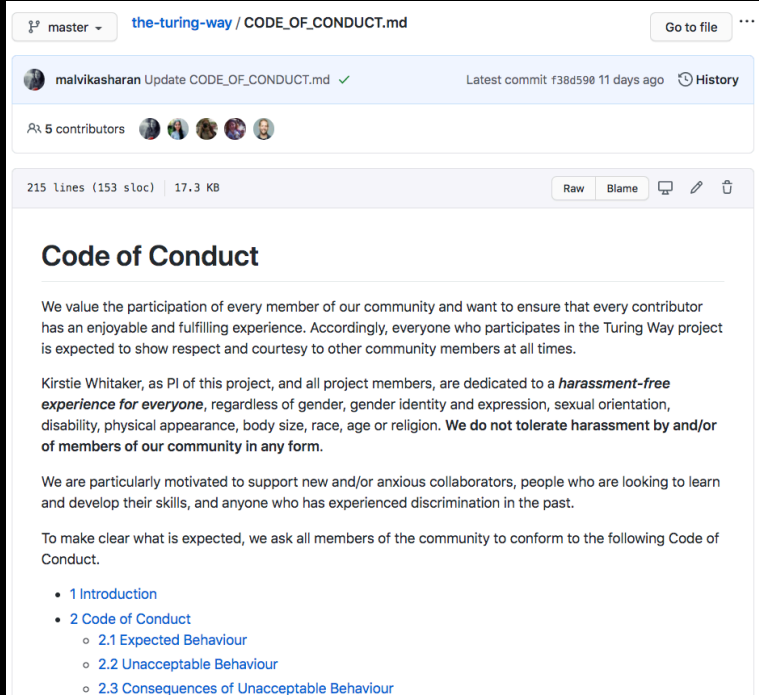
- how to file a bug report
- how to suggest a new feature
- how to contribute changes
- roadmap or vision for the project
- how contributors should (or should not) get in touch with you



<https://the-turing-way.netlify.app/reproducible-research/open/open-source.html>

@martinagvilas, @turingway #TuringWay

community files



CODE_OF_CONDUCT.md

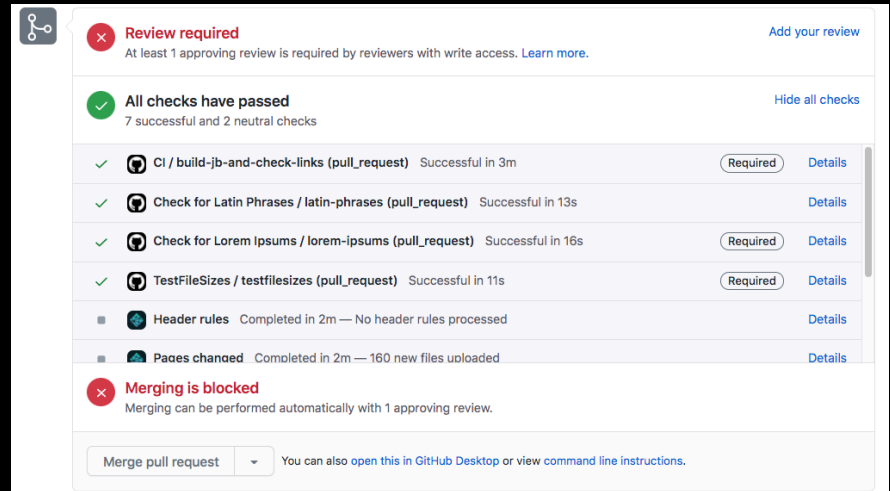
- how you expect participants to behave
- who these expectations apply to
- when they apply
- what to do if a violation occurs

<https://the-turing-way.netlify.app/reproducible-research/open/open-source.html>

@martinagvilas, @turingway #TuringWay

continuous integration

- practice of integrating changes to a project made by individuals into a main, shared version
- frequently



continuous integration

- Travis
- CircleCi
- GitHub Actions
- Azure

.yaml

```
language: python
python:
  - "2.7"

before_install:
  - pip install coverage

script:
  - pytest

after_success:
  - coverage run main.py
  - coverage report
```

```
language: python
python:
  - "2.6"
  - "2.7"
  - "3.2"
  - "3.3"
```

```
os:
  - linux
  - osx
```

etc.

- documentation
- packaging
- interactivity

Acknowledgements:

- Kirstie Whitaker ([@kirstie_j](https://twitter.com/kirstie_j)), Project Lead
- Malvika Sharan ([@malvikasharan](https://twitter.com/malvikasharan)), Community Manager
- *The Turing Way* community, friends & collaborators

Useful links:

- Book: the-turing-way.netlify.com
- Twitter: twitter.com/turingway
- Newsletter: tinyletter.com/TuringWay
- GitHub: github.com/alan-turing-institute/the-turing-way
- Slack: <https://tinyurl.com/jointuringwayslack>
- Artwork by Scriberia: <https://doi.org/10.5281/zenodo.3332808>

**The
Alan Turing
Institute**

