

#OpenScience

#OpenSource

Good practice in scientific programming

Robert Haase

Accessible

Reusable

Sustainable

April 2023

Document what you use

- Installation instructions enable reproducible science
- Not necessary as detailed as a blog-post

Getting started with Mambaforge and Python

Mara Lampert, January 26th, 2023

Introduction to Python and Mambaforge

This blog post explains what Python and Mamba/ Mambaforge, and how to install and setup it on your computer. We will also go through some steps how to use it for Bio-image Analysis.

Note: This is an update of a [previous Blogpost](#) written by Johannes.

Why do we need Mamba to use Python?

Creating a new environment

You can create a new environment typing the following command into the Command Prompt:

```
mamba create -n my_first_env devbio-napari python=3.9 -c conda-forge
```

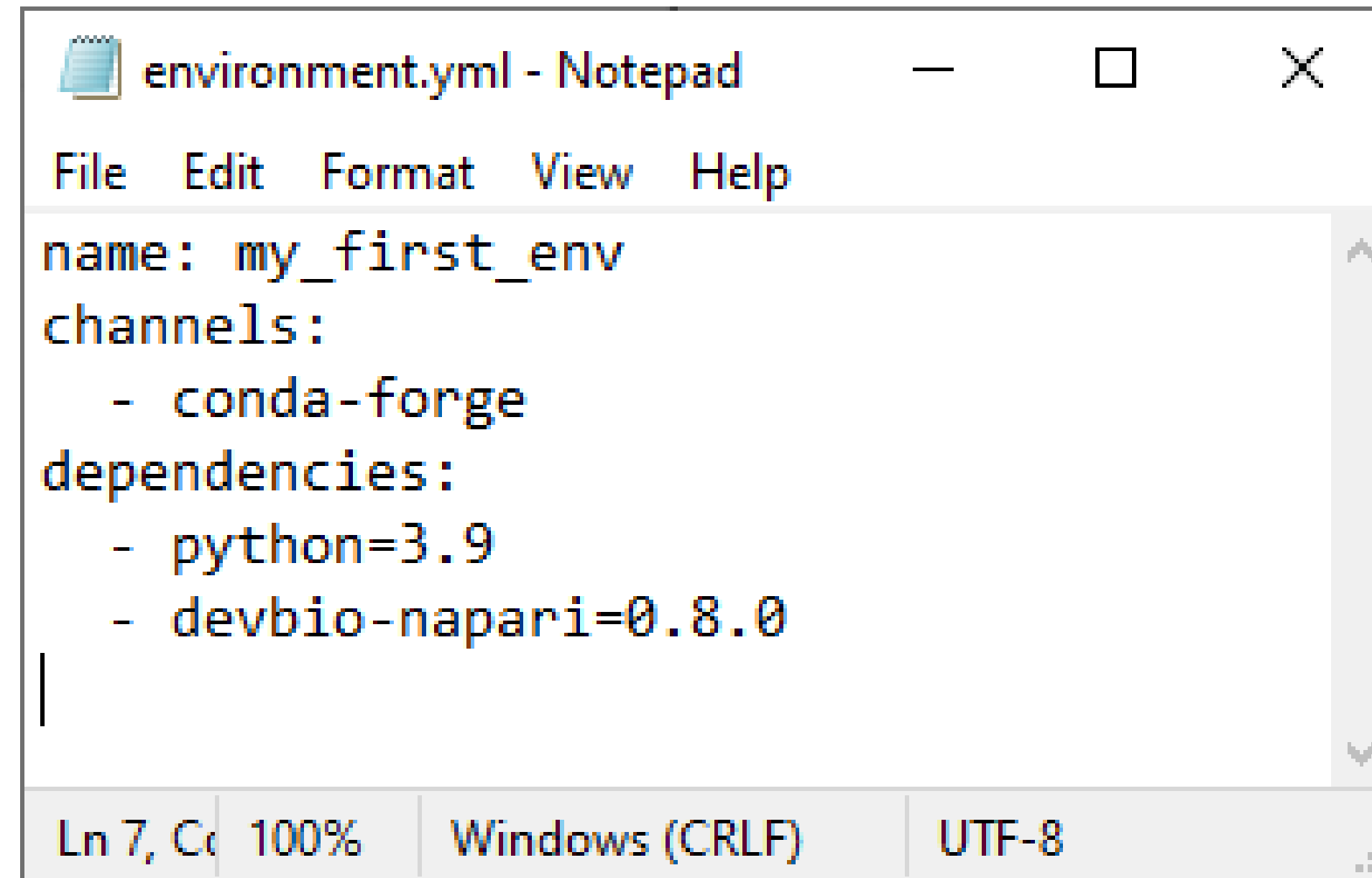
This will create a new environment with the name `my_first_env` and with Python 3.9 installed. Furthermore, the latest version of devbio-napari will be installed in this environment too. Devbio-napari is a collection of Python libraries and Napari plugins maintained by the BiAPoL team, that are useful for processing fluorescent microscopy image data. Confirm your permission to download the needed packages with `Proceed [y]/n`. Press `Enter` you confirm this and mamba will download and install the necessary packages.

Recommendation: Create one conda environment for every project you are working on. This allows you to keep an overview on the needed packages for the project, maintaining

You should at any time be able to rebuild the environment you're working with.

Document what you use

Maintain a document with the dependencies (and versions) you need in your project!



```
environment.yml - Notepad
File Edit Format View Help
name: my_first_env
channels:
  - conda-forge
dependencies:
  - python=3.9
  - devbio-napari=0.8.0
Ln 7, Col 100% Windows (CRLF) UTF-8
```

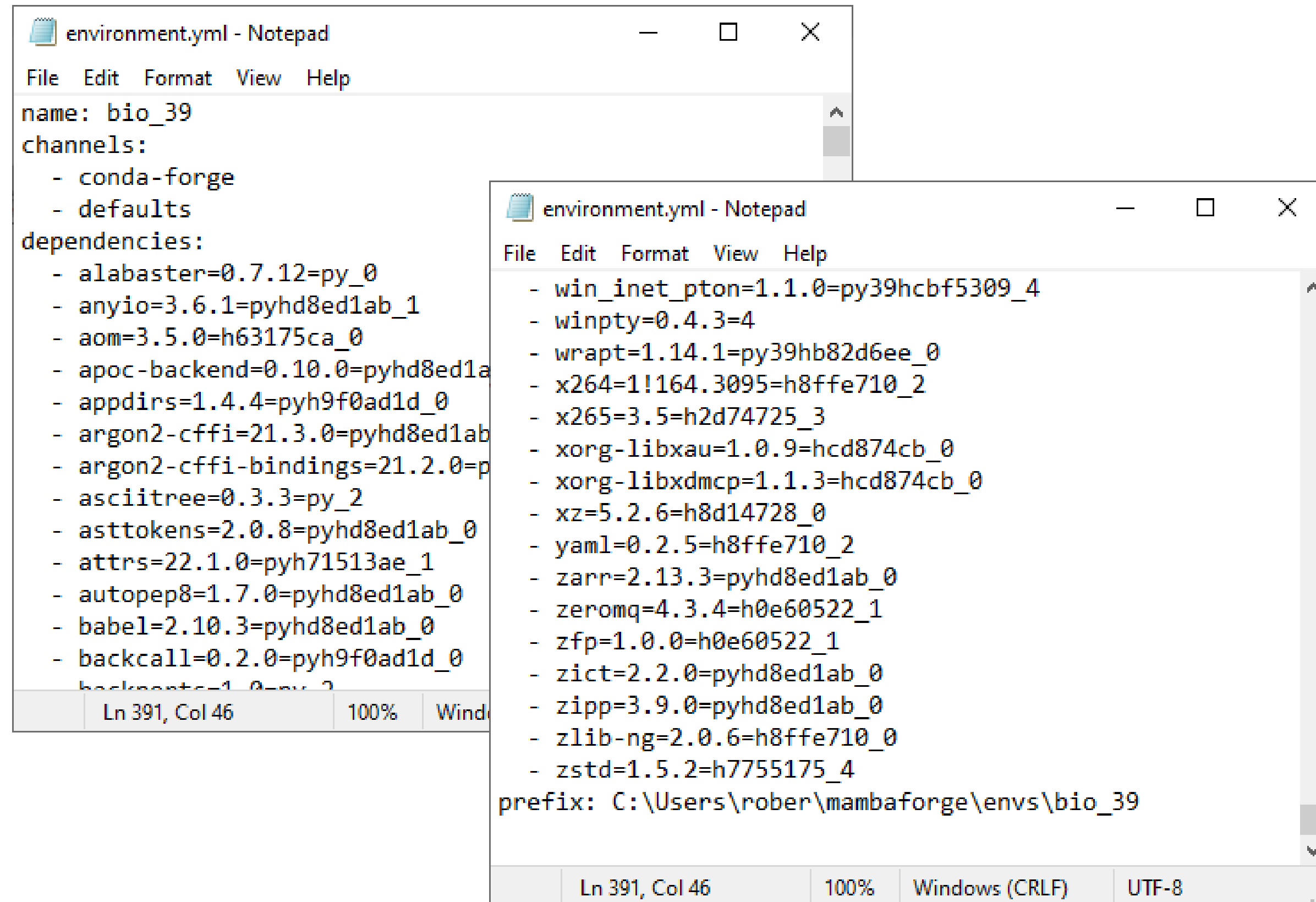
In case your environment is screwed up, you can rebuild it any time.

```
conda env create -f environment.yml
```

Document what you use

... the complete way.

```
conda env export > environment.yml
```



```
environment.yml - Notepad
File Edit Format View Help
name: bio_39
channels:
- conda-forge
- defaults
dependencies:
- alabaster=0.7.12=py_0
- anyio=3.6.1=pyhd8ed1ab_1
- aom=3.5.0=h63175ca_0
- apoc-backend=0.10.0=pyhd8ed1a
- appdirs=1.4.4=pyh9f0ad1d_0
- argon2-cffi=21.3.0=pyhd8ed1ab
- argon2-cffi-bindings=21.2.0=p
- asciitree=0.3.3=py_2
- asttokens=2.0.8=pyhd8ed1ab_0
- attrs=22.1.0=pyh71513ae_1
- autopep8=1.7.0=pyhd8ed1ab_0
- babel=2.10.3=pyhd8ed1ab_0
- backcall=0.2.0=pyh9f0ad1d_0
backports=1.0=py_2

environment.yml - Notepad
File Edit Format View Help
- win_inet_pton=1.1.0=py39hcbf5309_4
- winpty=0.4.3=4
- wrapt=1.14.1=py39hb82d6ee_0
- x264=1!164.3095=h8ffe710_2
- x265=3.5=h2d74725_3
- xorg-libxau=1.0.9=hcd874cb_0
- xorg-libxdmcp=1.1.3=hcd874cb_0
- xz=5.2.6=h8d14728_0
- yaml=0.2.5=h8ffe710_2
- zarr=2.13.3=pyhd8ed1ab_0
- zeromq=4.3.4=h0e60522_1
- zfp=1.0.0=h0e60522_1
- zict=2.2.0=pyhd8ed1ab_0
- zipp=3.9.0=pyhd8ed1ab_0
- zlib-ng=2.0.6=h8ffe710_0
- zstd=1.5.2=h7755175_4
prefix: C:\Users\rober\mambaforge\envs\bio_39
```

Excellent way to document all dependencies were *actually* used...

It is *questionable* if re-creating an environment from this yml file works.

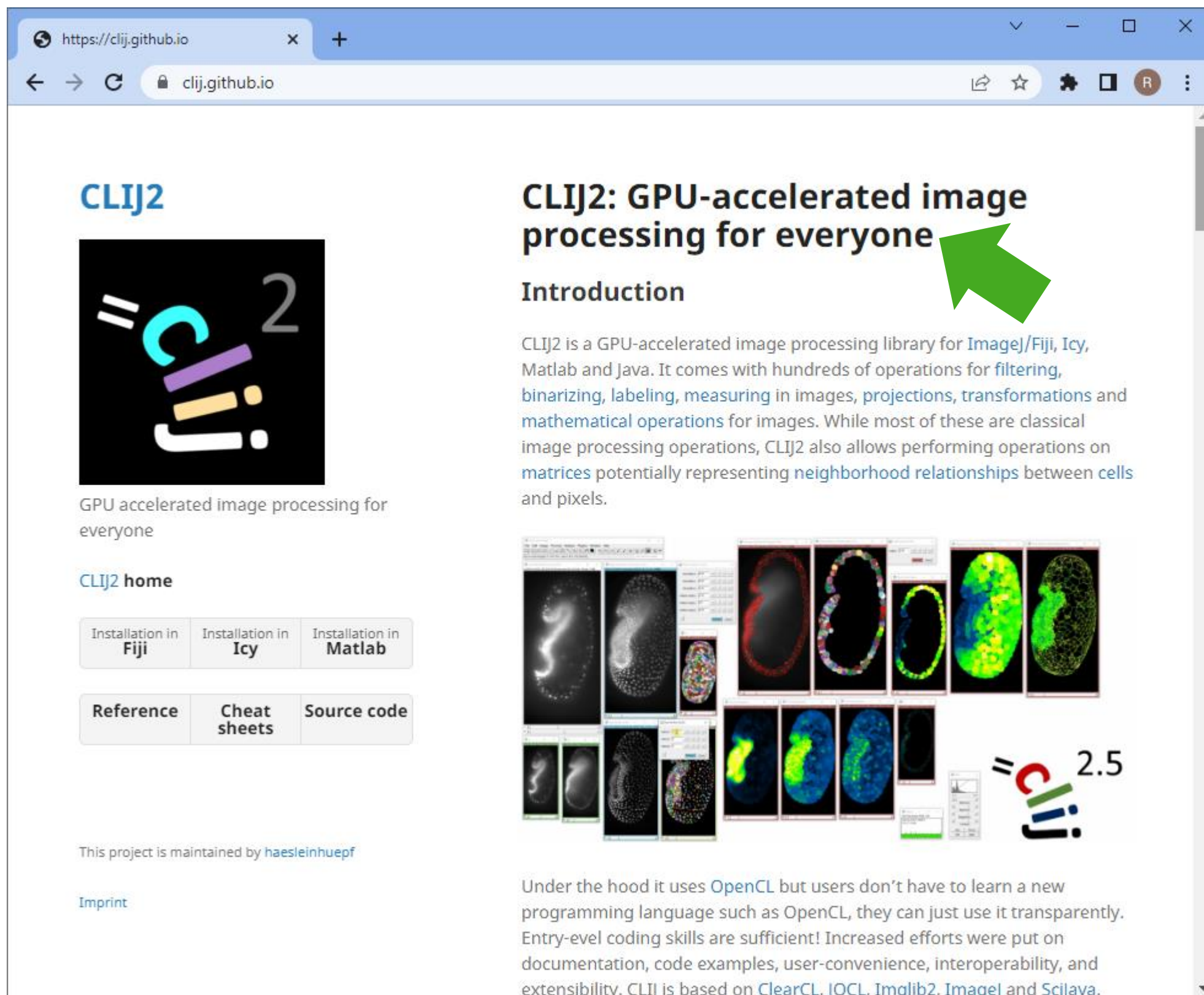
Software quality indicators

... to differentiate the good stuff from the bad.

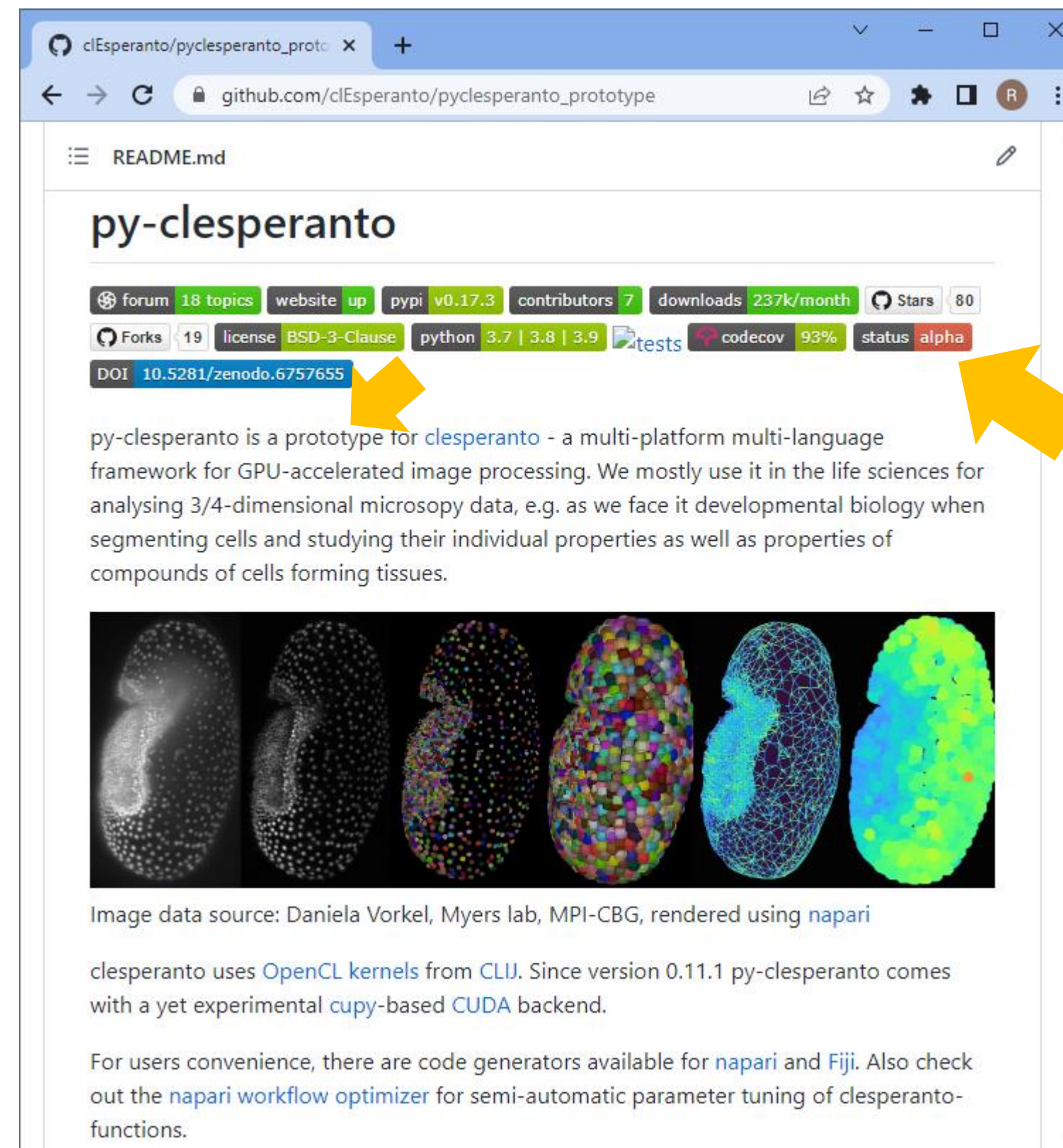
Target audience

- Documentation should tell who is the target audience and how far it is developed

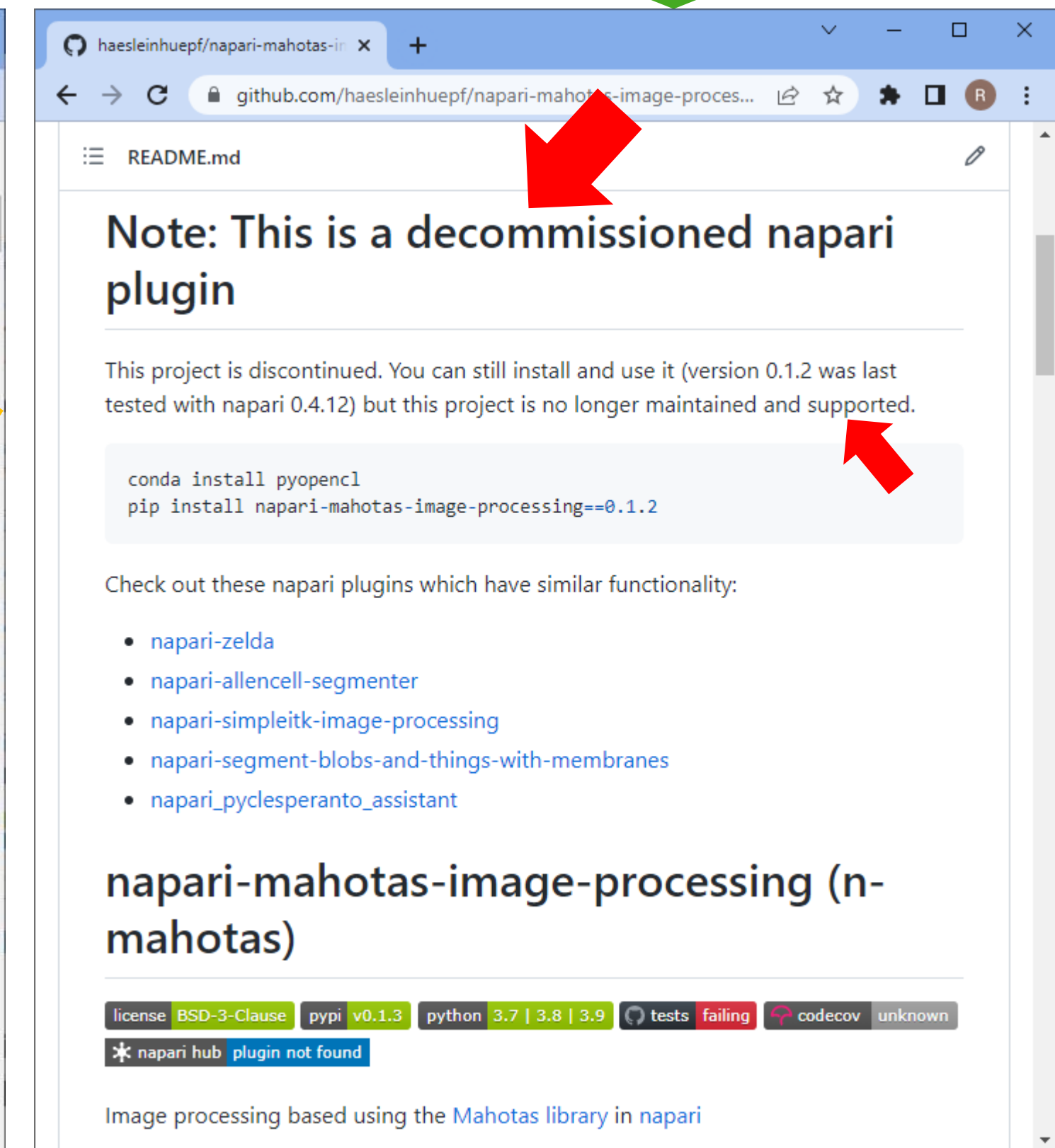
Communication is key!



The screenshot shows the CLIJ2 website. The main heading is "CLIJ2: GPU-accelerated image processing for everyone". A green arrow points to this heading. Below the heading is an introduction paragraph and a grid of image processing results. A green arrow also points to the word "everyone" in the heading. At the bottom, it says "This project is maintained by haesleinhuepf".



The screenshot shows the GitHub repository for py-clesperanto. The main heading is "py-clesperanto". A yellow arrow points to the "status alpha" badge. Another yellow arrow points to the DOI link "10.5281/zenodo.6757655". Below the heading is a paragraph describing the project and a grid of image processing results. At the bottom, it says "For users convenience, there are code generators available for napari and Fiji. Also check out the napari workflow optimizer for semi-automatic parameter tuning of clesperanto-functions."

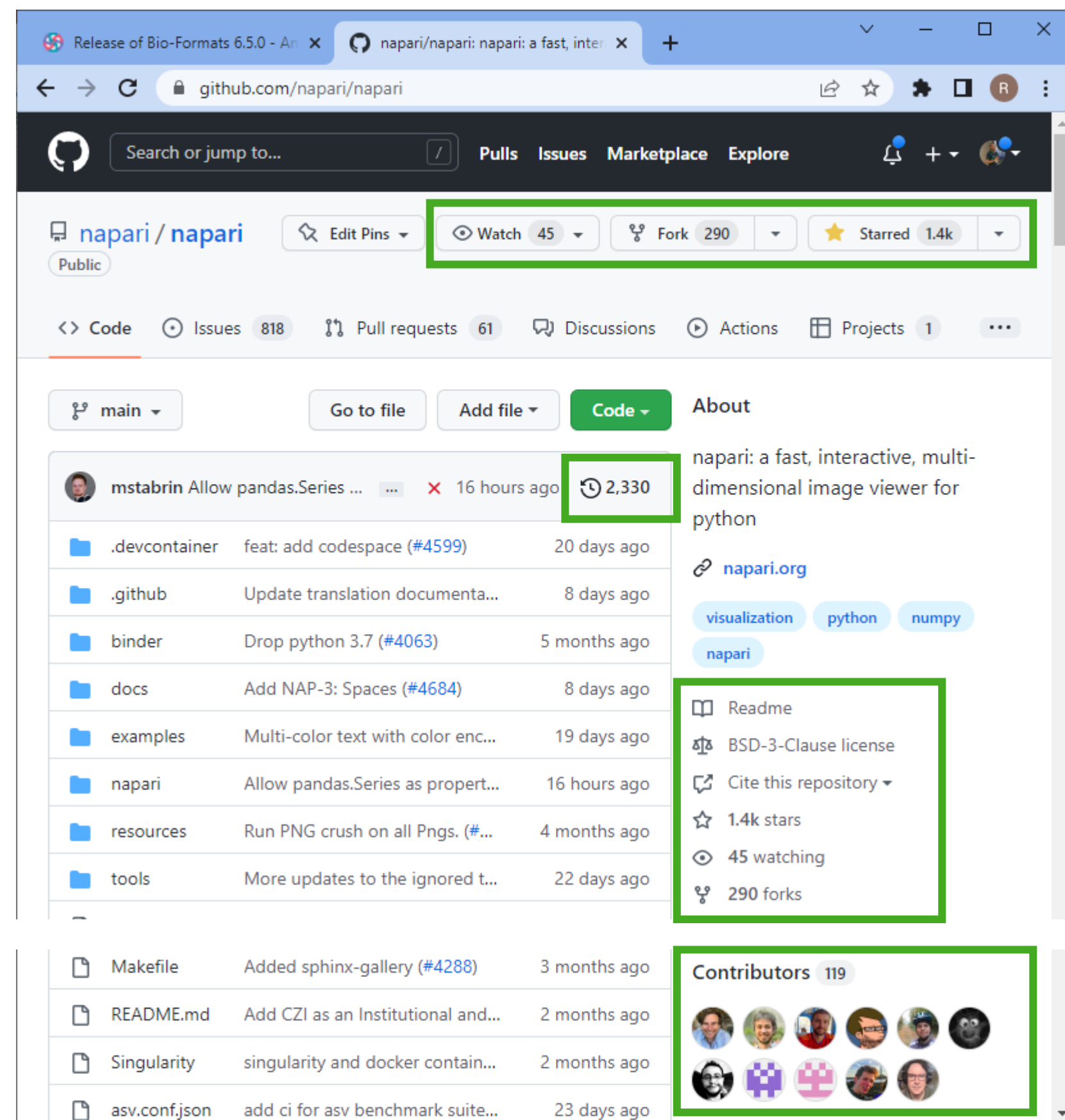


The screenshot shows the GitHub repository for napari-mahotas-image-processing. The main heading is "Note: This is a decommissioned napari plugin". A red arrow points to this heading. Below the heading is a paragraph stating "This project is discontinued. You can still install and use it (version 0.1.2 was last tested with napari 0.4.12) but this project is no longer maintained and supported." A red arrow points to the installation instructions: "conda install pyopenc1" and "pip install napari-mahotas-image-processing==0.1.2". Below this is a list of similar plugins and the repository name "napari-mahotas-image-processing (n-mahotas)". A red arrow points to the "plugin not found" badge.

Software quality indicators

Visit the project's github or gitlab page and review indicators.

- **Stars:** People like software, similarly to tweets on Twitter
- **Watching:** People receive updates for new releases
- **Forks:** People made a copy of the code, e.g. to contribute to the project
- **Contributors:** People who contributed to the code
- **Commits:** Changes to the code



Release of Bio-Formats 6.5.0 - An x napari/napari: napari: a fast, inter x +

github.com/napari/napari

Search or jump to... Pulls Issues Marketplace Explore

napari / napari Edit Pins Watch 45 Fork 290 Starred 1.4k

Code Issues 818 Pull requests 61 Discussions Actions Projects 1

main Go to file Add file Code About

mstabrin Allow pandas.Series ... 16 hours ago 2,330

napari: a fast, interactive, multi-dimensional image viewer for python

napari.org

visualization python numpy napari

Readme

BSD-3-Clause license

Cite this repository

1.4k stars

45 watching

290 forks

Contributors 119

Makefile Added sphinx-gallery (#4288) 3 months ago

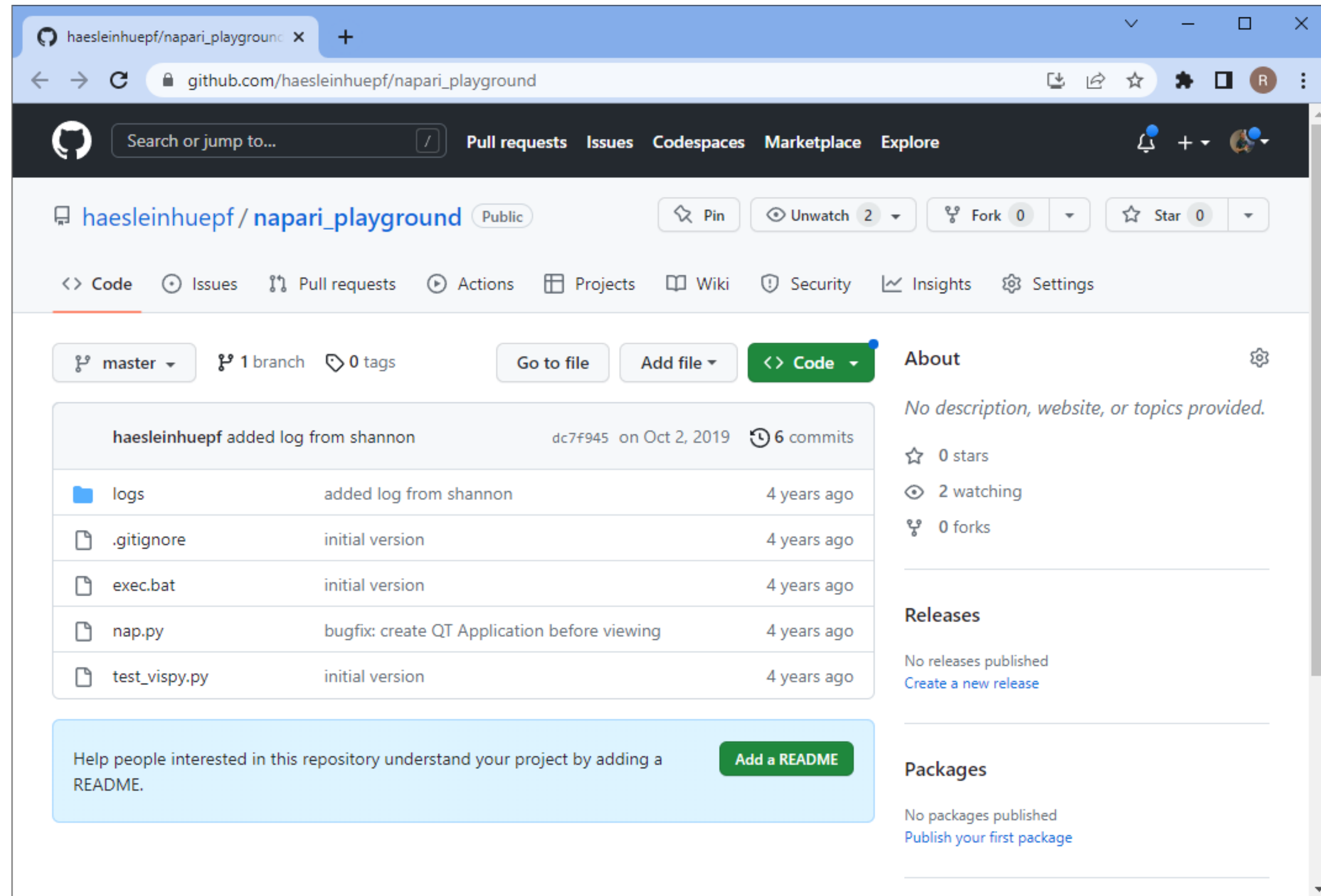
README.md Add CZI as an Institutional and... 2 months ago

Singularity singularity and docker contain... 2 months ago

asv.conf.json add ci for asv benchmark suite... 23 days ago

Bad example

- No readme / documentation
- No license / copyright statement
- No stars / users (?)
- Not maintained (last update 4 years ago)
- bus factor = 1



haesleinhuepf / napari_playground Public

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags

File	Commit Message	Commit Hash	Date	Commits
logs	added log from shannon	dc7f945	on Oct 2, 2019	6 commits
.gitignore	initial version		4 years ago	
exec.bat	initial version		4 years ago	
nap.py	bugfix: create QT Application before viewing		4 years ago	
test_vispy.py	initial version		4 years ago	

Help people interested in this repository understand your project by adding a README. [Add a README](#)

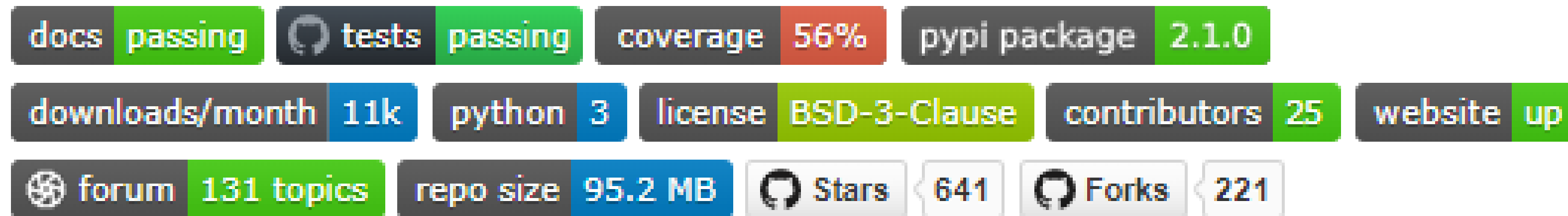
About: No description, website, or topics provided. 0 stars, 2 watching, 0 forks.

Releases: No releases published. [Create a new release](#)

Packages: No packages published. [Publish your first package](#)

Software quality indicators

Visit the project's github or gitlab page and review indicators.

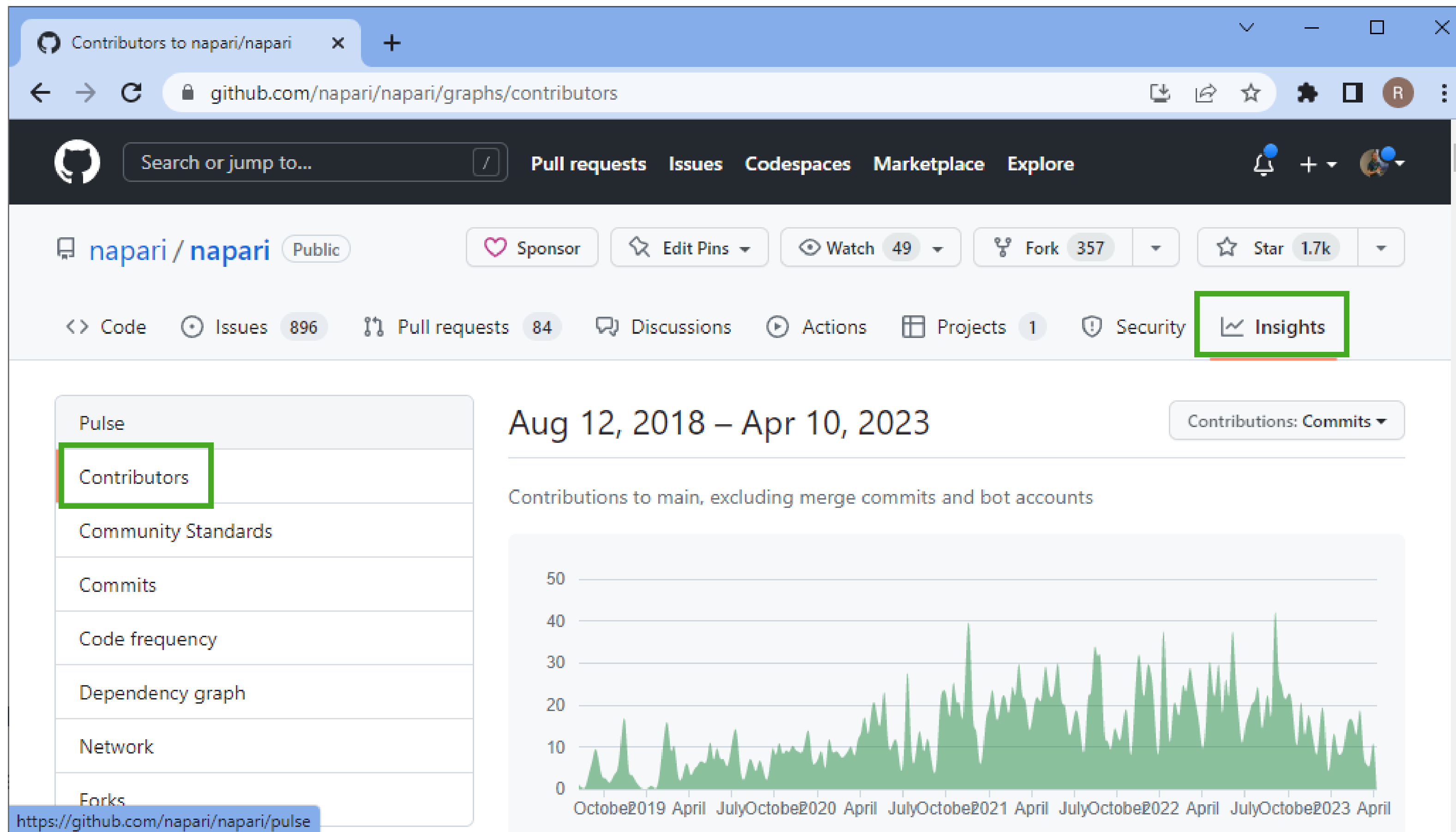


Note, github badges cannot be *deserved*. Developers put them there



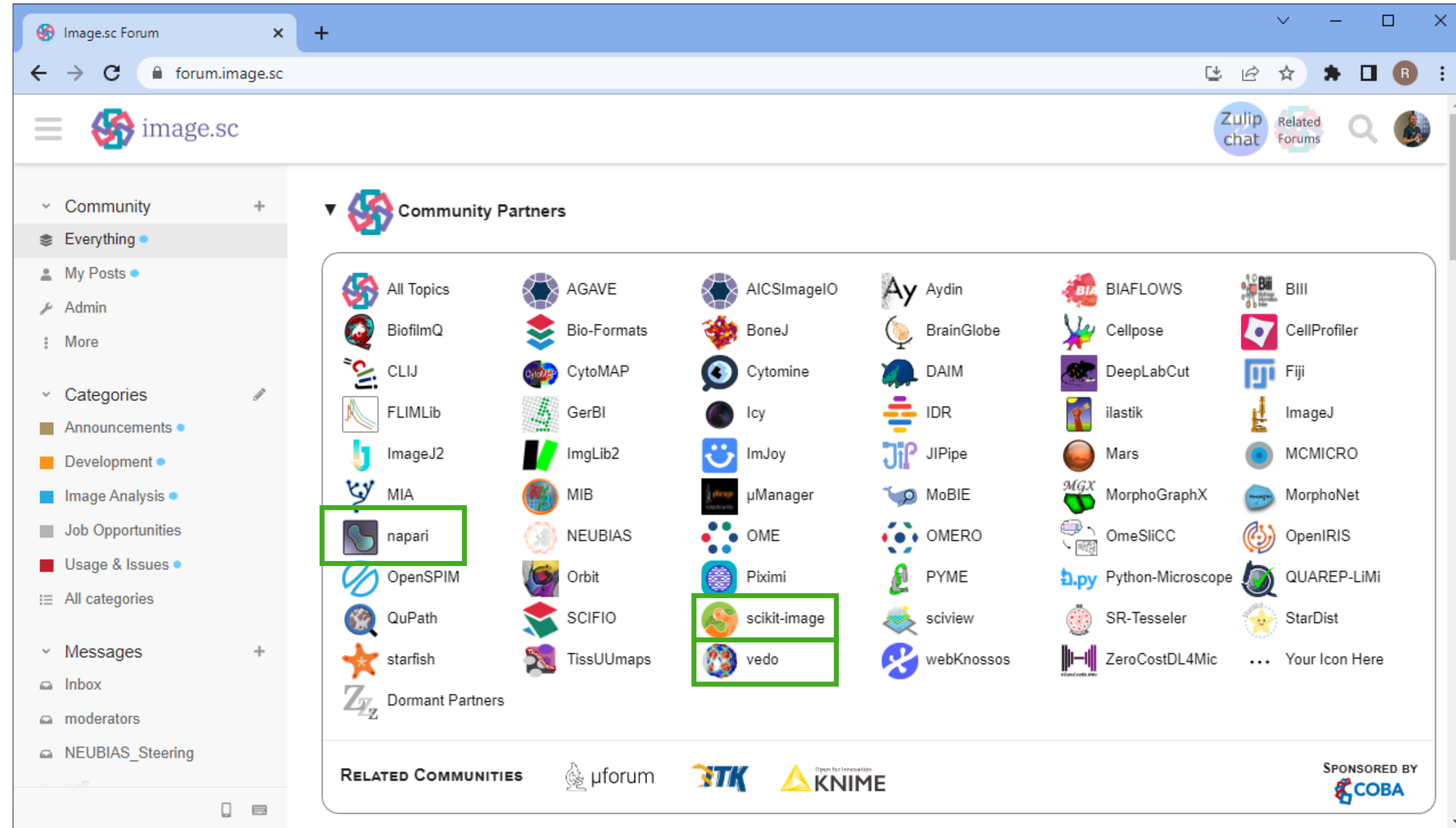
Software quality indicators

Visit the project's github or gitlab page and review indicators.



Software quality indicators

- Community actively involved

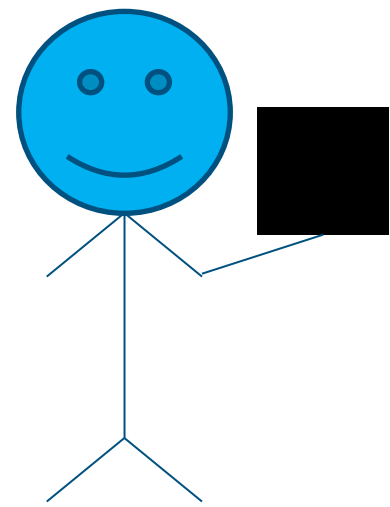


The screenshot shows the Image.sc Forum website. The main content area is titled "Community Partners" and displays a grid of logos for various software tools and organizations. The logos are arranged in a grid with 5 columns and 6 rows. The first row includes logos for "All Topics", "AGAVE", "AICSImageIO", "Ay Aydin", "BIAFLOWS", and "Bill". The second row includes "BiofilmQ", "Bio-Formats", "BoneJ", "BrainGlobe", "Cellpose", and "CellProfiler". The third row includes "CLIJ", "CytoMAP", "Cytomine", "DAIM", "DeepLabCut", and "Fiji". The fourth row includes "FLIMLib", "GerBI", "Icy", "IDR", "ilastik", and "ImageJ". The fifth row includes "ImageJ2", "ImgLib2", "ImJoy", "JIPipe", "Mars", and "MCMICRO". The sixth row includes "MIA", "MIB", "µManager", "MoBIE", "MorphoGraphX", and "MorphoNet". The seventh row includes "OpenSPIM", "Orbit", "Piximi", "OMERO", "OmeSliCC", and "OpenIRIS". The eighth row includes "QuPath", "SCIFIO", "scikit-image", "PYME", "Python-Microscope", and "QUAREP-LIMi". The ninth row includes "starfish", "TissUUmaps", "vedo", "sciview", "SR-Tesseler", and "StarDist". The tenth row includes "Dormant Partners", "webKnossos", "ZeroCostDL4Mic", and "Your Icon Here". The "napari" logo is highlighted with a green box, and the "scikit-image" and "vedo" logos are also highlighted with green boxes. At the bottom of the page, there are logos for "RELATED COMMUNITIES" including "µforum", "STK", and "KNIME", and a "SPONSORED BY COBA" logo.

Openness of software / projects

Choose your project's level wisely, and communicate it clearly

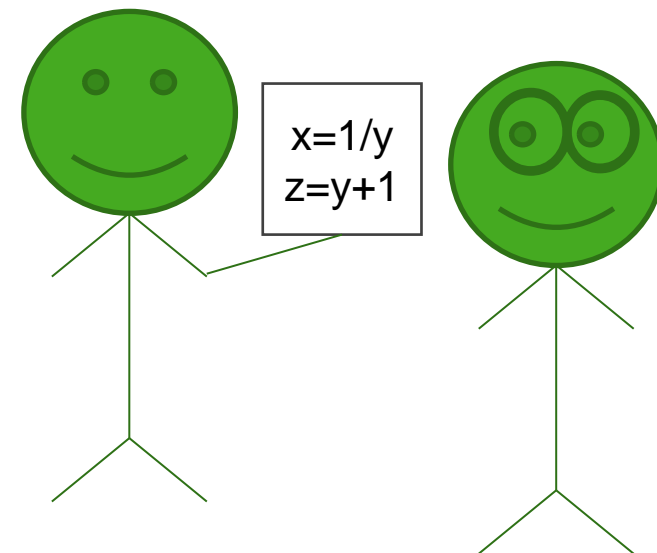
Closed source



- Open to collaborations
- “Black box”
- Compiled code (e.g. C/C++)
- Good for protecting intellectual properties (\$\$\$)

Hardware device drivers

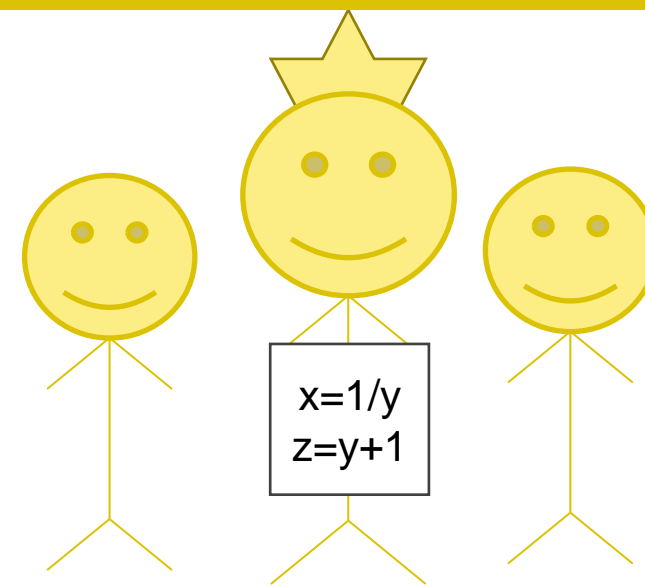
Open source



- Code available to read
- Not necessarily executable code
- No maintenance / support efforts

Custom image analysis scripts

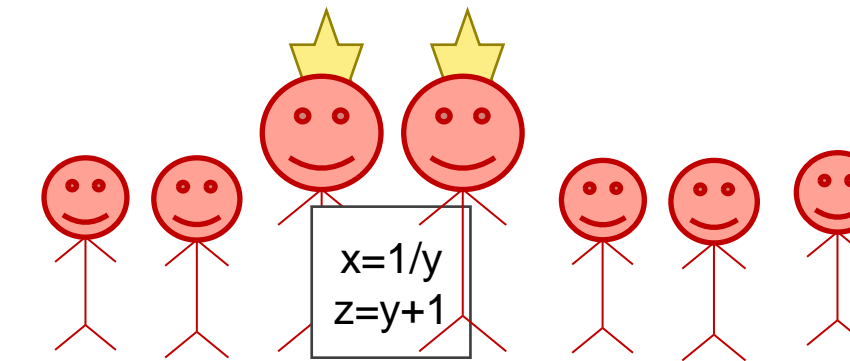
Benevolent dictatorship



- Open to contributions
- Single maintainer, often overwhelmed
- Efficient decision making
- Bus factor ≈ 1

TrackMate, SNT, MorpholibJ, CLIJ

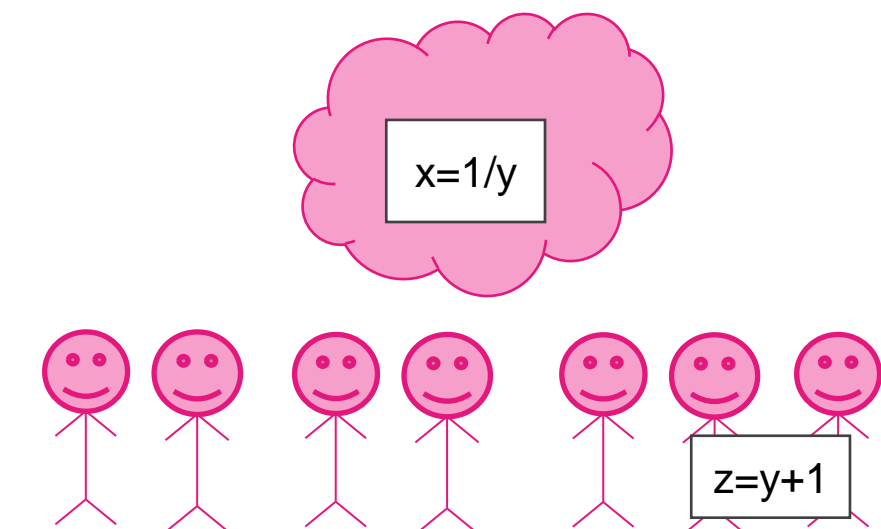
Community driven



- Open to contributions
- Partially democratic
- Board of maintainers (core developers)
- Long-winded decision making

scikit-image, scipy, OpenCL

Openly extensible



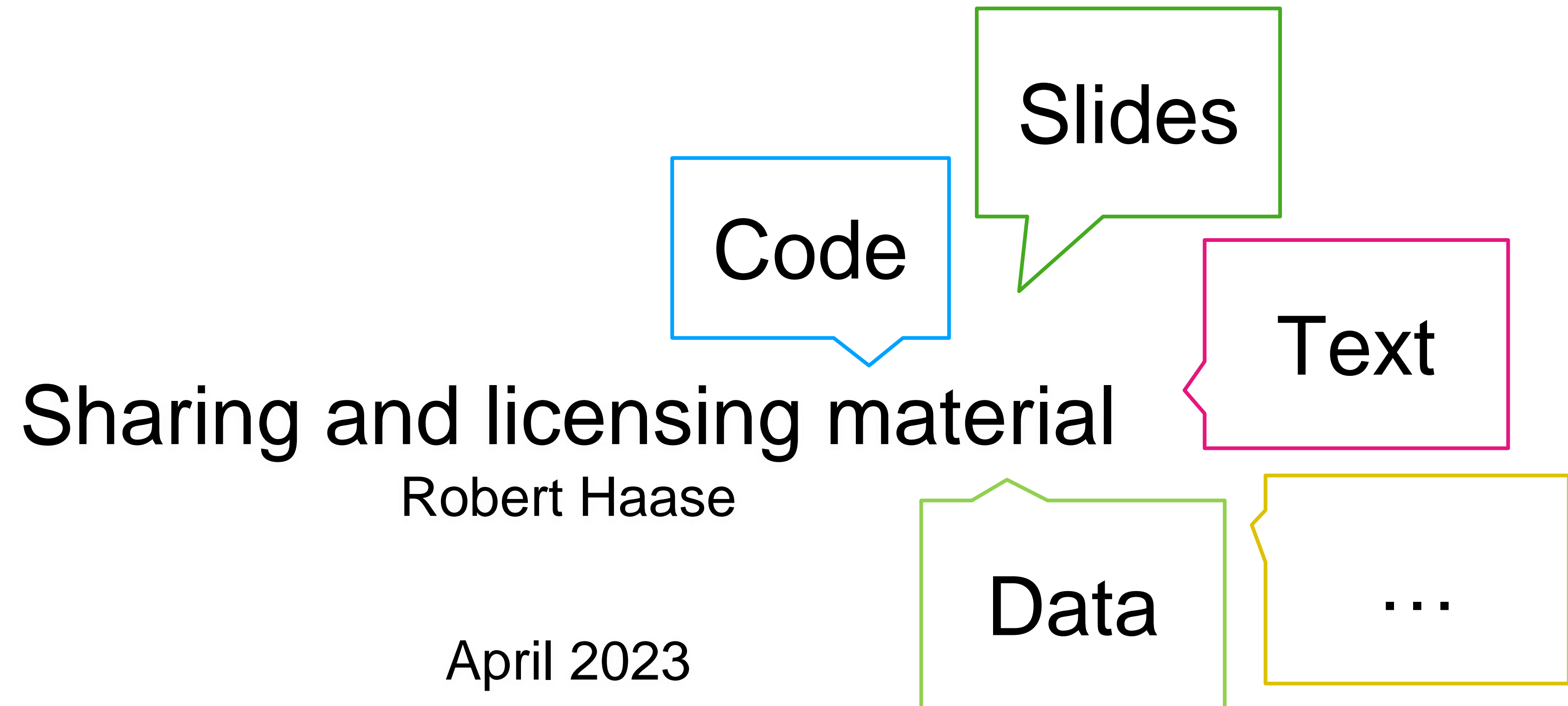
- Openly extensible; without maintainers involved
- Partially community driven

ImageJ, Python, numpy

Take home message

When using [open-source] software, make sure

- it's maintained
- used by others
- supported by an active community
- well-documented



Sharing and licensing material

Robert Haase

April 2023

Code

Slides

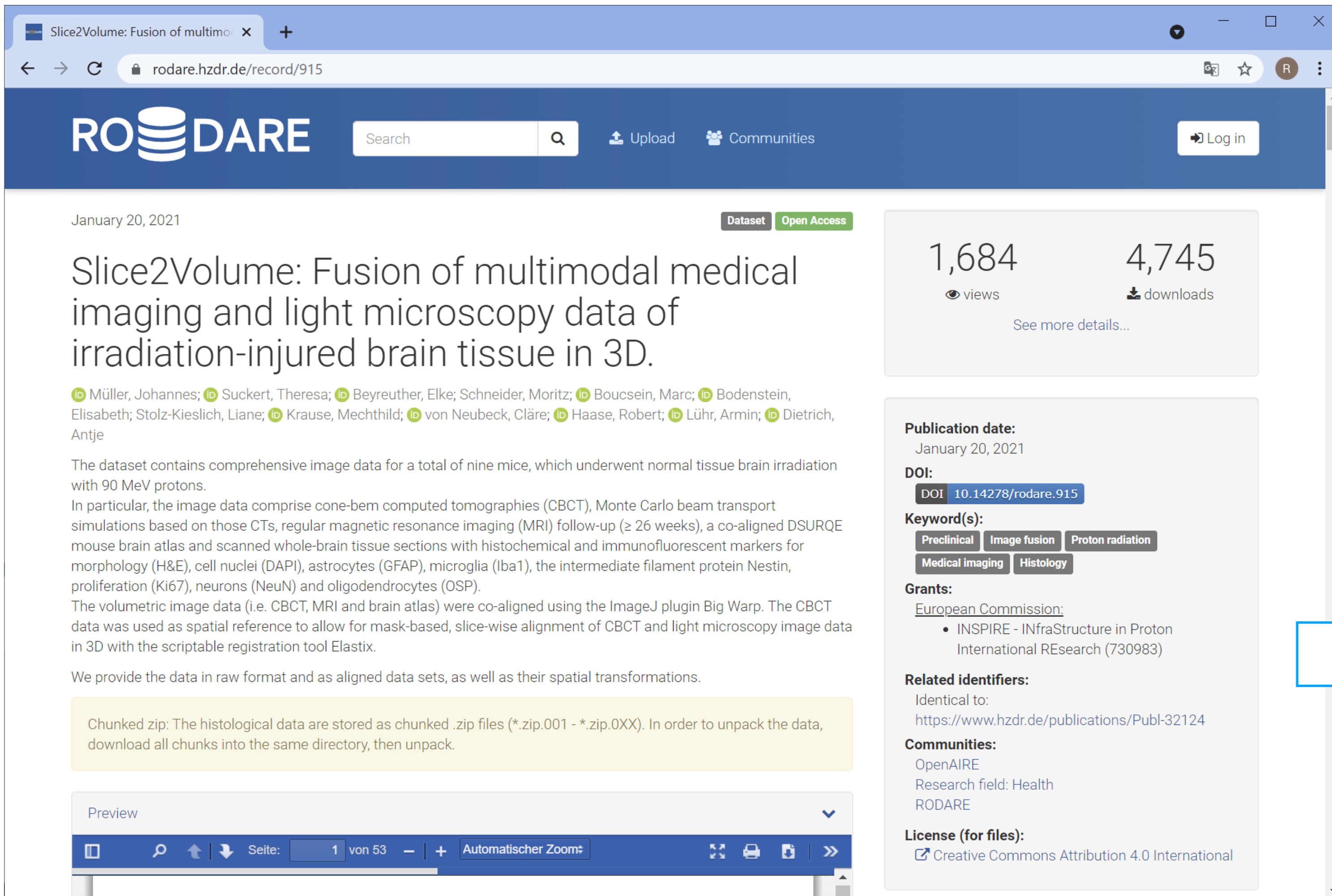
Text

Data

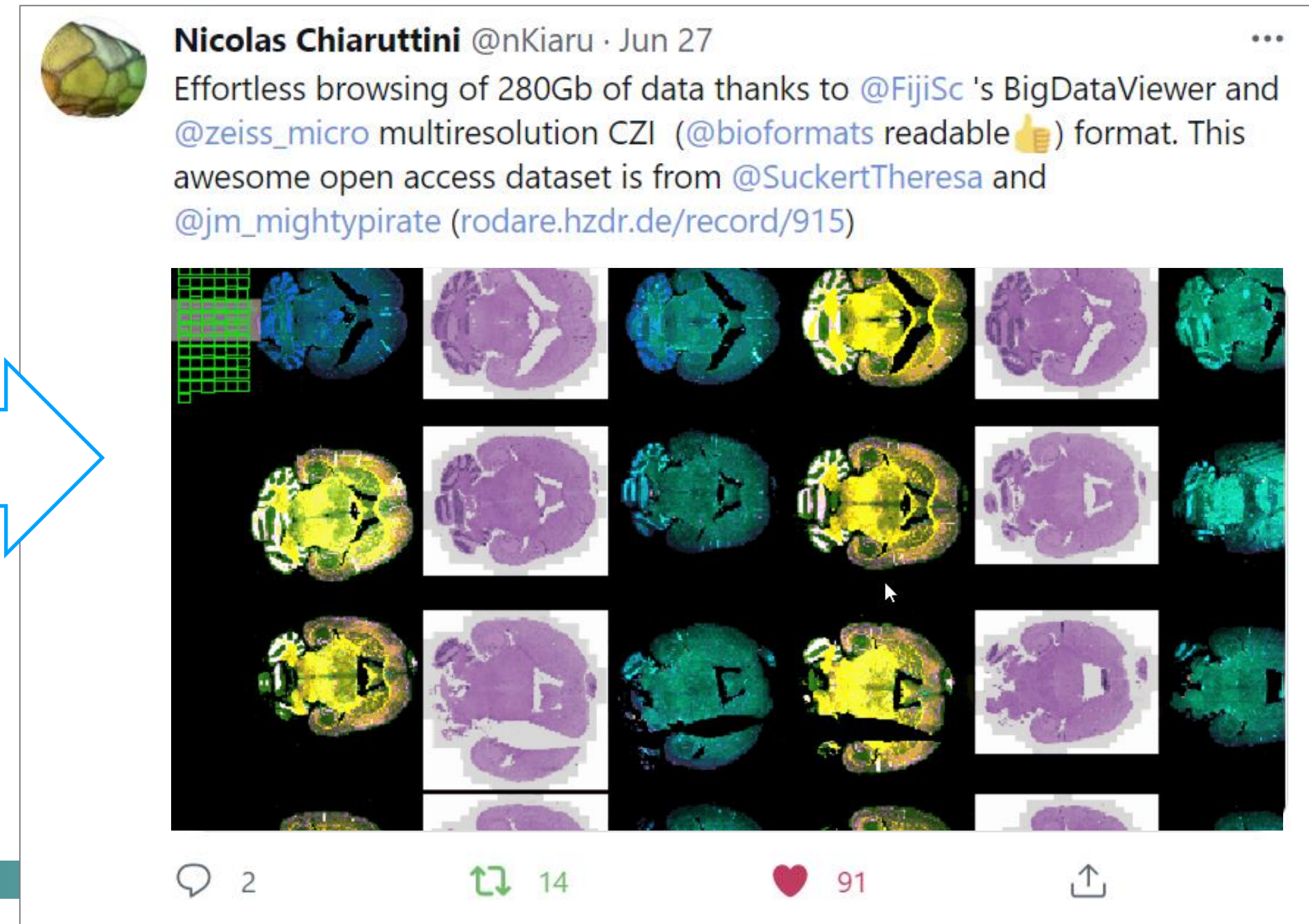
...

Use cases: Data

Unique datasets
Valuable for biologists
Valuable for software developers
Institutional servers / services
<https://idr.openmicroscopy.org/>
<https://zenodo.org>



The screenshot shows the RODARE dataset page. The title is "Slice2Volume: Fusion of multimodal medical imaging and light microscopy data of irradiation-injured brain tissue in 3D." It lists authors: Müller, Johannes; Suckert, Theresa; Beyreuther, Elke; Schneider, Moritz; Boucsein, Marc; Bodenstein, Elisabeth; Stolz-Kieslich, Liane; Krause, Mechthild; von Neubeck, Cläre; Haase, Robert; Lühr, Armin; Dietrich, Antje. The dataset is dated January 20, 2021, and is Open Access. It has 1,684 views and 4,745 downloads. The DOI is 10.14278/rodare.915. Keywords include Preclinical, Image fusion, Proton radiation, Medical imaging, and Histology. The dataset is funded by the European Commission (INSPIRE - INfraStructure in Proton International REsearch). It is available in raw format and as aligned data sets. A preview of the data is shown at the bottom.



The tweet by Nicolas Chiaruttini (@nKiaru) from June 27, 2021, reads: "Effortless browsing of 280Gb of data thanks to @FijiSc's BigDataViewer and @zeiss_micro multiresolution CZI (@bioformats readable 👍) format. This awesome open access dataset is from @SuckertTheresa and @jm_mightypirate (rodare.hzdr.de/record/915)". The tweet includes a grid of 15 brain slice images showing different modalities and a blue arrow pointing from the dataset page to the tweet.

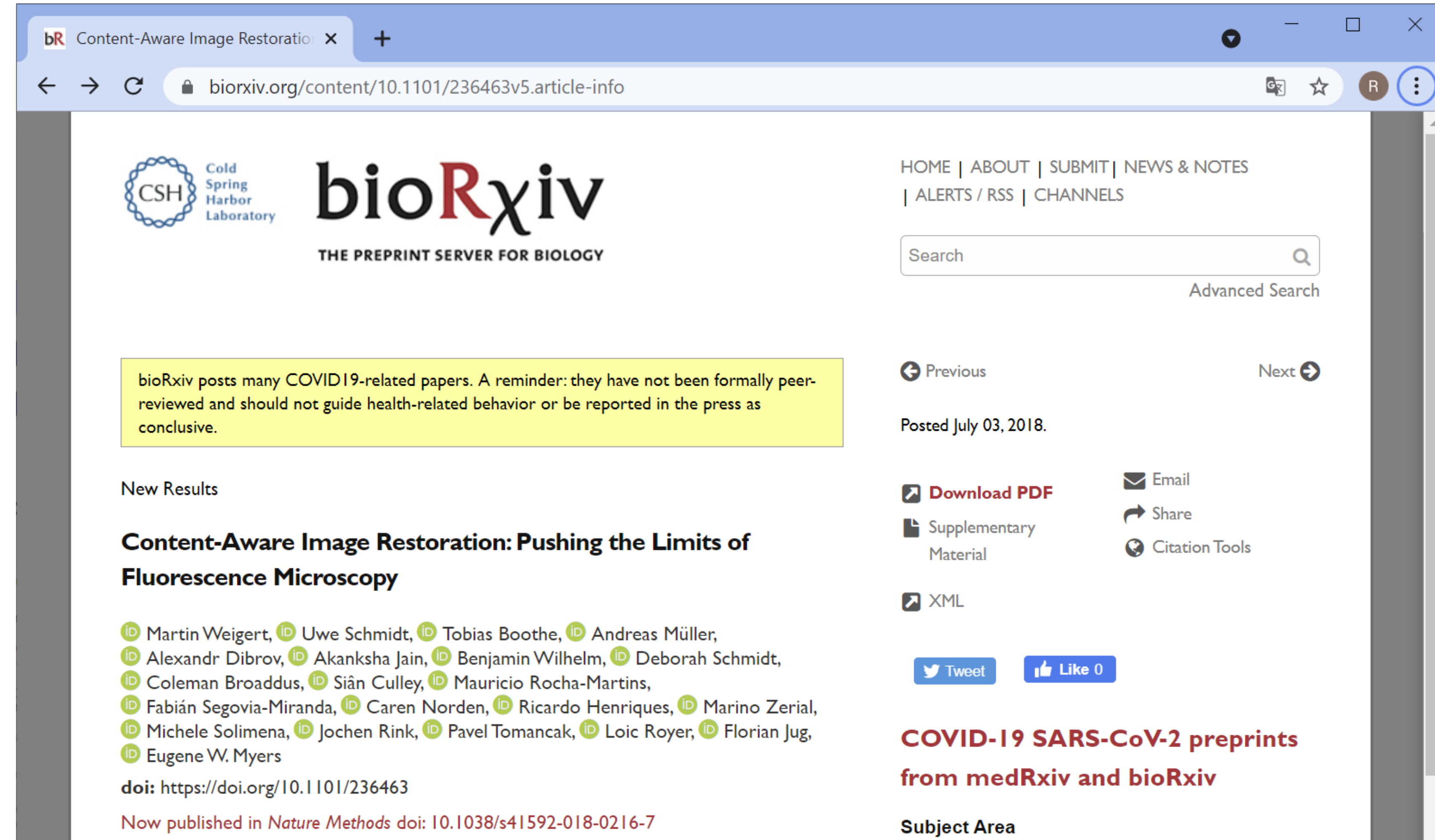
<https://rodare.hzdr.de/record/915>
<https://twitter.com/nKiaru/status/1409194004219142148?s=20>

Use cases: Manuscripts

Preprints

- Accessible / reusable
- <https://arxiv.org/>
- <https://biorxiv.org/>
- <https://medrxiv.org/>

Journals



The screenshot shows a web browser window displaying a bioRxiv article. The browser's address bar shows the URL <https://www.biorxiv.org/content/10.1101/236463v5.article-info>. The page features the bioRxiv logo and the text 'THE PREPRINT SERVER FOR BIOLOGY'. A yellow warning box states: 'bioRxiv posts many COVID19-related papers. A reminder: they have not been formally peer-reviewed and should not guide health-related behavior or be reported in the press as conclusive.' The article title is 'Content-Aware Image Restoration: Pushing the Limits of Fluorescence Microscopy'. The authors listed are Martin Weigert, Uwe Schmidt, Tobias Boothe, Andreas Müller, Alexandr Dibrov, Akanksha Jain, Benjamin Wilhelm, Deborah Schmidt, Coleman Broaddus, Siân Culley, Mauricio Rocha-Martins, Fabián Segovia-Miranda, Caren Norden, Ricardo Henriques, Marino Zerial, Michele Solimena, Jochen Rink, Pavel Tomancak, Loic Royer, Florian Jug, and Eugene W. Myers. The article has a DOI of <https://doi.org/10.1101/236463> and is noted as 'Now published in Nature Methods doi: 10.1038/s41592-018-0216-7'. On the right side, there are navigation links (HOME, ABOUT, SUBMIT, NEWS & NOTES, ALERTS / RSS, CHANNELS), a search bar, and social media sharing options (Download PDF, Supplementary Material, XML, Email, Share, Citation Tools, Tweet, Like 0). A banner at the bottom right promotes 'COVID-19 SARS-CoV-2 preprints from medRxiv and bioRxiv'.

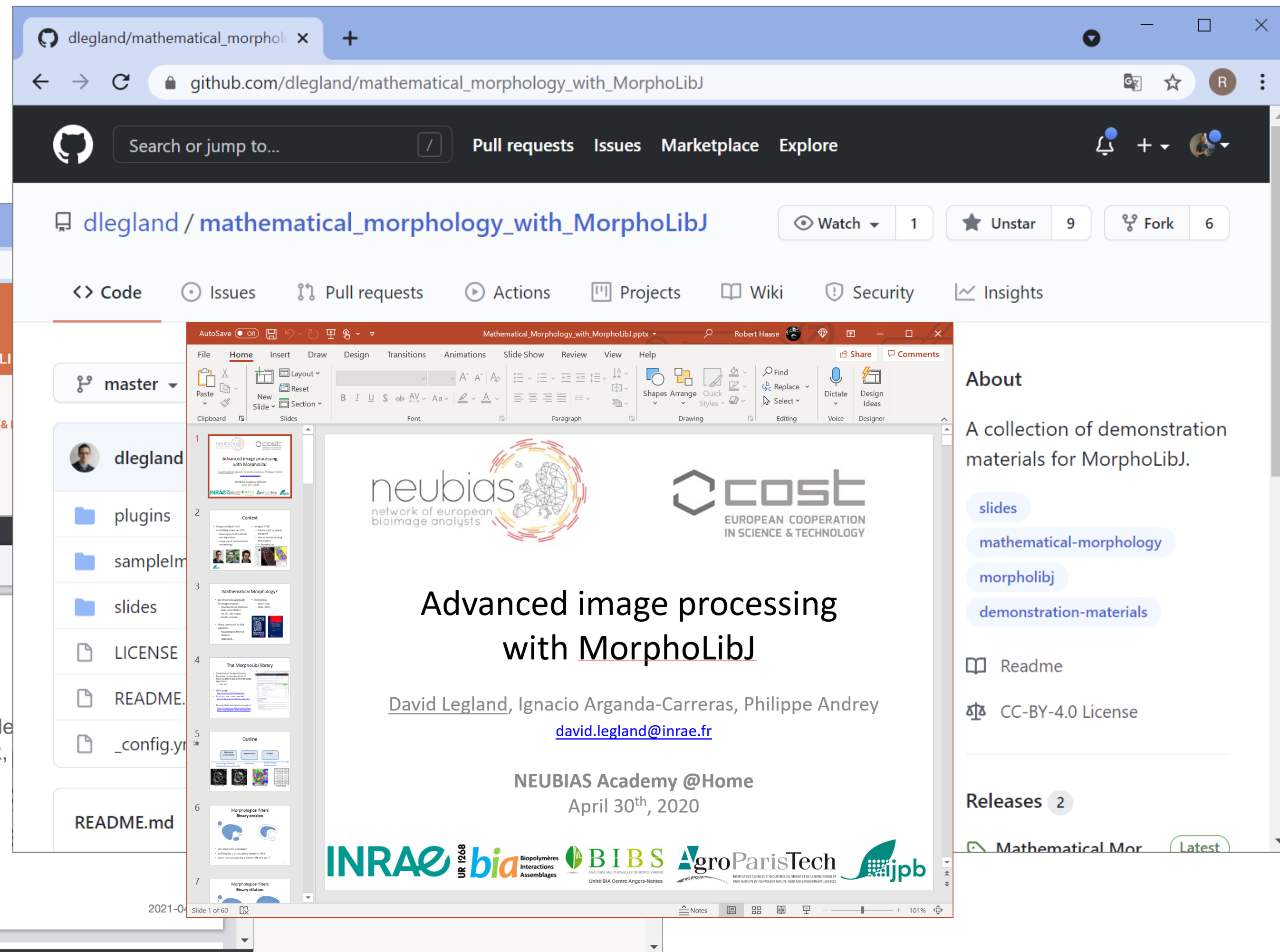
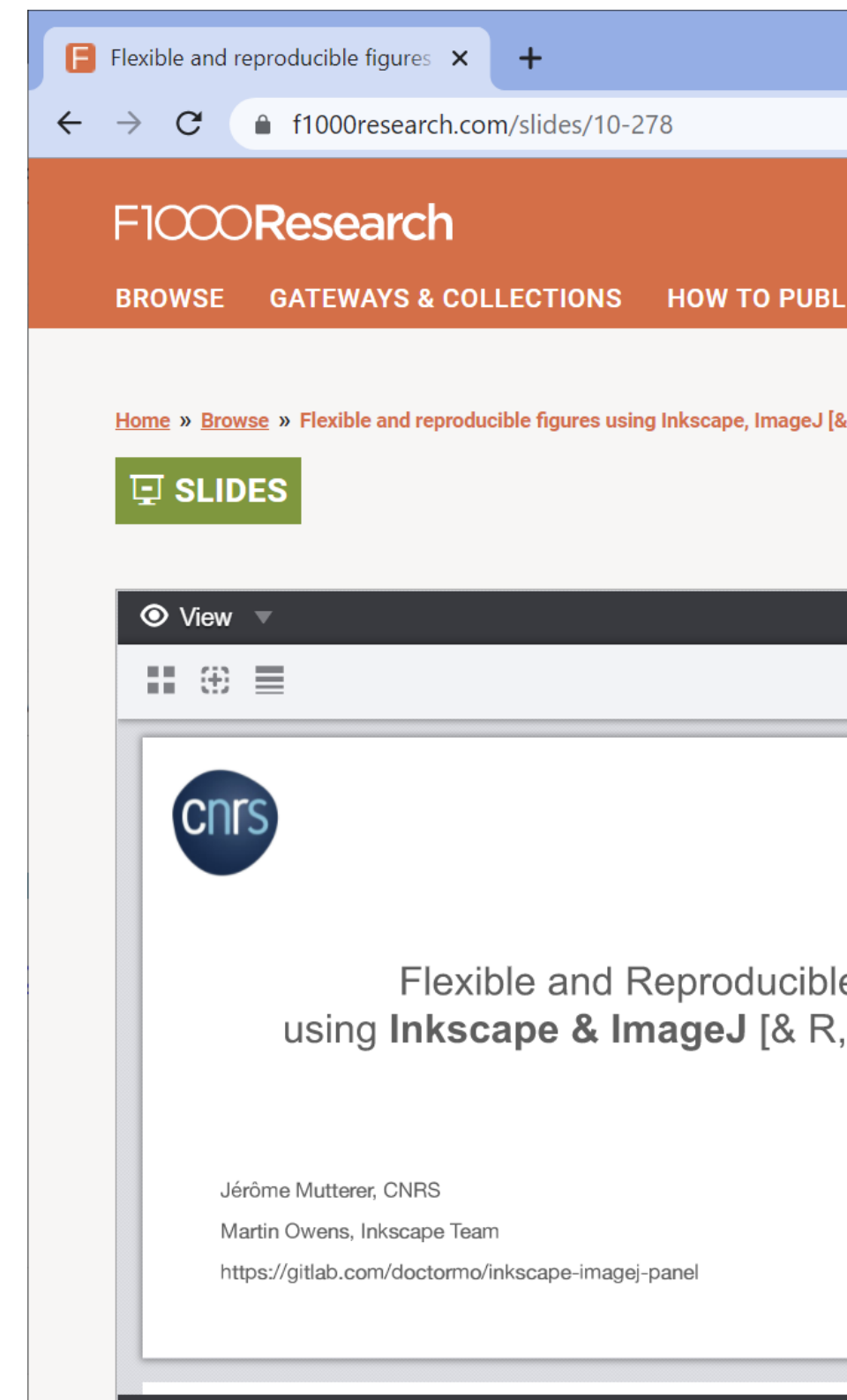
Use cases: Teaching material

Re-using, advertising your work.

<https://f1000research.com/neubias>

<https://figshare.com>

<https://github.com>



<https://f1000research.com/slides/10-278> https://github.com/dlegland/mathematical_morphology_with_MorphoLibJ

Use cases: Figures

Share efforts

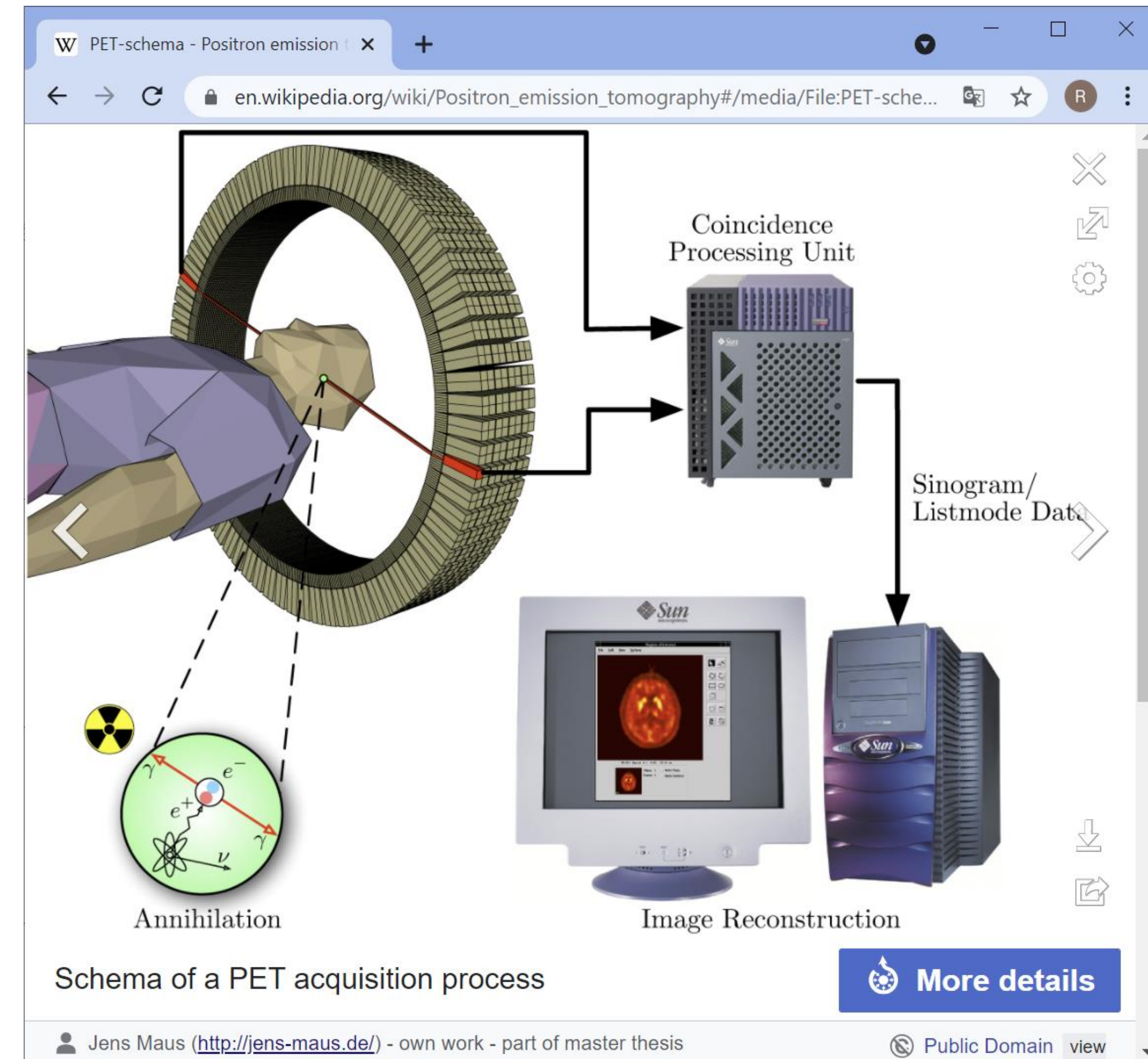
—Talk about each others' work

Advertise your work

... because our work is often publicly funded

https://commons.wikimedia.org/wiki/Main_Page

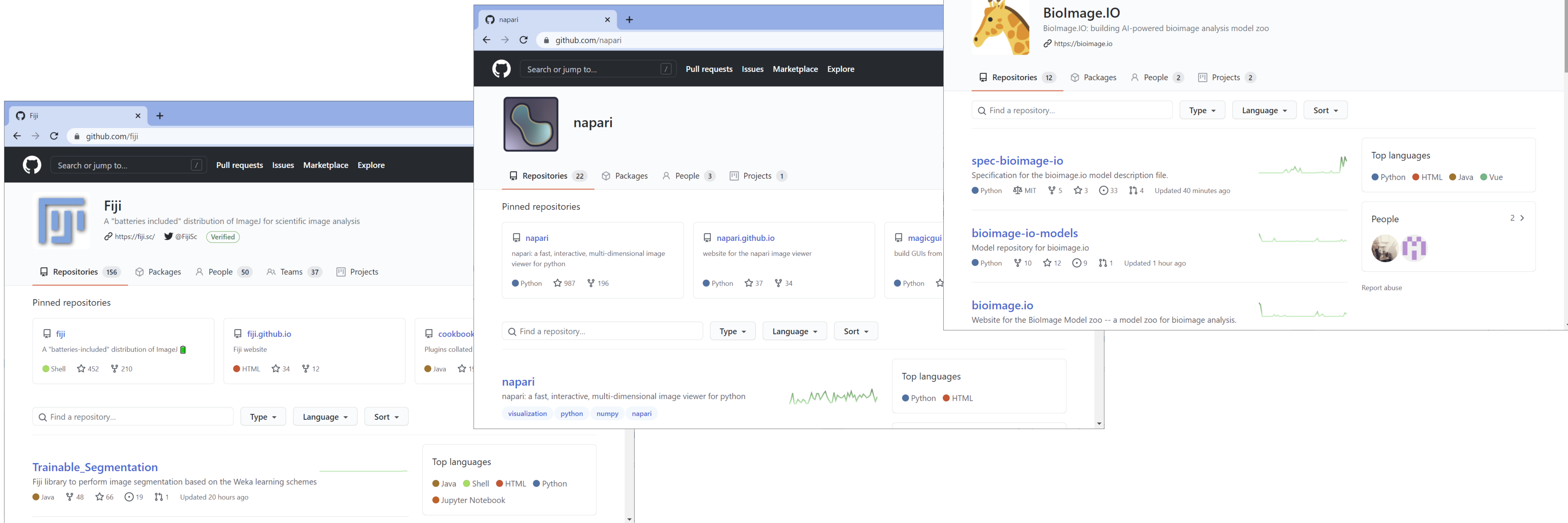
<https://figshare.com>



https://en.wikipedia.org/wiki/Positron_emission_tomography#/media/File:PET-schema.png

Use cases: Code

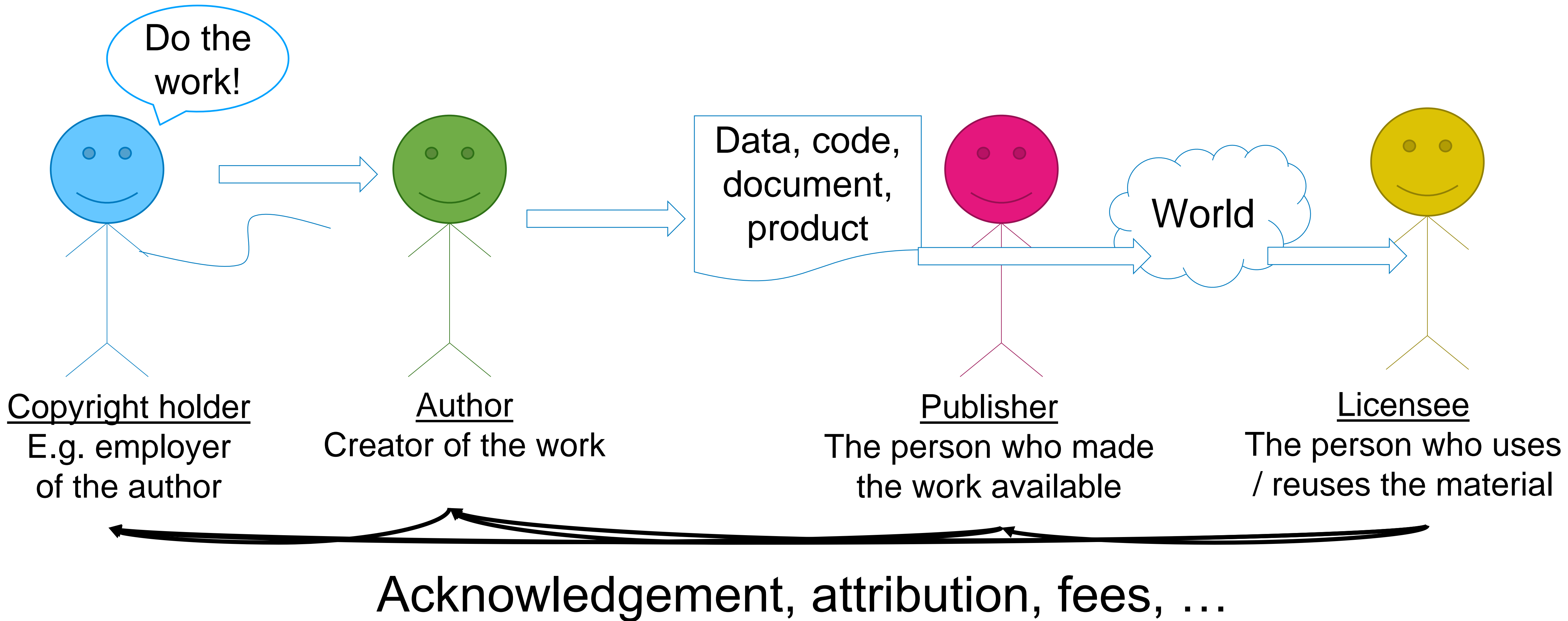
Collaboration in open-source projects *unthinkable* without openly sharing and transparent licensing



The image displays three overlapping screenshots of GitHub profiles, illustrating open-source collaboration. The top-left screenshot shows the profile for **Fiji**, described as a "batteries included" distribution of ImageJ for scientific image analysis. The middle screenshot shows the profile for **napari**, a fast, interactive, multi-dimensional image viewer for Python. The rightmost screenshot shows the profile for **BioImage.IO**, building an AI-powered bioimage analysis model zoo. Each profile includes statistics for repositories, packages, people, and projects, along with pinned repositories and top languages used.

<https://github.com/fiji> <https://github.com/napari> <https://github.com/bioimage-io>

Terminology



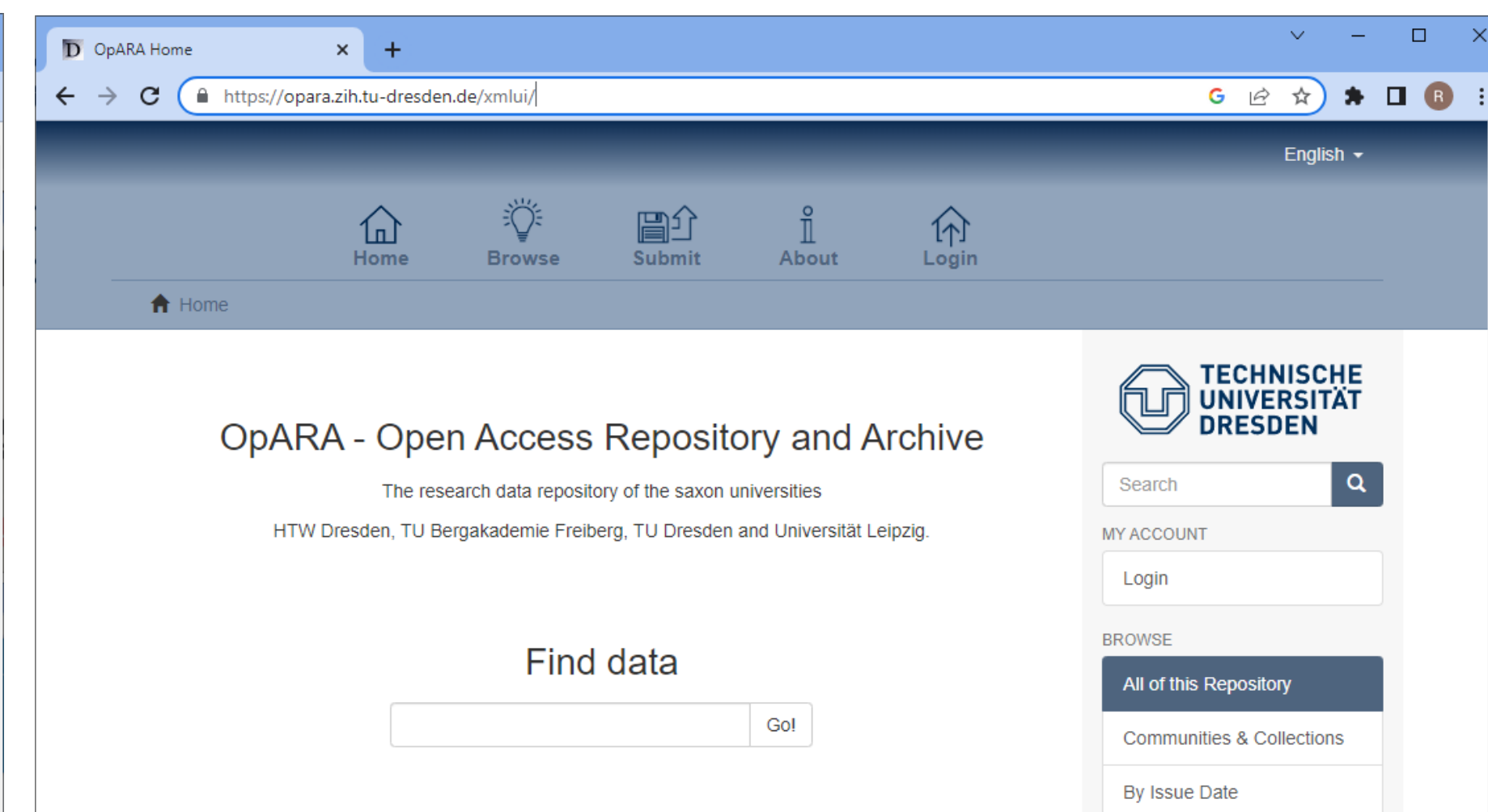
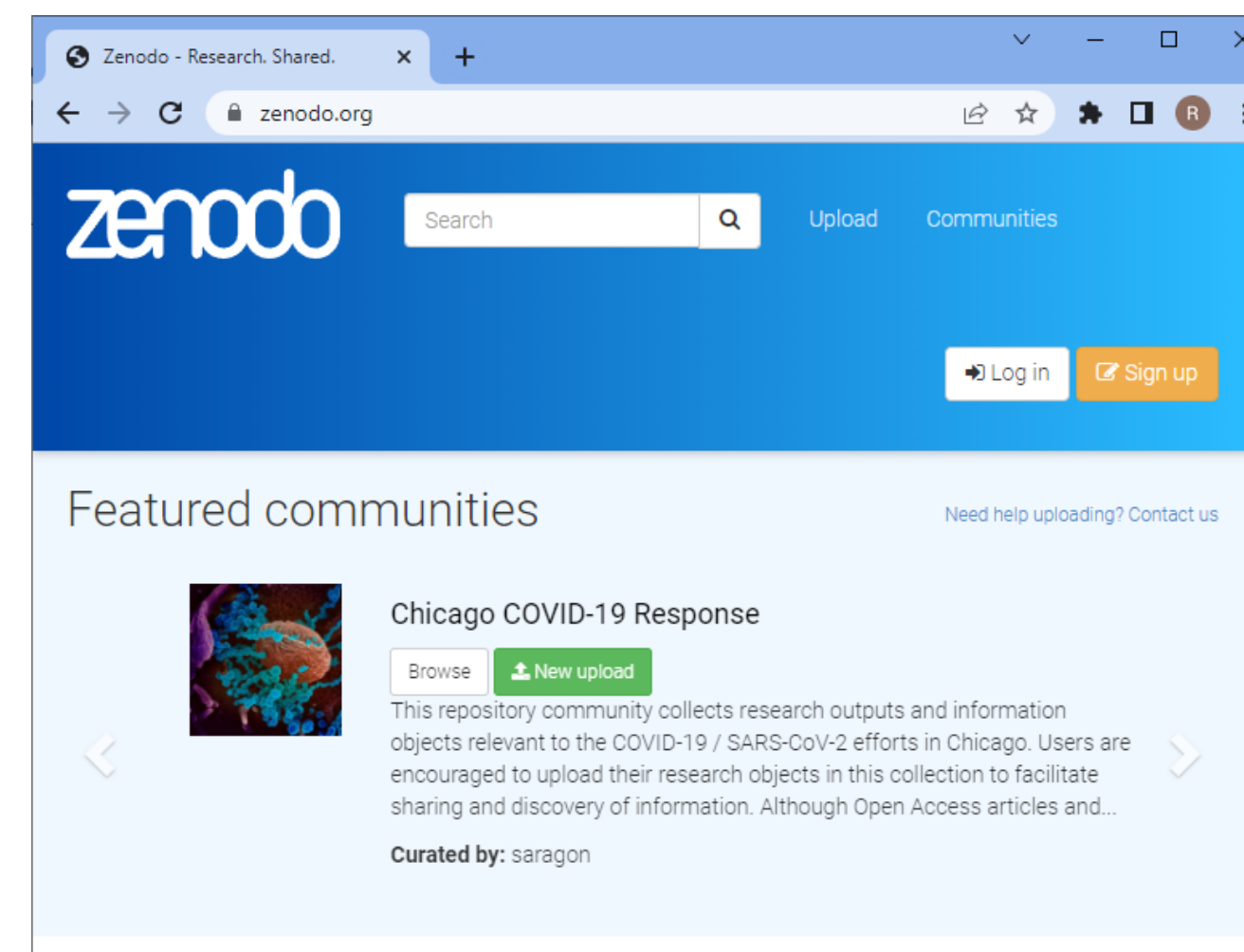
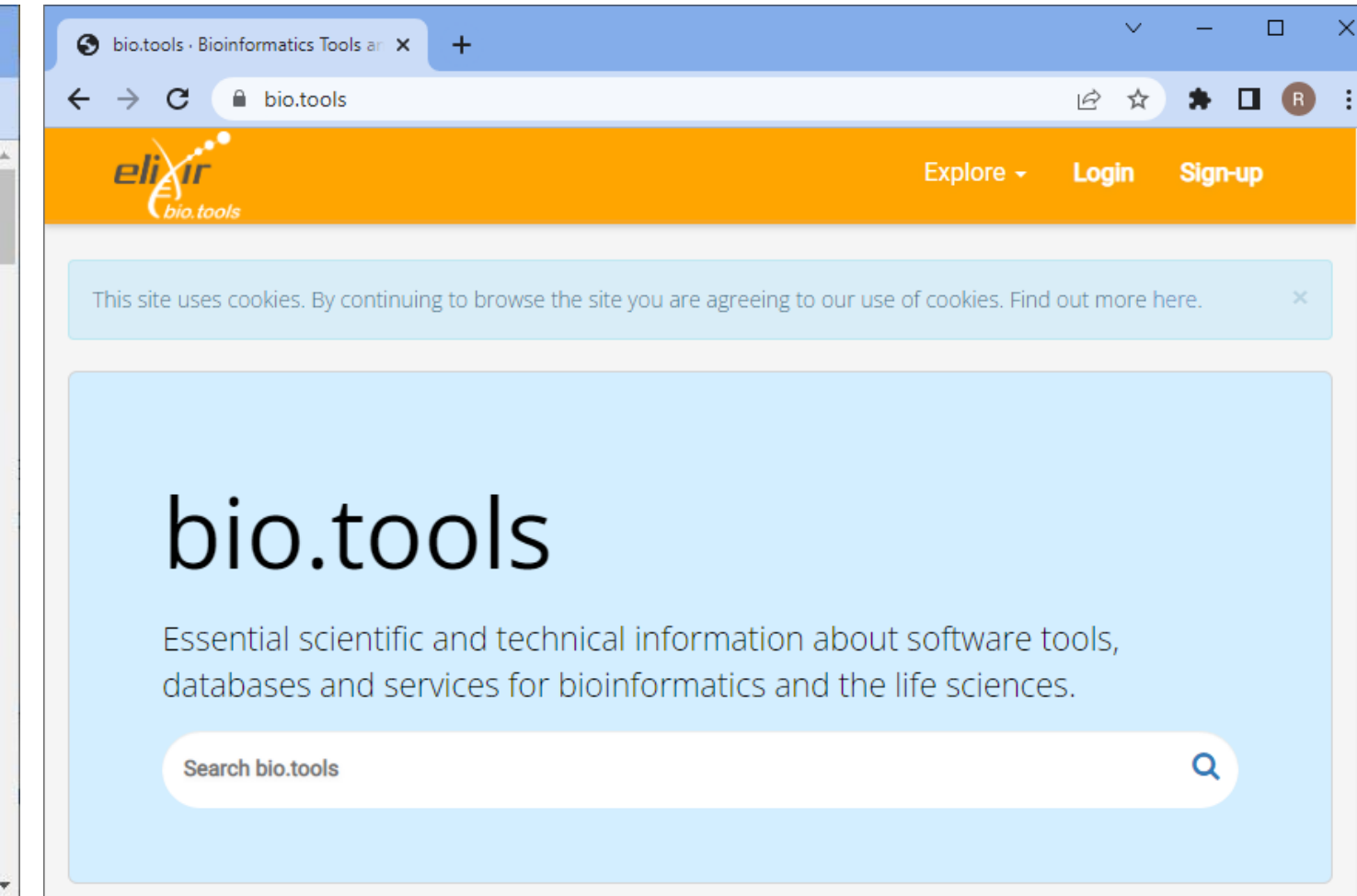
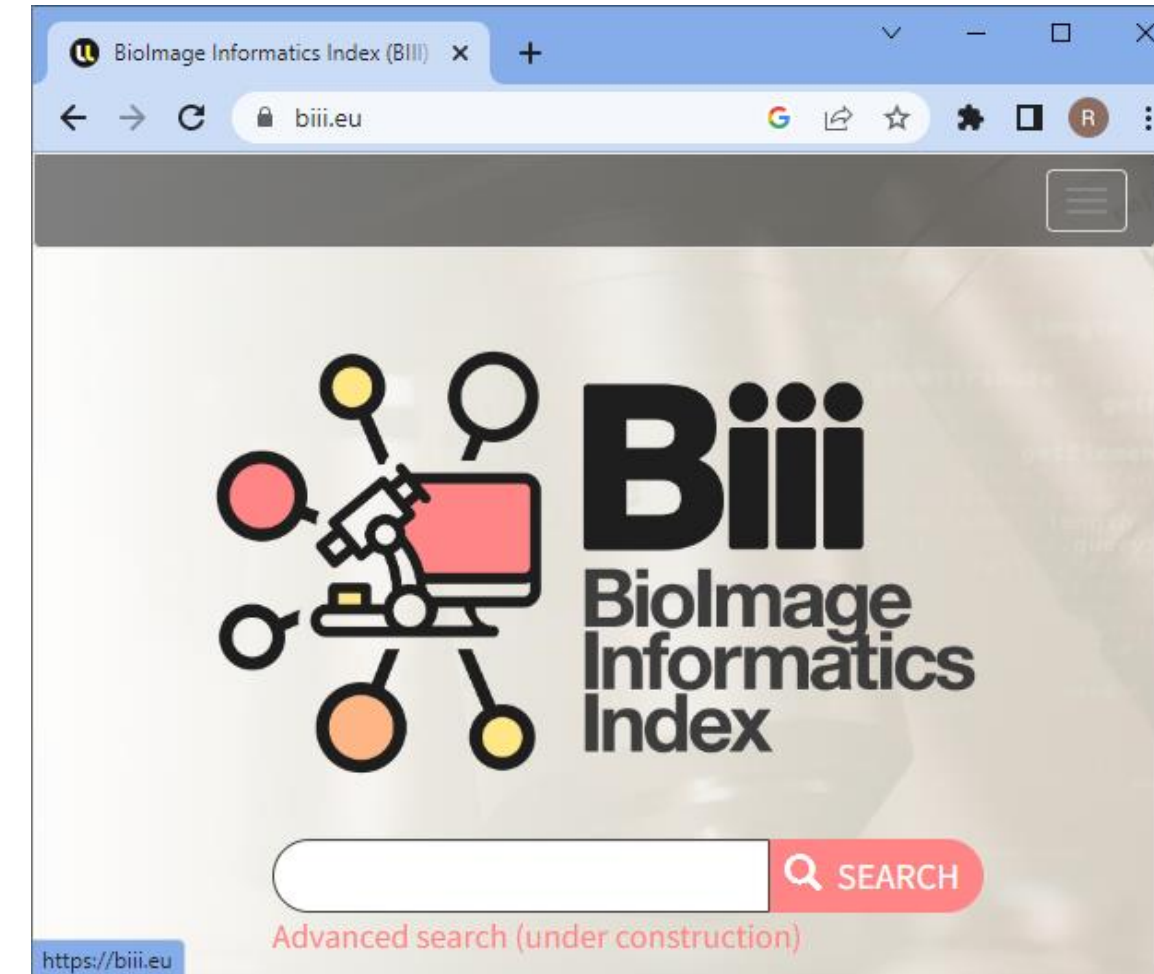
FAIR principles

- Findable
- Accessible
- Interoperable
- Reusable

=> State-of-the-art Research Data Management (RDM)

FAIR principles: Findable

- Research data / code / ... can be found if it's listed in *repositories*
 - Preferably: global, public, field-specific repository
 - Alternative: institutional repository
- Findability can be improved through attaching
 - *meta data*
 - *unique digital object identifiers (DOI)*



<https://biii.eu> <https://bio.tools> <https://zenodo.org> <https://opara.zih.tu-dresden.de/xmlui/>

Quiz: Digital object identifiers

Which of these is a *unique* digital object identifier?

<https://twitter.com/haesleinhuepf/status/891596662782779392>

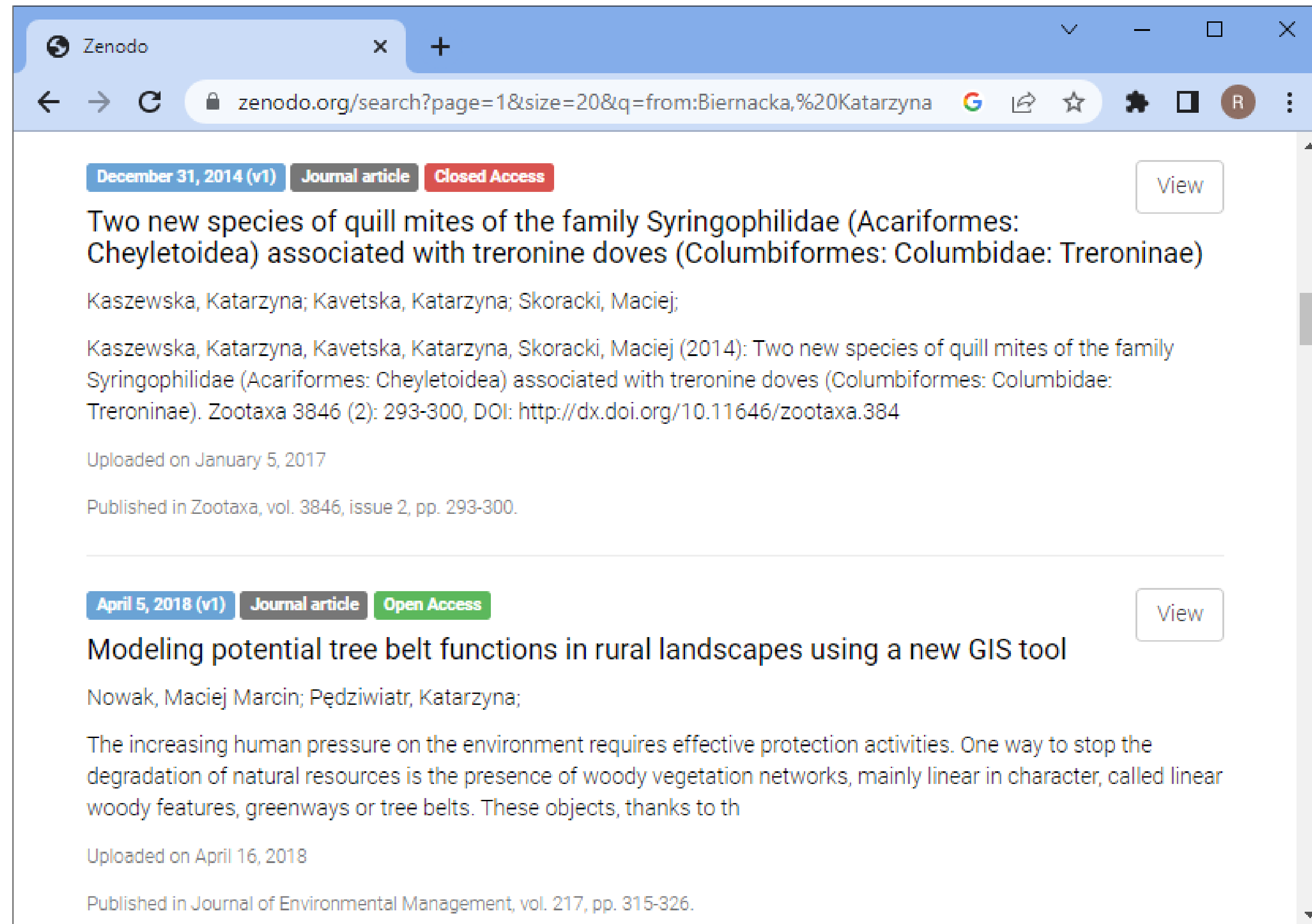
<https://doi.org/10.5281/zenodo.28325>

<https://github.com/haesleinhuepf/devbio-napari>

<https://napari.org/>

FAIR principles: Accessible

- Research data can be made accessible (after it was found by potential users)
 - Open Access is just one form of accessibility
 - Authentication enables other forms



The screenshot shows a web browser window with the Zenodo search results page. The address bar shows the URL: zenodo.org/search?page=1&size=20&q=from:Biernacka,%20Katarzyna. The page displays two search results:

December 31, 2014 (v1) Journal article Closed Access [View](#)

Two new species of quill mites of the family Syringophilidae (Acariformes: Cheyletoidea) associated with treronine doves (Columbiformes: Columbidae: Treroninae)
Kaszewska, Katarzyna; Kavetska, Katarzyna; Skoracki, Maciej;
Kaszewska, Katarzyna, Kavetska, Katarzyna, Skoracki, Maciej (2014): Two new species of quill mites of the family Syringophilidae (Acariformes: Cheyletoidea) associated with treronine doves (Columbiformes: Columbidae: Treroninae). Zootaxa 3846 (2): 293-300, DOI: <http://dx.doi.org/10.11646/zootaxa.384>
Uploaded on January 5, 2017
Published in Zootaxa, vol. 3846, issue 2, pp. 293-300.

April 5, 2018 (v1) Journal article Open Access [View](#)

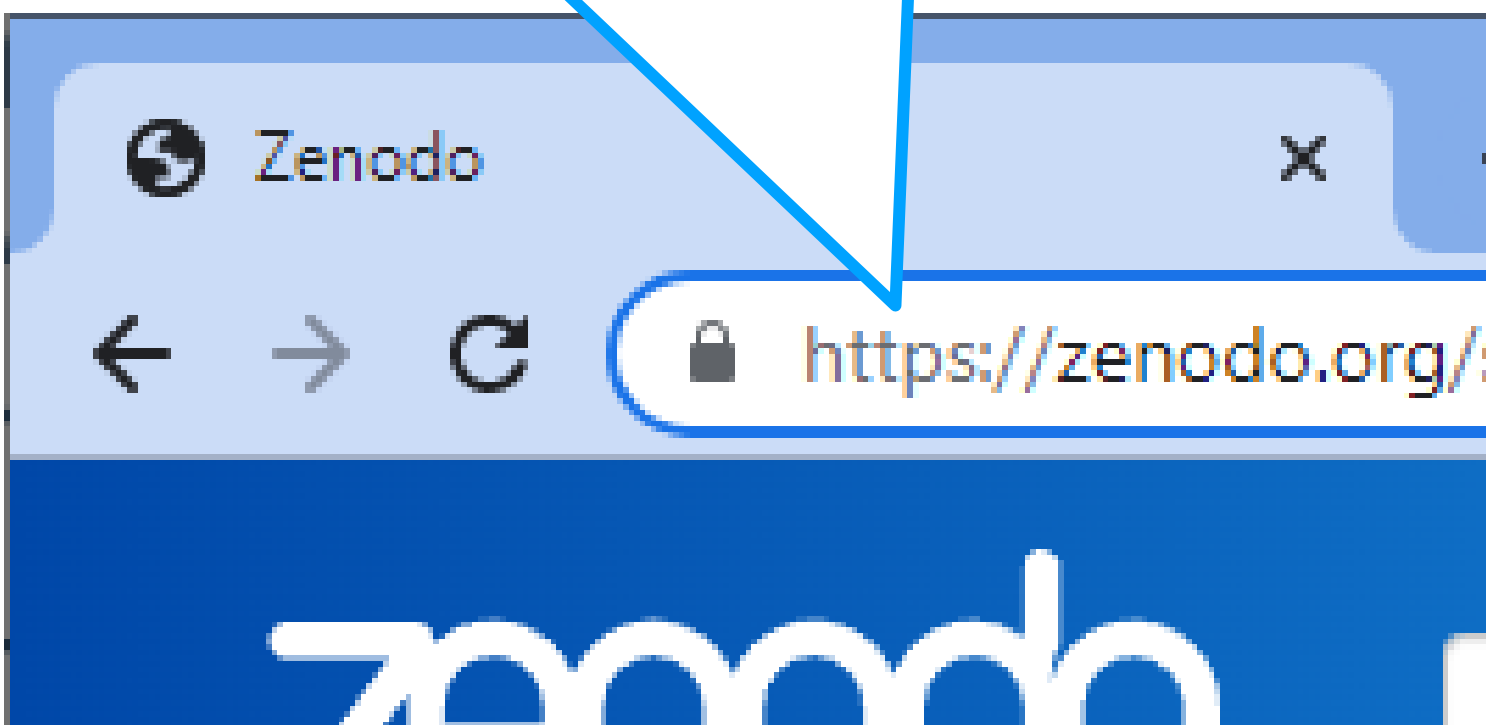
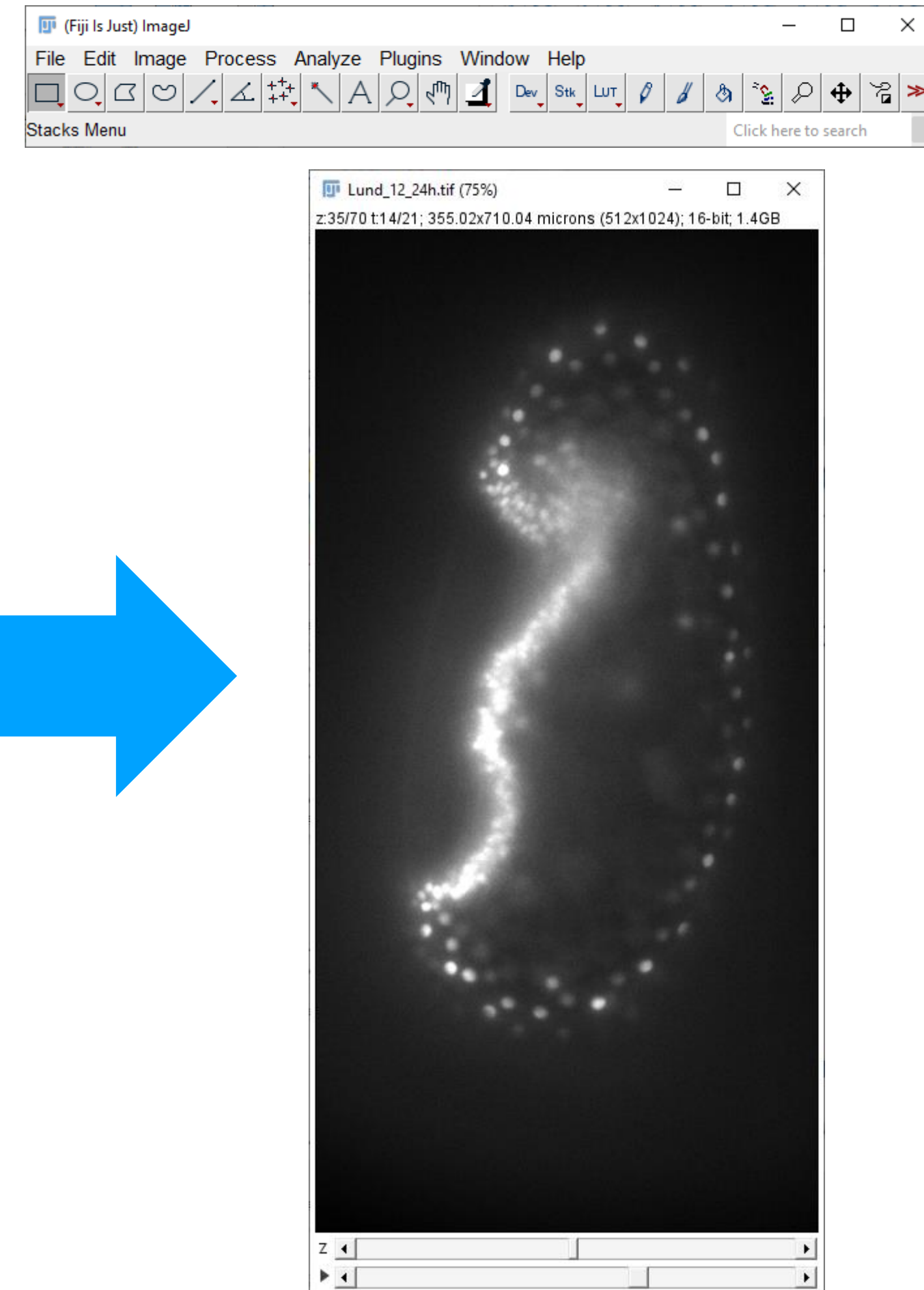
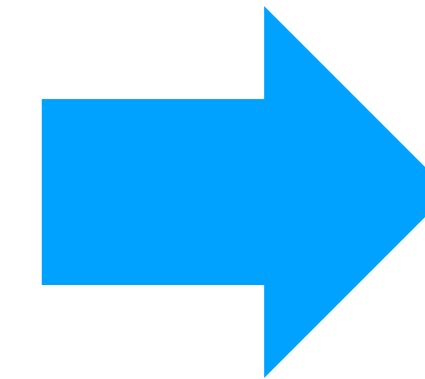
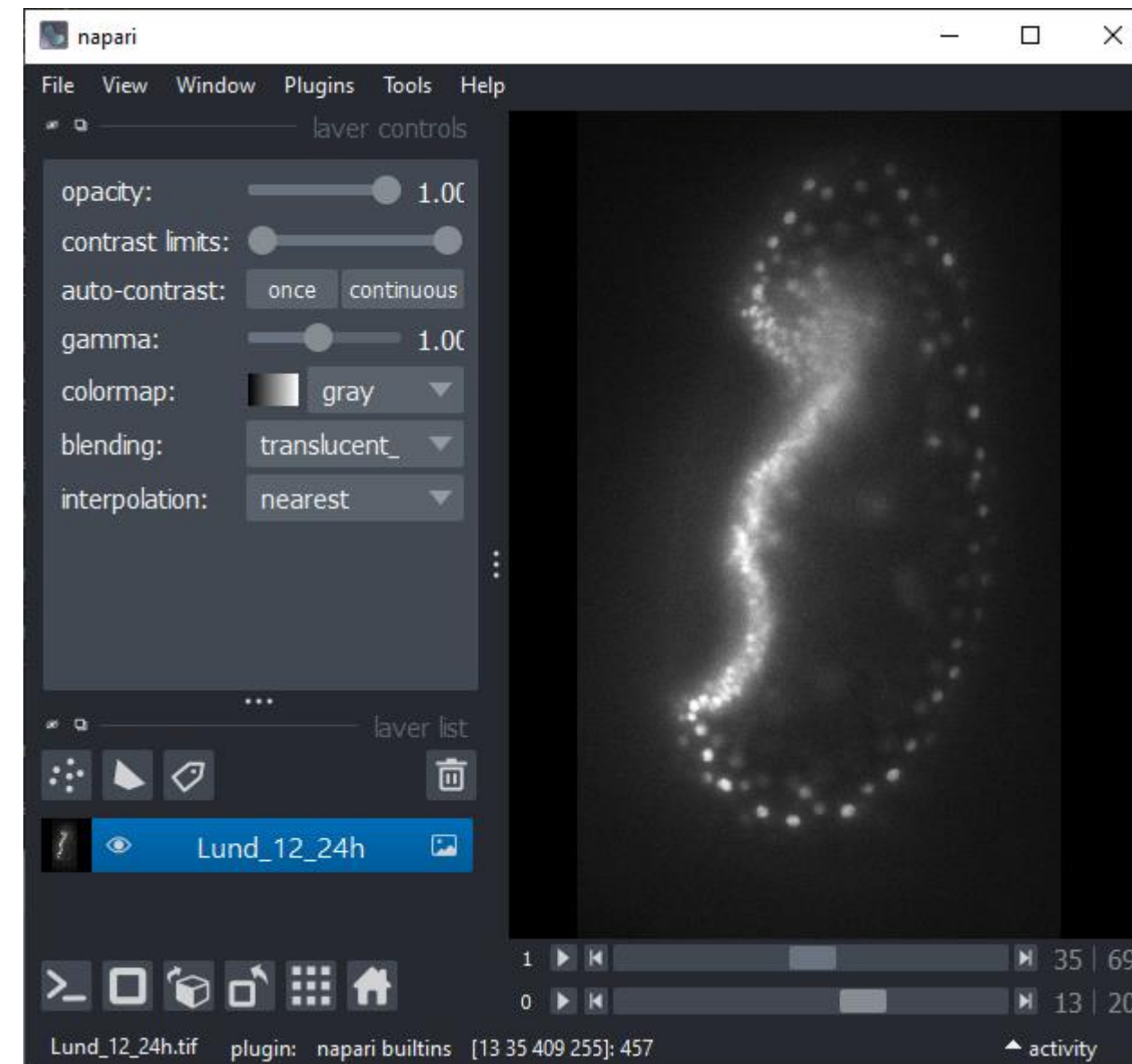
Modeling potential tree belt functions in rural landscapes using a new GIS tool
Nowak, Maciej Marcin; Pędziwiatr, Katarzyna;
The increasing human pressure on the environment requires effective protection activities. One way to stop the degradation of natural resources is the presence of woody vegetation networks, mainly linear in character, called linear woody features, greenways or tree belts. These objects, thanks to th
Uploaded on April 16, 2018
Published in Journal of Environmental Management, vol. 217, pp. 315-326.

FAIR principles: Interoperable

Data can be opened in multiple software through open and documented...

- file-formats
- protocols

hypertext transfer protocol (secure)

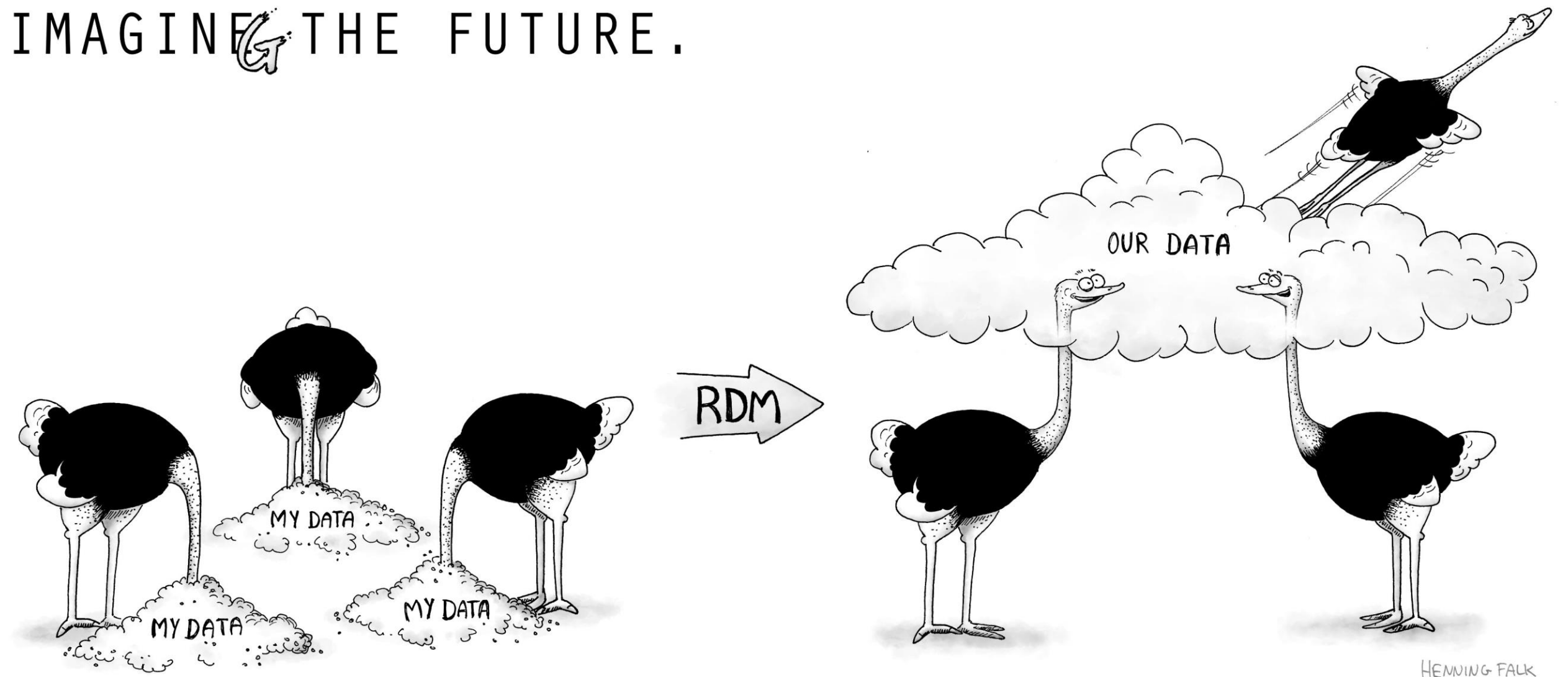


FAIR principles: Reusable

Data can be reused if

- Other FAIR principles are fulfilled
- Data is properly licensed
 - copyright statement is given
 - No copyright statement means: You do not have the right to copy

IMAGINE THE FUTURE.



Licensing: Creative Commons (CC)

Public domain (CC0)

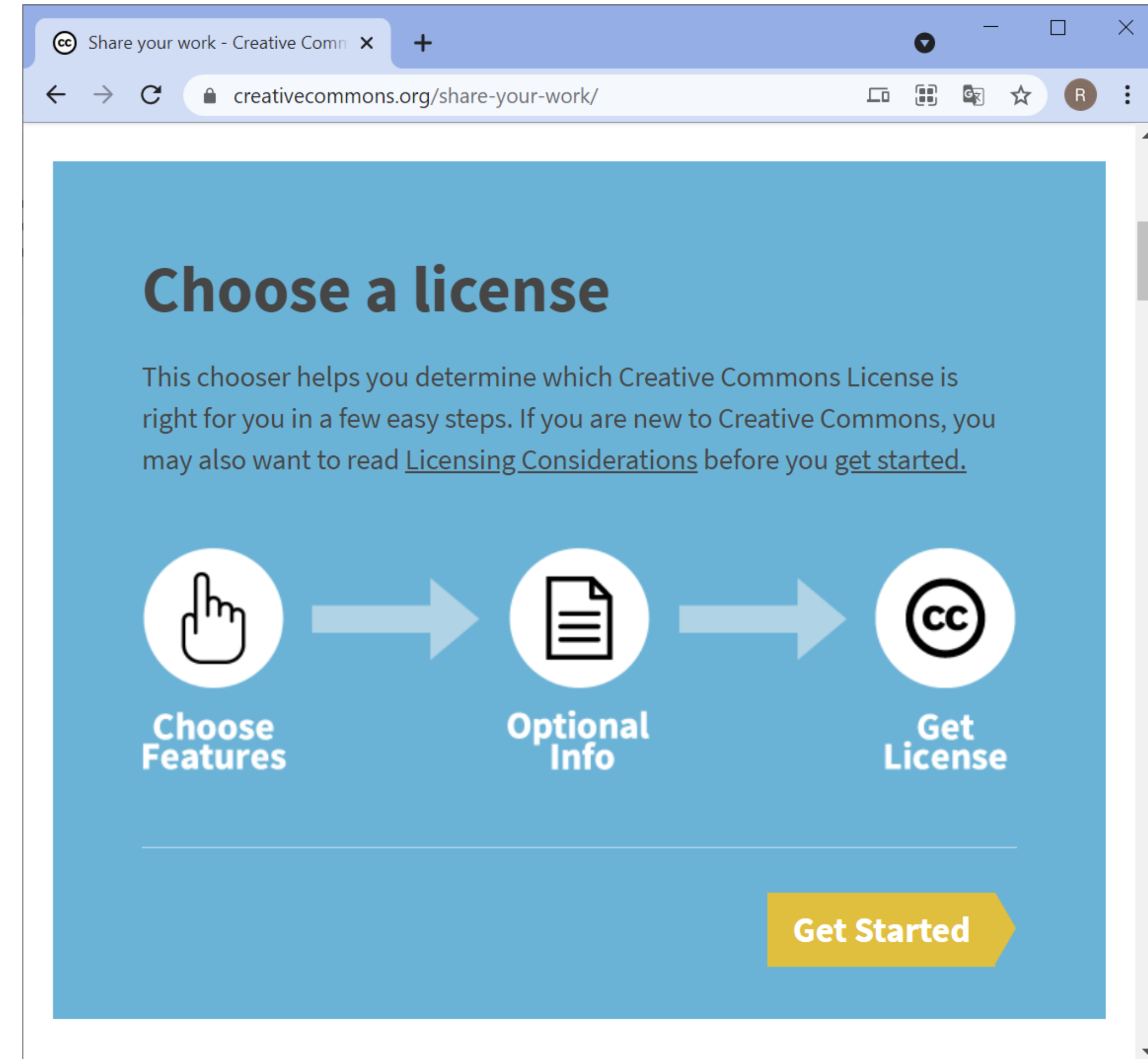
Attribution International (CC-BY)

Attribution ShareAlike Int. (CC-BY-SA)

Attribution Non-Commercial Int. (CC-BY-NC)

Attribution NoDerivatives Int. (CC-BY-ND)

+ Combinations, e.g. CC-BY-NC-ND



Content on this site is licensed under a [Creative Commons Attribution 4.0 International license](https://creativecommons.org/licenses/by/4.0/). [Icons](#) by The Noun Project

Licensing: Creative Commons (CC)



Public domain (CC0)

Everyone can reuse without mentioning the source or author of the shared resource.

Public domain licenses cannot be revoked.

The author must own the right to copy (copyright) the resource.

—If you authored work as part of your job, you may not be the copyright holder. (check your employers' guidelines)

Employers don't like this one because you give away the rights to *exploit* you work.

Content on this slide was adapted from <https://creativecommons.org/about/ccllicenses/> which is licensed under a [Creative Commons Attribution 4.0 International license](https://creativecommons.org/licenses/by/4.0/). Icons by The Noun Project.

<https://creativecommons.org/share-your-work/>
<https://creativecommons.org/about/ccllicenses/>


Licensing: Creative Commons (CC)



“By attribution”

CC BY: This license allows reusers to distribute, remix, adapt, and build upon the material in any medium or format, so long as attribution is given to the creator. The license allows for commercial use.

CC BY includes the following elements:

BY  – Credit must be given to the creator

Content on this slide was adapted from <https://creativecommons.org/about/ccllicenses/> which is licensed under a [Creative Commons Attribution 4.0 International license](https://creativecommons.org/licenses/by/4.0/). Icons by The Noun Project.

<https://creativecommons.org/share-your-work/>
<https://creativecommons.org/about/ccllicenses/>

Licensing: Creative Commons (CC)



Example



You must put such a sentence and keep the link to CC-BY

Figure adapted from <https://www.openmicroscopy.org/> licensed by University of Dundee & Open Microscopy Environment under [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/)

Licensing: Creative Commons (CC)



CC BY-SA: This license allows reusers to distribute, remix, adapt, and build upon the material in any medium or format, so long as attribution is given to the creator. The license allows for commercial use. If you remix, adapt, or build upon the material, you must license the modified material under identical terms.

CC BY-SA includes the following elements:

BY  – Credit must be given to the creator

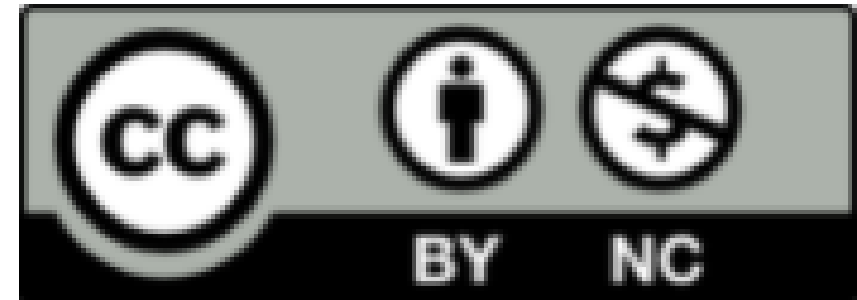
SA  – Adaptations must be shared under the same terms

“Share alike”

“Restrictive”
licensing

Content on this slide was adapted from <https://creativecommons.org/about/ccllicenses/> which is licensed under a [Creative Commons Attribution 4.0 International license](https://creativecommons.org/licenses/by-sa/4.0/). [Icons](https://creativecommons.org/licenses/by-sa/4.0/) by The Noun Project.


Licensing: Creative Commons (CC)



CC BY-NC: This license allows reusers to distribute, remix, adapt, and build upon the material in any medium or format for noncommercial purposes only, and only so long as attribution is given to the creator.

It includes the following elements:

BY  – Credit must be given to the creator

NC  – Only noncommercial uses of the work are permitted

Content on this slide was adapted from <https://creativecommons.org/about/ccllicenses/> which is licensed under a [Creative Commons Attribution 4.0 International license](https://creativecommons.org/licenses/by-nc/4.0/). [Icons](#) by The Noun Project.

<https://creativecommons.org/share-your-work/>
<https://creativecommons.org/about/ccllicenses/>

Licensing: Creative Commons (CC)



CC BY-ND: This license allows reusers to copy and distribute the material in any medium or format in unadapted form only, and only so long as attribution is given to the creator. The license allows for

commercial use.

CC BY-ND includes the following elements:

BY  – Credit must be given to the creator

ND  – No derivatives or adaptations of the work are permitted

“Restrictive”
licensing

Content on this slide was adapted from <https://creativecommons.org/about/ccllicenses/> which is licensed under a [Creative Commons Attribution 4.0 International license](https://creativecommons.org/licenses/by/4.0/). [Icons](https://creativecommons.org/licenses/by/4.0/) by The Noun Project.

<https://creativecommons.org/share-your-work/>
<https://creativecommons.org/about/ccllicenses/>

Licensing: Creative Commons (CC)




CC BY-NC-ND: This license allows reusers to copy and distribute the material in any medium or format in unadapted form only, for noncommercial purposes only, and only so long as attribution is given

to the creator.

CC BY-NC-ND includes the following elements:

BY  – Credit must be given to the creator

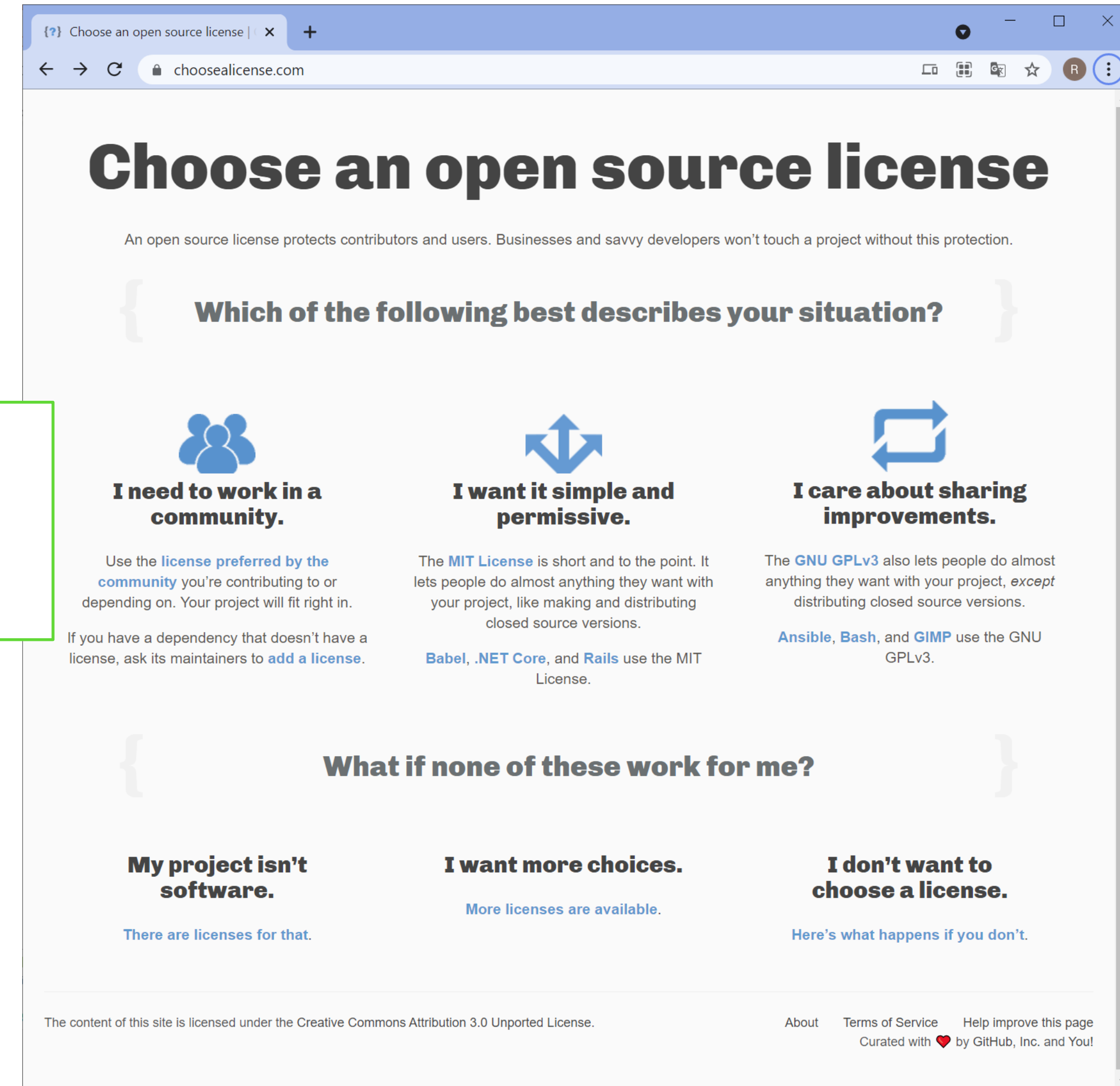
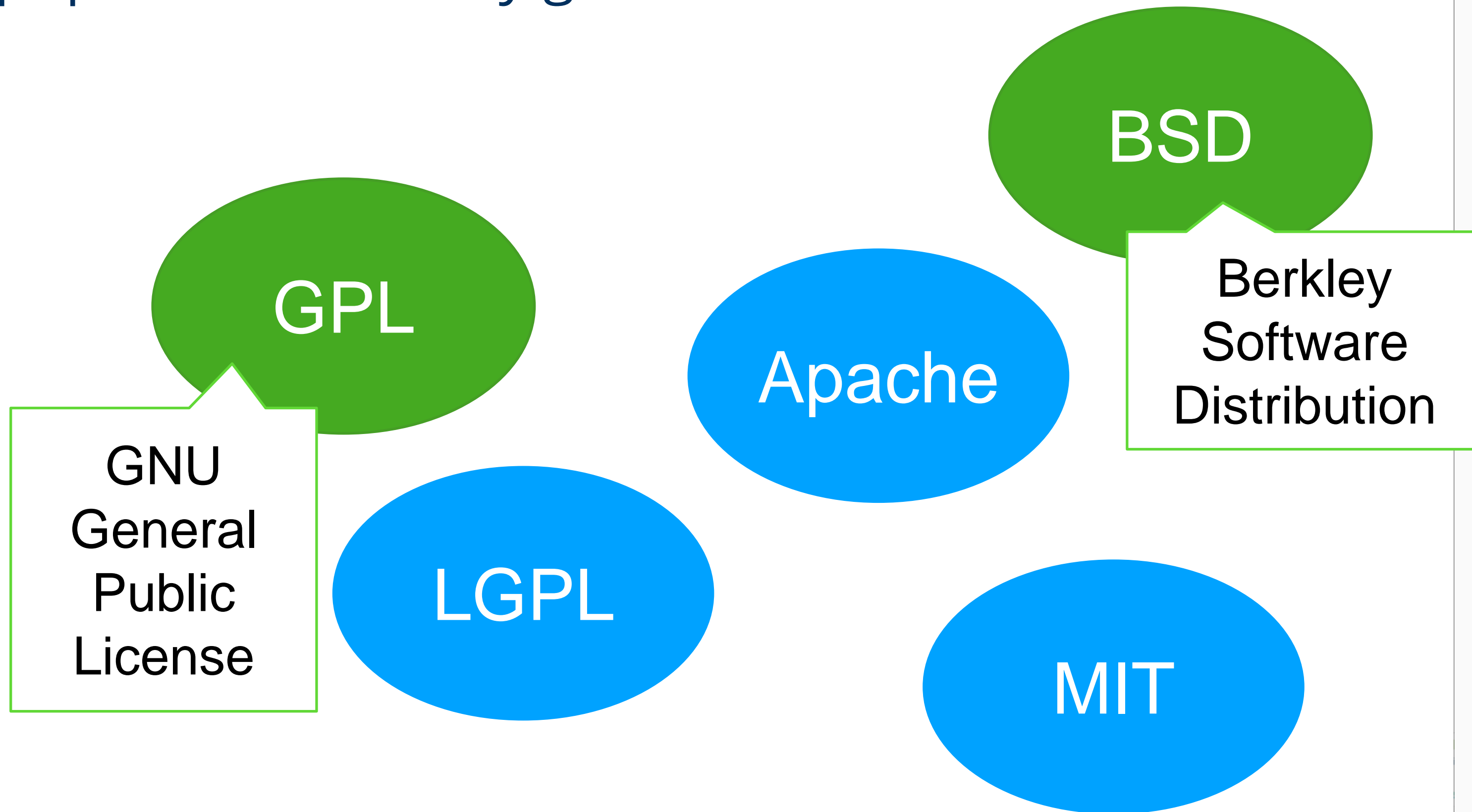
NC  – Only noncommercial uses of the work are permitted

ND  – No derivatives or adaptations of the work are permitted

Content on this slide was adapted from <https://creativecommons.org/about/ccllicenses/> which is licensed under a [Creative Commons Attribution 4.0 International license](https://creativecommons.org/licenses/by/4.0/). [Icons](#) by The Noun Project.

Licensing software

In the software world, other licenses are more popular, historically grown.



Licensing Software: GPL

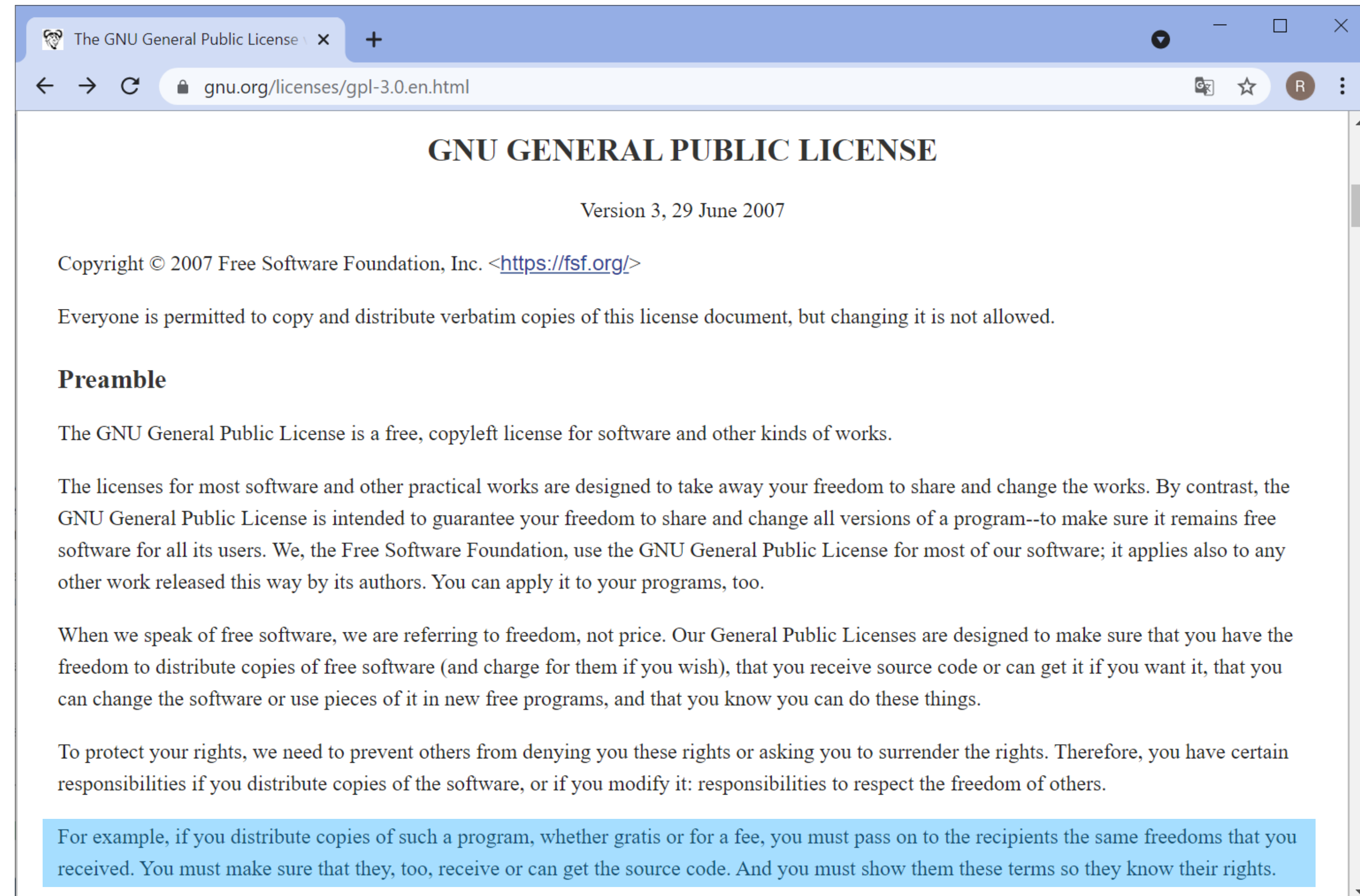
GPL

- Derivatives must also be GPL-licensed

“Restrictive”
licensing

See also:

- Lesser General Public License (LGPL)
 - Integrate LGPL-licensed code into not-LGPL-licensed code



The screenshot shows a web browser window displaying the GNU General Public License (GPL) page. The browser's address bar shows the URL gnu.org/licenses/gpl-3.0.en.html. The page content includes the title "GNU GENERAL PUBLIC LICENSE", the version "Version 3, 29 June 2007", and the copyright notice "Copyright © 2007 Free Software Foundation, Inc. <<https://fsf.org/>>". The text states that everyone is permitted to copy and distribute verbatim copies of the license document, but changing it is not allowed. The "Preamble" section explains that the GNU General Public License is a free, copyleft license for software and other kinds of works, designed to guarantee freedom to share and change all versions of a program. It also mentions that the license is used for most of the Free Software Foundation's software and can be applied to other works. A highlighted blue box at the bottom of the page contains the text: "For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights."

Licensing Software: BSD0

Copyright (C) [year] by [copyright holder] <[email]>

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted.

} Similar to CC0
(public domain)

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE

Licensing Software: BSD0

Copyright (C) [year] by [copyright holder] <[email]>

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE

Disclaimer

Whatever you do with it,
we [the authors] are not
liable

Relevant for [image]
data analysis script-
authors!

Licensing Software: BSD2

Copyright (c) <year>, <copyright holder>

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

} Similar to CC-BY

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE

Licensing Software: BSD3

Copyright <year> <copyright holder>

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE

Similar to CC-BY

You must not use the copyright holder's name to endorse your derivative of the work.

That's also part of all CC-BY licenses

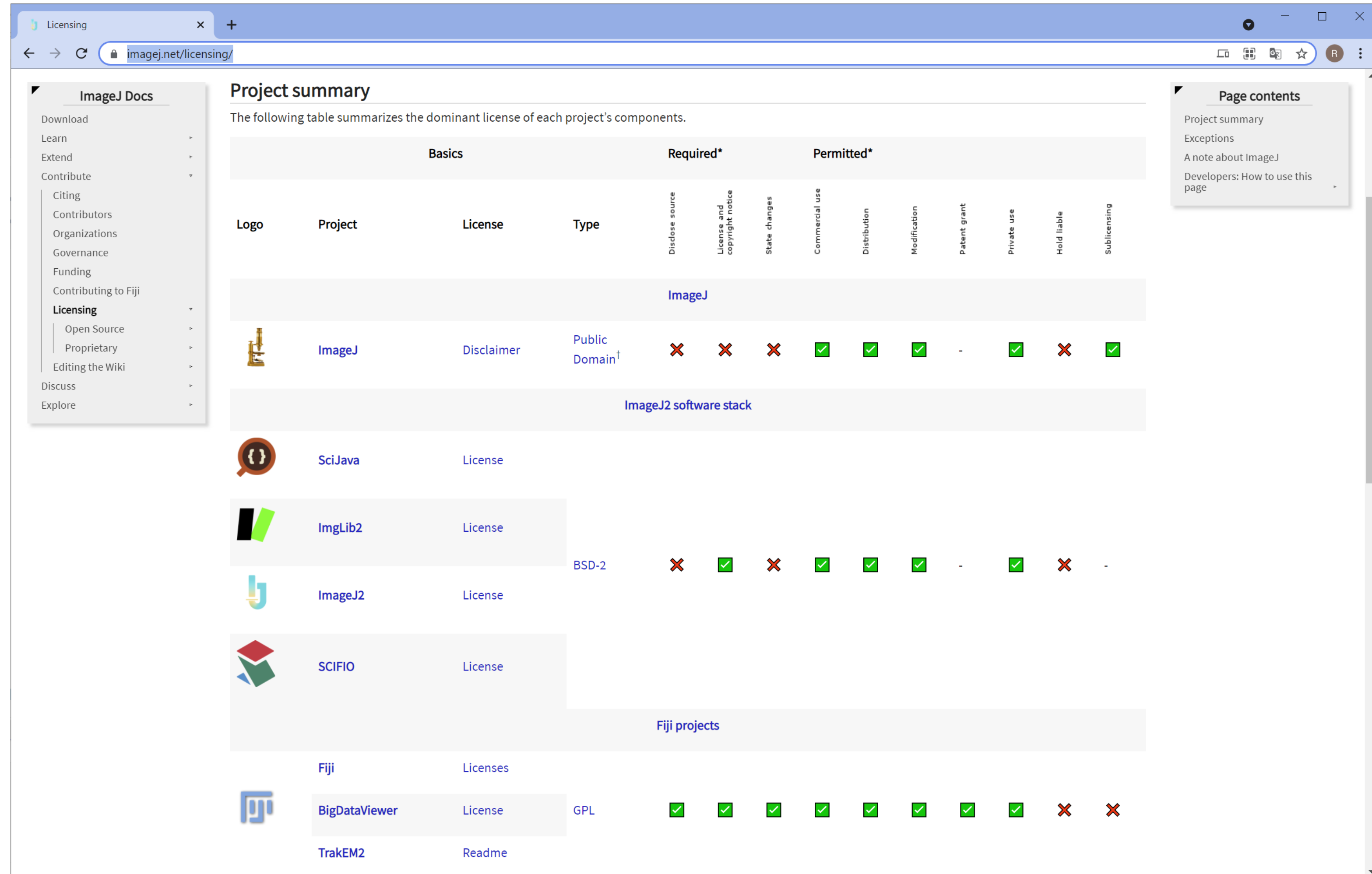
How/when to decide for a license

How?









- Choose a license compatible with your ecosystem

When?

- When the project starts
- As early as possible
- Changing the license later may be hard.
- Beware: Your employer might be the copyright holder. They have the final word on how to license, publish and make accessible your work!



The screenshot shows the 'Project summary' page on imagej.net/licensing/. The page contains a table summarizing the dominant license of each project's components. The table has columns for 'Basics', 'Required*', and 'Permitted*'. The 'Required*' and 'Permitted*' columns are further divided into sub-columns: 'Disclose source', 'License and copyright notice', 'State changes', 'Commercial use', 'Distribution', 'Modification', 'Patent grant', 'Private use', 'Hold liable', and 'Sublicensing'.

Logo	Project	License	Type	Required*			Permitted*						
				Disclose source	License and copyright notice	State changes	Commercial use	Distribution	Modification	Patent grant	Private use	Hold liable	Sublicensing
ImageJ													
	ImageJ	Disclaimer	Public Domain†	✗	✗	✗	✓	✓	✓	-	✓	✗	✓
ImageJ2 software stack													
	SciJava	License											
	ImgLib2	License	BSD-2	✗	✓	✗	✓	✓	✓	-	✓	✗	-
	ImageJ2	License											
	SCIFIO	License											
Fiji projects													
	Fiji	Licenses											
	BigDataViewer	License	GPL	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗
	TrakEM2	Readme											

Permissive versus restrictive

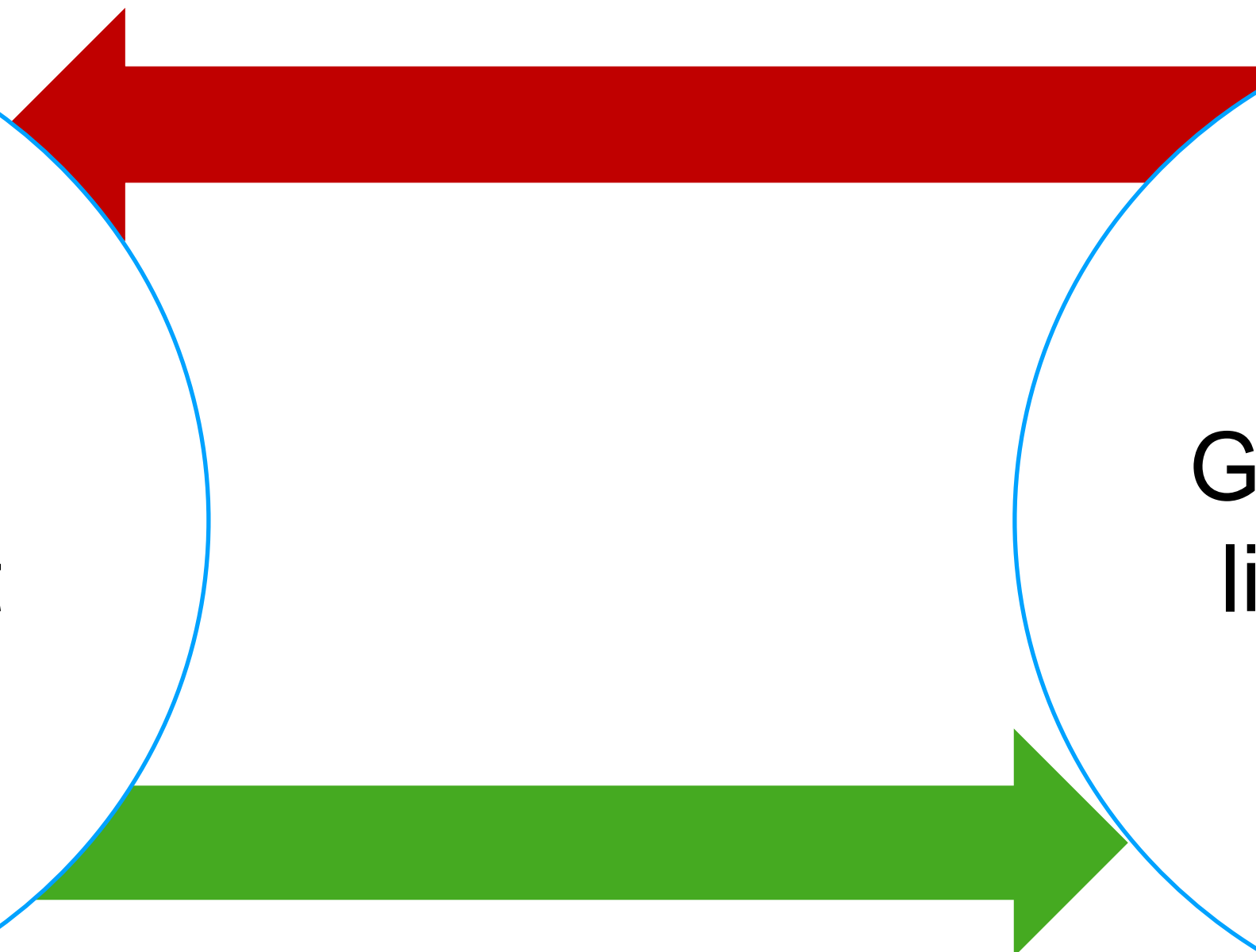
Use a license which is compatible to other projects you're collaborating with and fits to your needs / role.

If you want your stuff to be used, use BSD / CC-BY

BSD3 / CC-BY
licensed project

If you want to make use of many other things, use GPL / CC-BY-SA

GPL / CC-BY-SA
licensed project



Quiz

May I use one of the Figures from this preprint?

May I download and redistribute this preprint to students of a course for free?



bioRxiv posts many COVID19-related papers. A reminder: they have not been formally peer-reviewed and should not guide health-related behavior or be reported in the press as conclusive.

New Results

Follow this preprint

Content-Aware Image Restoration: Pushing the Limits of Fluorescence Microscopy

Martin Weigert, Uwe Schmidt, Tobias Boothe, Andreas Müller, Alexandr Dibrov, Akanksha Jain, Benjamin Wilhelm, Deborah Schmidt, Coleman Broaddus, Siân Culley, Mauricio Rocha-Martins, Fabián Segovia-Miranda, Caren Norden, Ricardo Henriques, Marino Zerial, Michele Solimena, Jochen Rink, Pavel Tomancak, Loic Royer, Florian Jug, Eugene W. Myers

doi: <https://doi.org/10.1101/236463>

Now published in *Nature Methods* doi: [10.1038/s41592-018-0216-7](https://doi.org/10.1038/s41592-018-0216-7)



Abstract Full Text **Info/History** Metrics Preview PDF

ARTICLE INFORMATION

doi <https://doi.org/10.1101/236463>

History July 3, 2018.

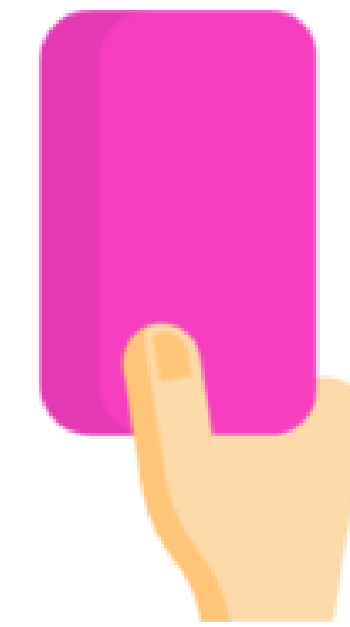
ARTICLE VERSIONS

Copyright The copyright holder for this preprint is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a [CC-BY-NC-ND 4.0 International license](https://creativecommons.org/licenses/by-nc-nd/4.0/).

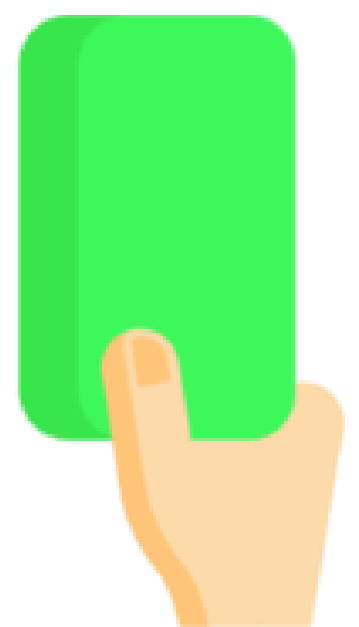
You are viewing Version 5, the most recent version of this article.

Copyright The copyright holder for this preprint is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a [CC-BY-NC-ND 4.0 International license](https://creativecommons.org/licenses/by-nc-nd/4.0/).

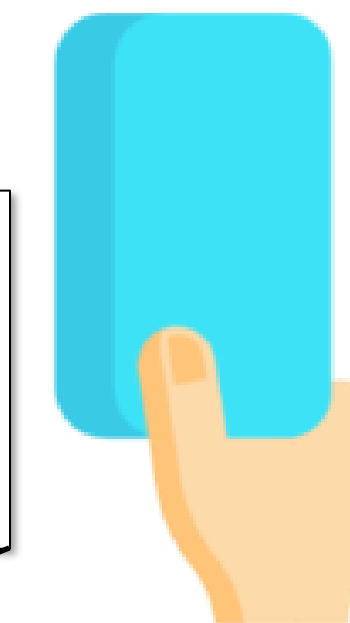
No



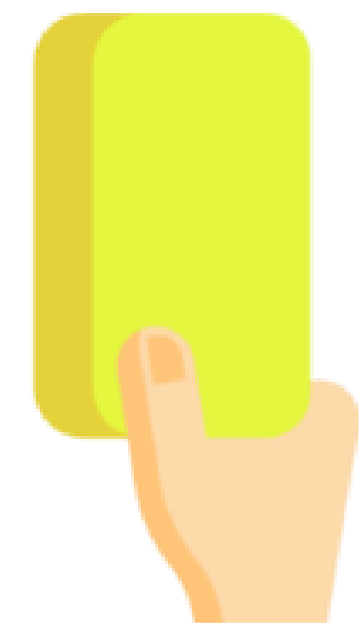
Yes



No



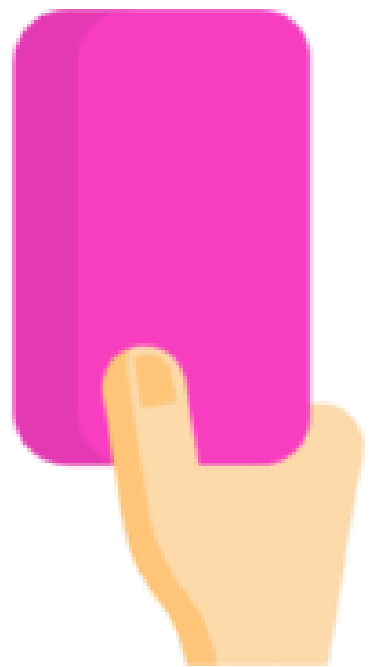
Yes



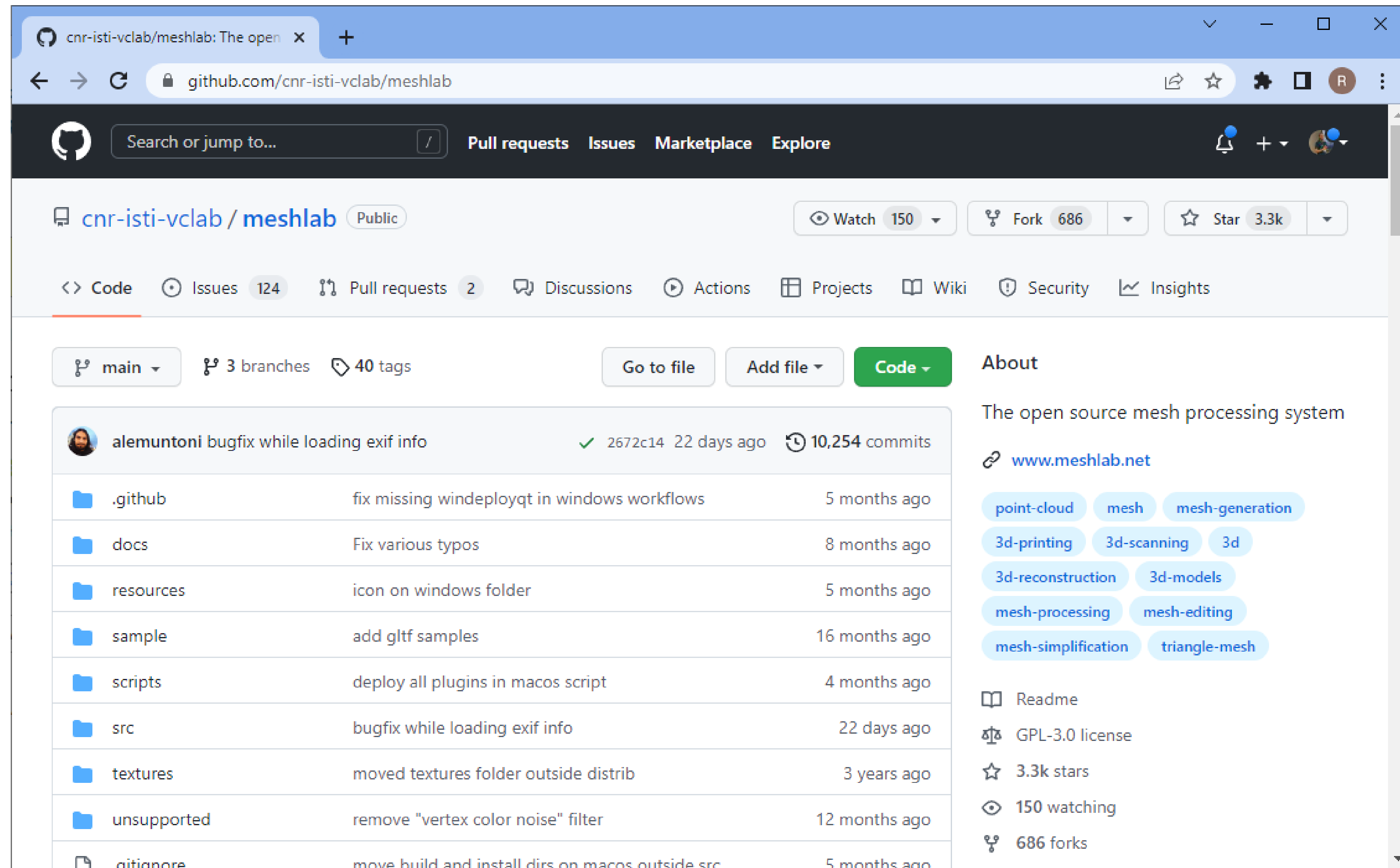
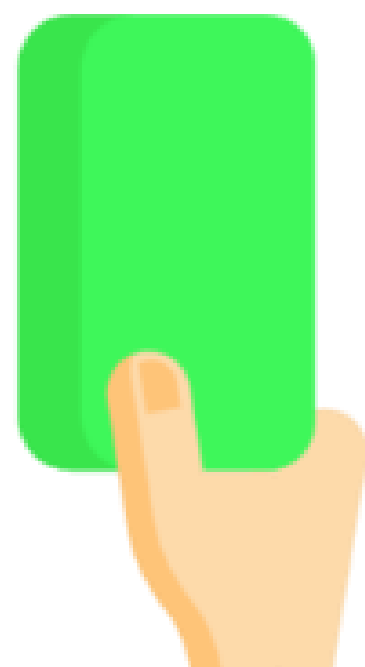
Quiz

May I reuse code from this repository in my own BSD-licensed work?

No



Yes

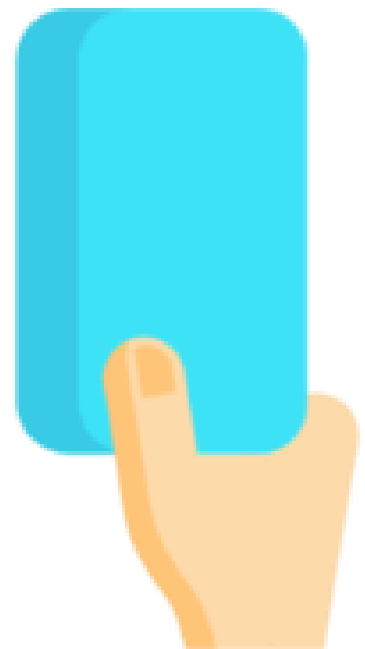


The screenshot shows the GitHub repository page for `cnr-isti-vclab/meshlab`. The repository is public and has 150 watchers, 686 forks, and 3.3k stars. It contains 124 issues, 2 pull requests, and 10,254 commits. The repository is licensed under GPL-3.0. The main branch is selected, and the repository contains several folders and files, including `.github`, `docs`, `resources`, `sample`, `scripts`, `src`, `textures`, `unsupported`, and `.gitignore`. The repository is described as "The open source mesh processing system" and includes a link to `www.meshlab.net`. The repository also includes a README, a GPL-3.0 license, and 3.3k stars.

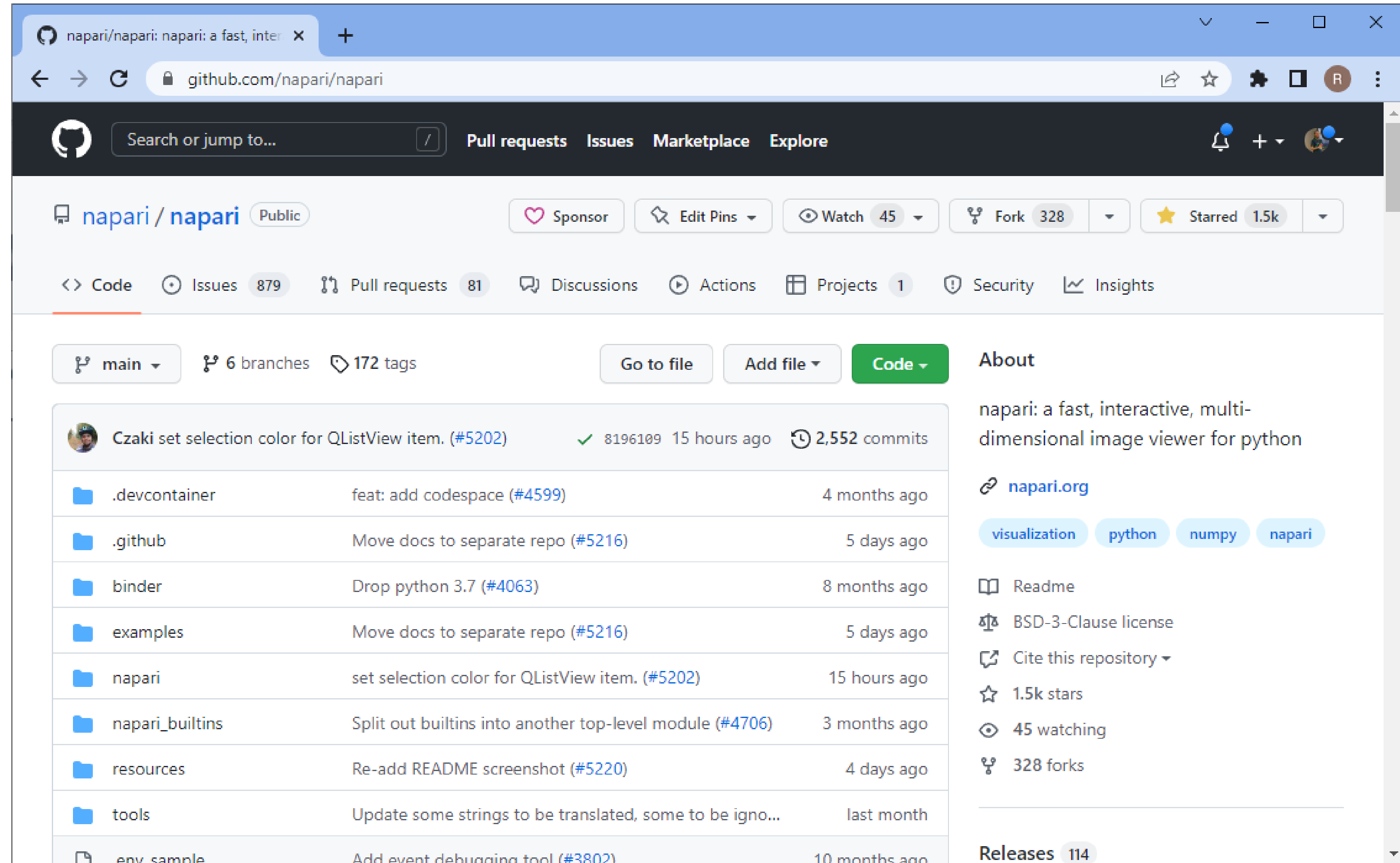
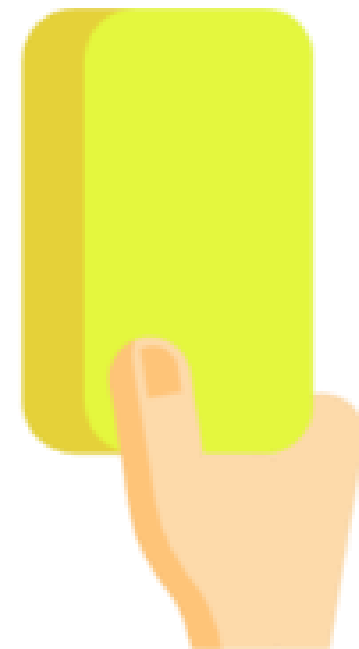
Quiz

May I reuse code from this repository in my own GPL-licensed work?

No



Yes



The screenshot shows the GitHub repository page for `napari/napari`. The repository is public and has 1.5k stars, 328 forks, and 45 watchers. It is licensed under BSD-3-Clause. The repository contains several folders and files, including `.devcontainer`, `.github`, `binder`, `examples`, `napari`, `napari_builtins`, `resources`, `tools`, and `env_sample`. The `napari` folder is currently selected, showing a commit by Czaki titled "set selection color for QListView item. (#5202)" 15 hours ago. The repository also has 879 issues, 81 pull requests, and 114 releases.

Take home message

If you share material (openly or not)

license it

and it'll be harder to steal it

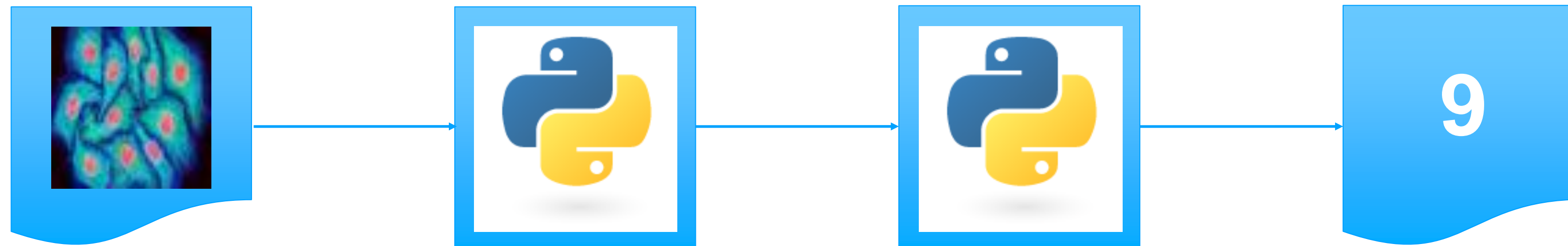
Python Algorithms: Conditions, loops, functions and custom libraries

Robert Haase

April 2023

Conditions

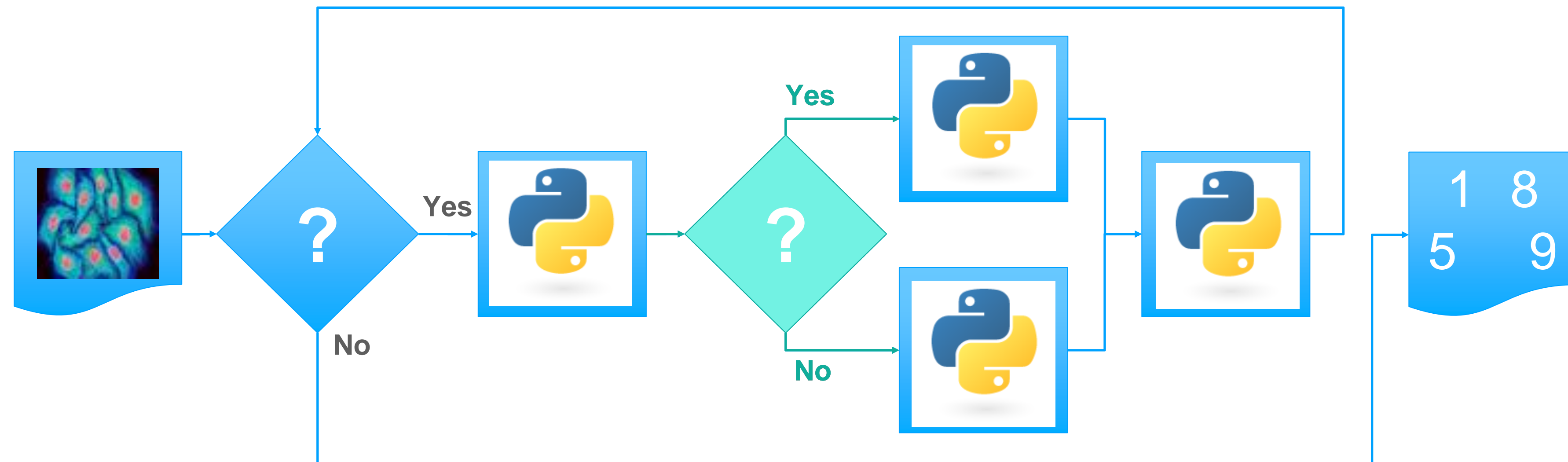
Data science workflows *rarely* look like this



Conditions

Data science workflows *rather* look like this

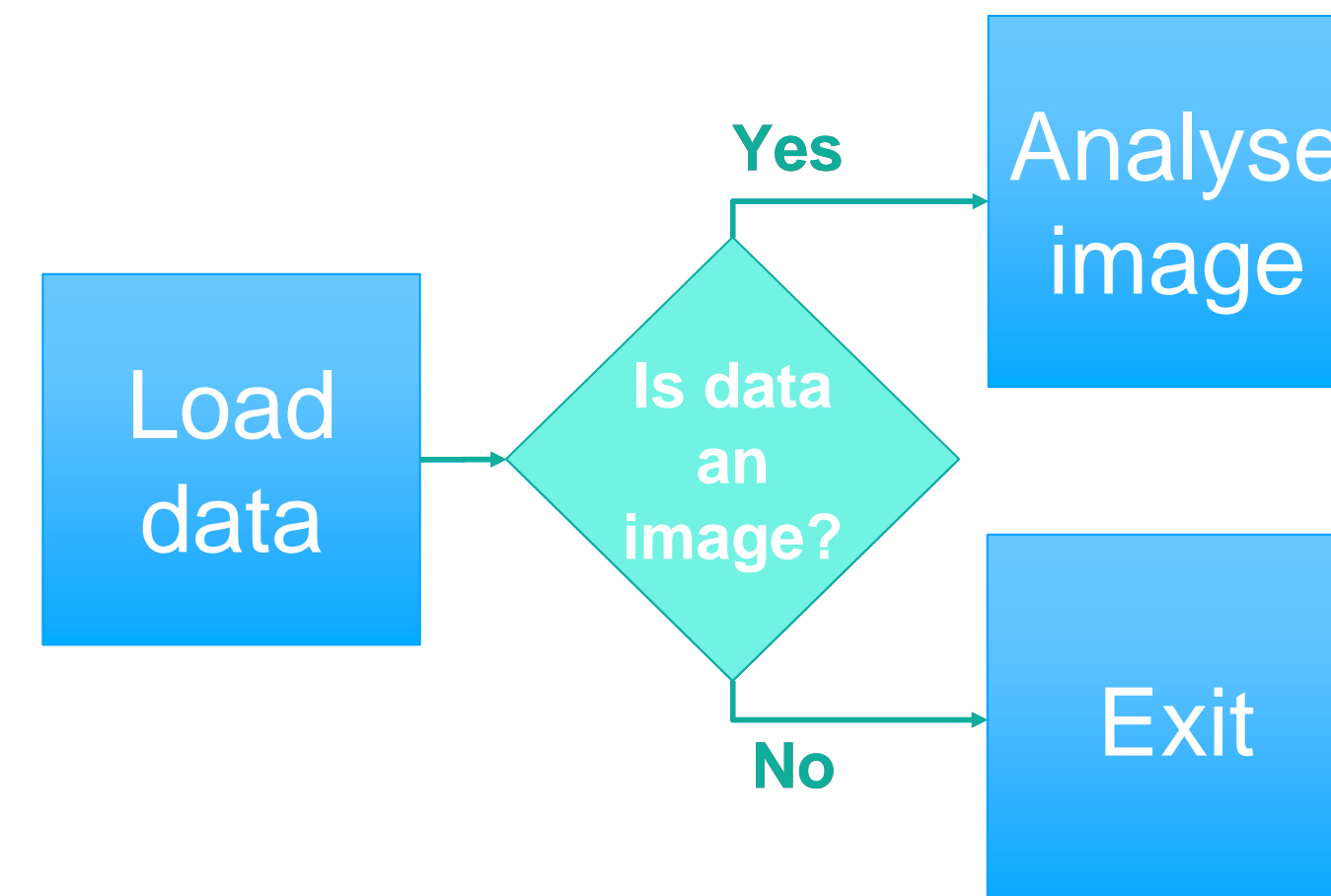
Conditional statement



Conditions

Conditional statements can be used to

- Check if pre-requisites are met
- Check if data has the right format
- Check if processing results are within an expected range
- Check for errors

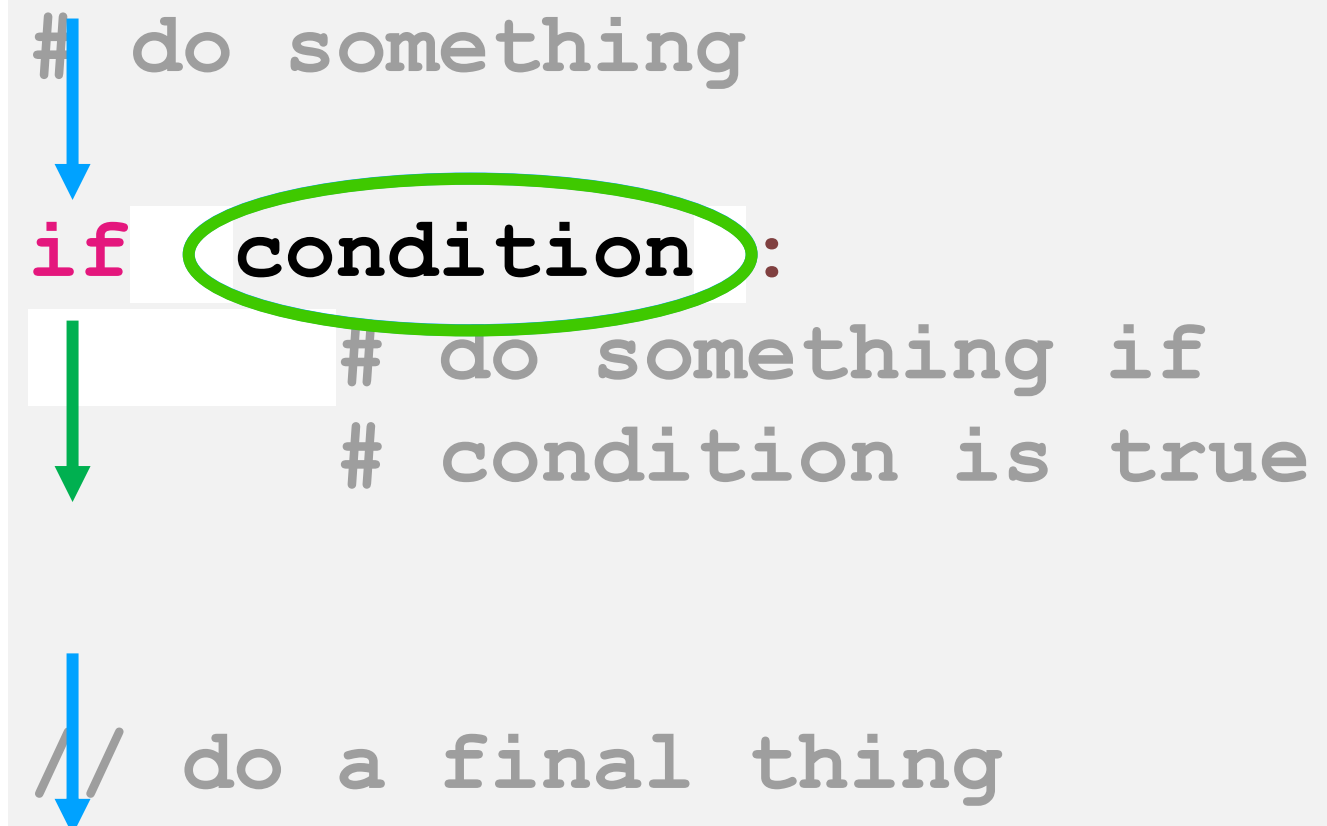


If-statement

Depending on a condition, some lines of code are executed or not.

```
# do something
if condition:
    # do something if
    # condition is true

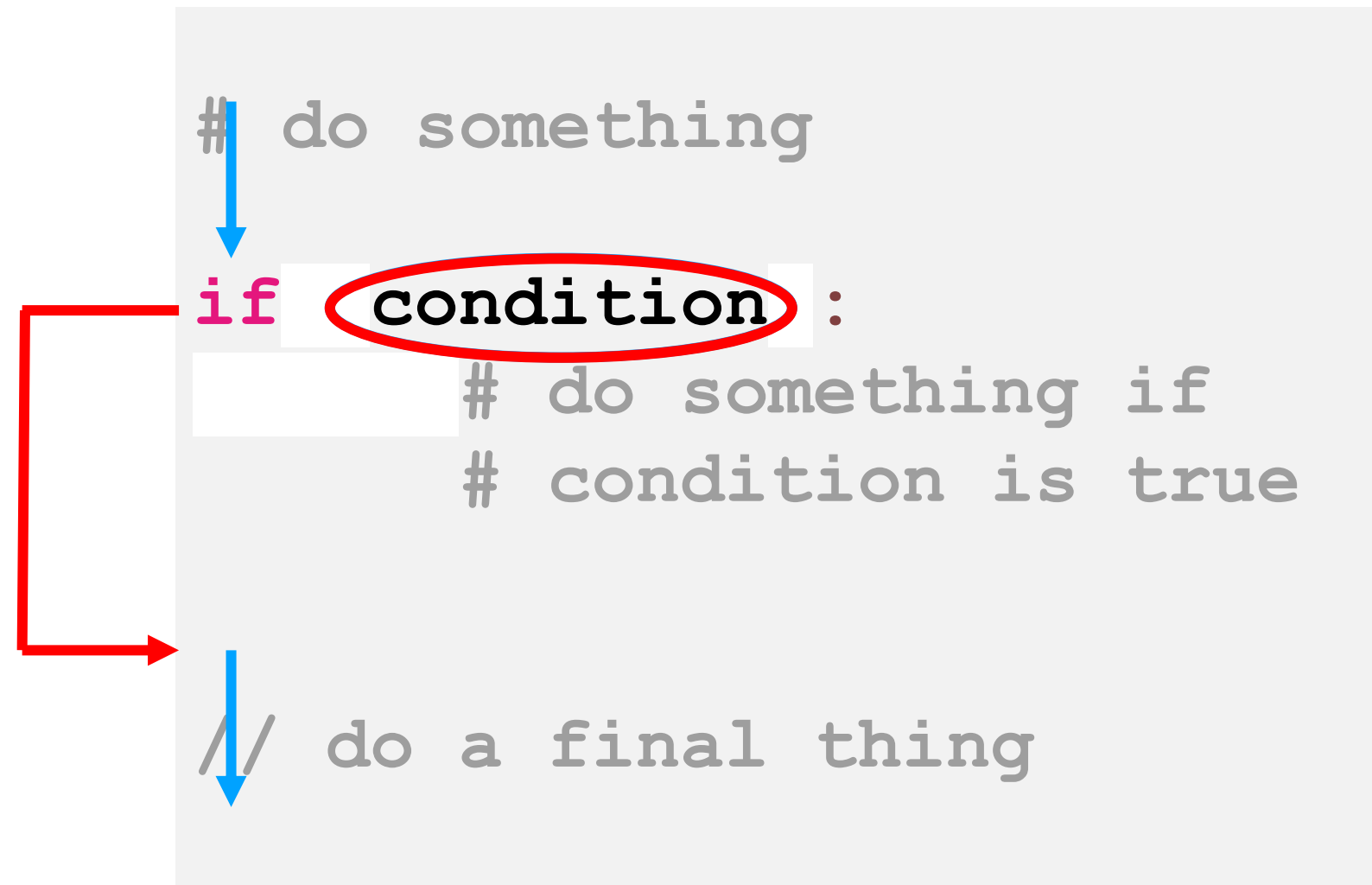
// do a final thing
```



If-statement

Depending on a condition, some lines of code are executed or not.

```
# do something
if condition :
    # do something if
    # condition is true
// do a final thing
```



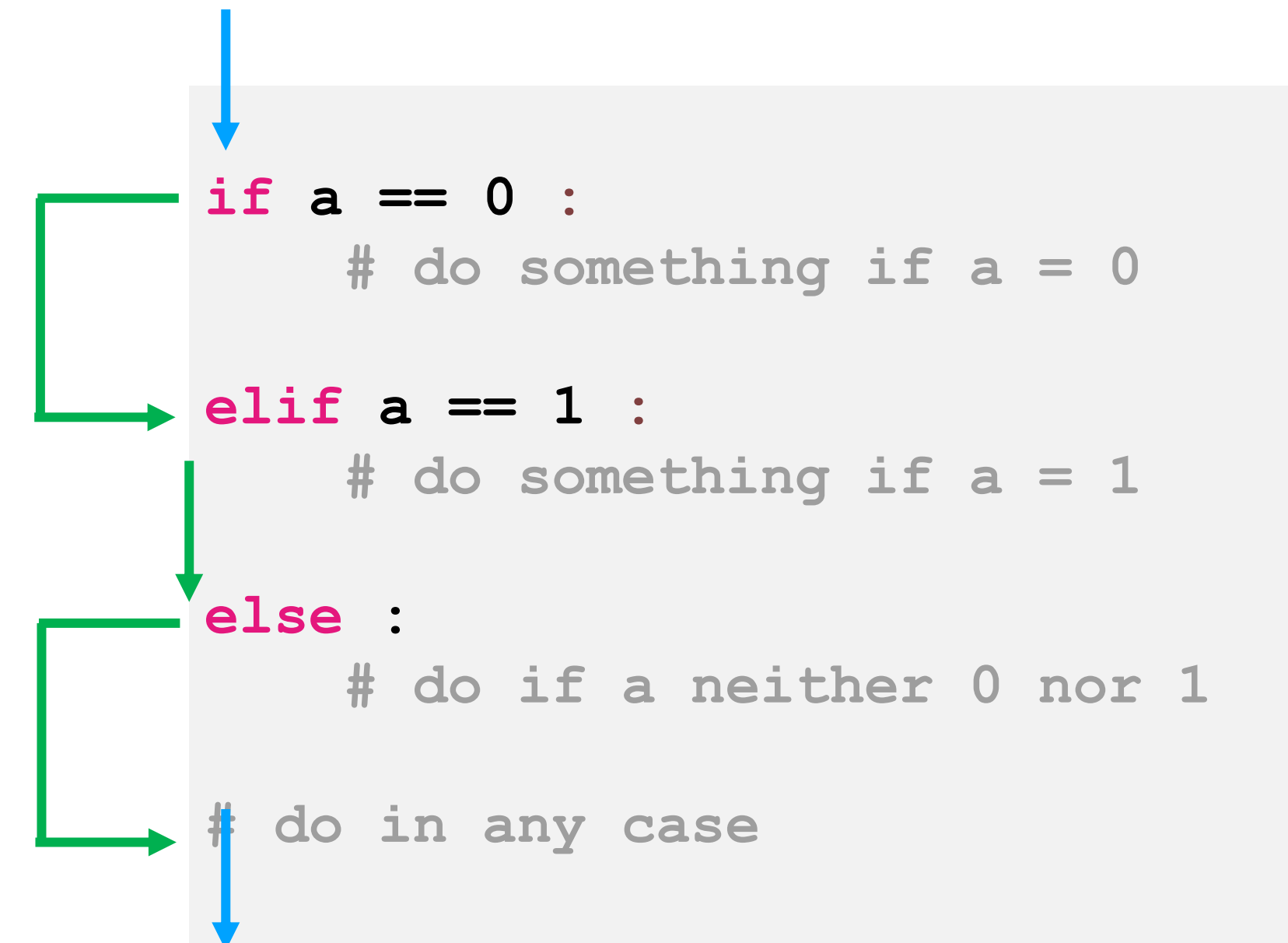
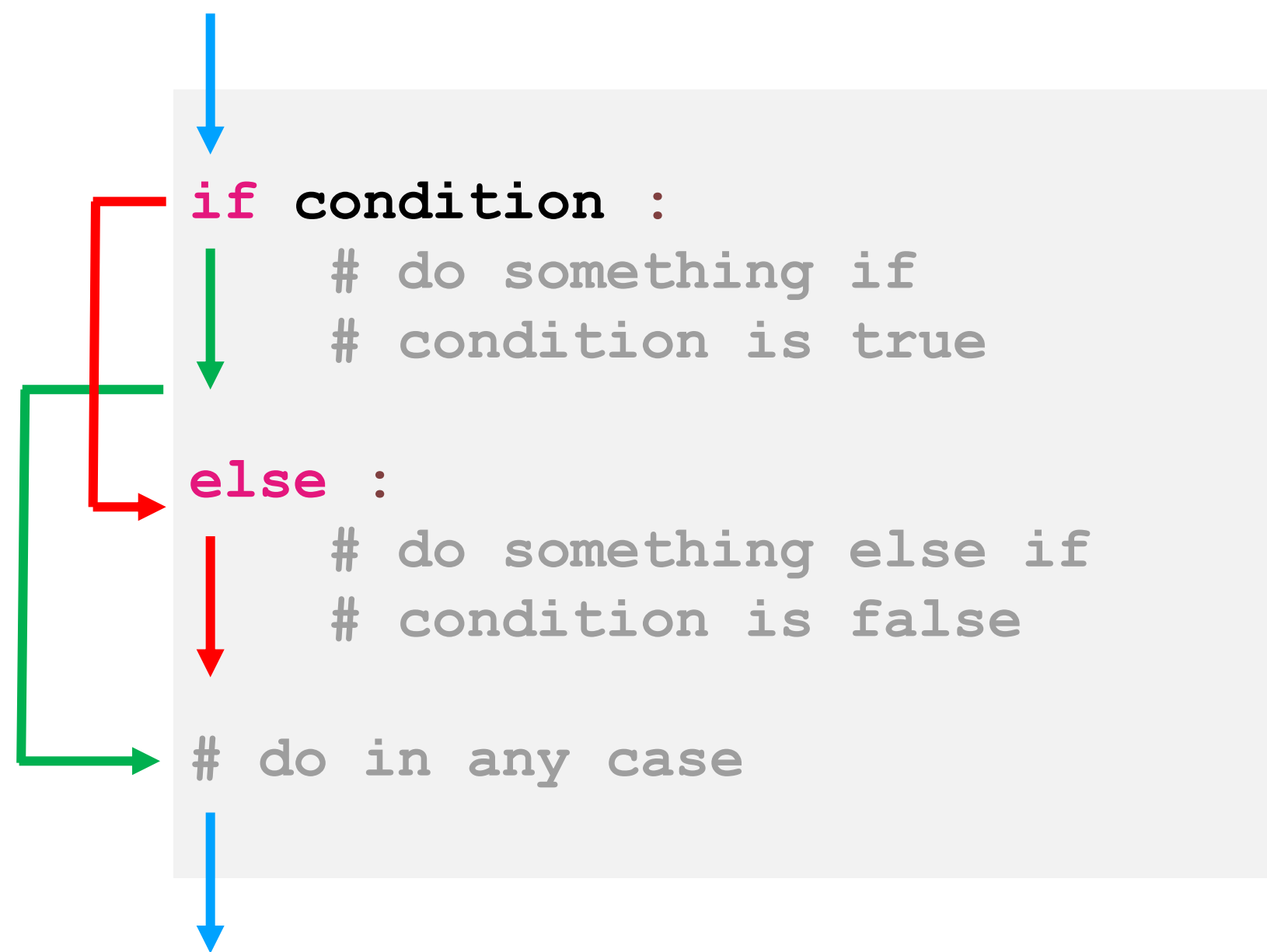
If-statement

The **if** / **elif** / **else** statement allows to program alternatives.

Depending on conditions, only one block is computed

Indentation is used to mark where a block starts and ends.

Indentation helps reading blocks,



If-statement

Comparison operators always have True (1) or False (0) as results.

```
# initialise program
quality = 99.5

# evaluate result
if quality > 99.9 :
    print("Everything is fine.")
else :
    print("We need to improve!")
```

```
In [1]: a = 4
if a = 5:
    print("Hello world")

File "<ipython-input-1-13fb587c9332>", line 3
    if a = 5:
        ^
SyntaxError: invalid syntax
```

Note: These are two equal signs!

Operator	Description	Example
<, <=	smaller than, smaller or equal to	a < b
>, >=	greater than, greater or equal to	a > b
==	equal to	a == b
!=	not equal to	a != 1

Combined conditions

Logic operators always take conditions as operands and result in a condition.

- **and**
- **or**
- **not**

Also combined conditions can be either True (1) or False (0).

```
# initialise program
quality = 99.9
age = 3

if quality >= 99.9 and age > 5 :
    print("The item is ok.")
```

```
# initialise program
quality = 99.9

if not quality < 99.9 :
    print("The item is ok.")
```

Conditions with arrays

Checking contents of lists can be done intuitively using the `in` statement

```
# initialise program
my_list = [1, 5, 7, 8]
item = 3

if item in my_list :
    print("The item is in the list.")
else :
    print("There is no", item, "in", my_list )
```

```
# initialise program
my_list = [1, 5, 7, 8]
item = 3

if item not in my_list :
    print("There is no", item, "in", my_list )
else :
    print("The item is in the list.")
```


Readable code

- Every command belongs on its own line
- Insert empty lines to separate important processing steps
- Put spaces between operators and operands, because:

This is easier to read than that, or isn't it?
- Indent every conditional block (if/else) using the TAB key

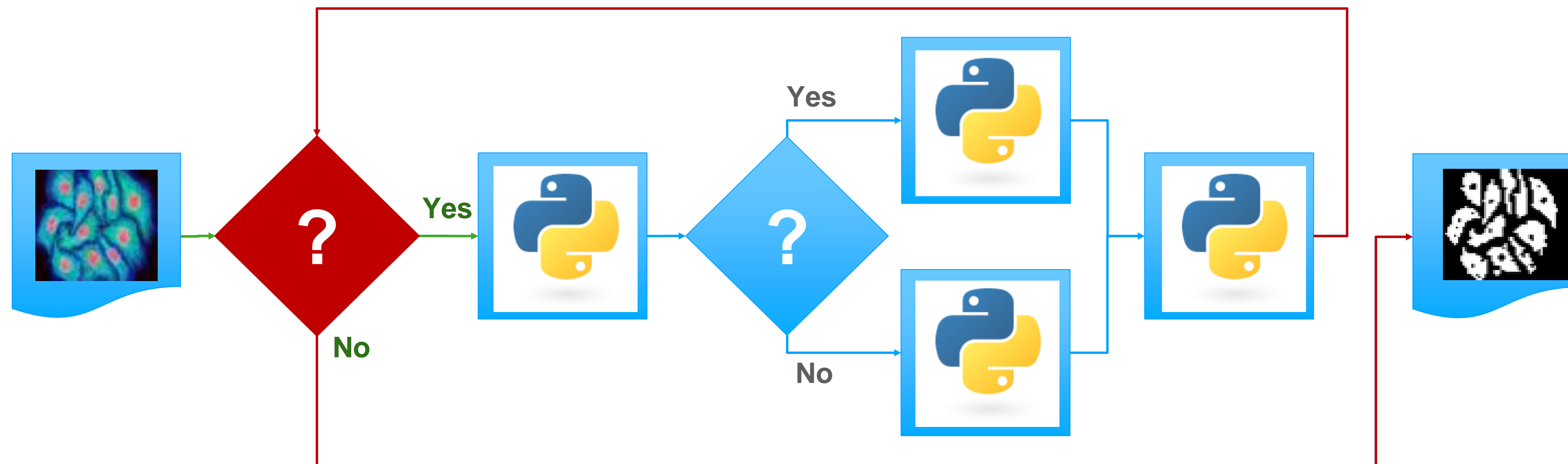
```
# initialise program
a = 5
b = 3
c = 8

# execute algorithm
d = (a + b) / c

# evaluate result
if a == 5 :
    print("Yin")
else :
    print("Yang")
```

Loops

To repeat actions, you run code in loops

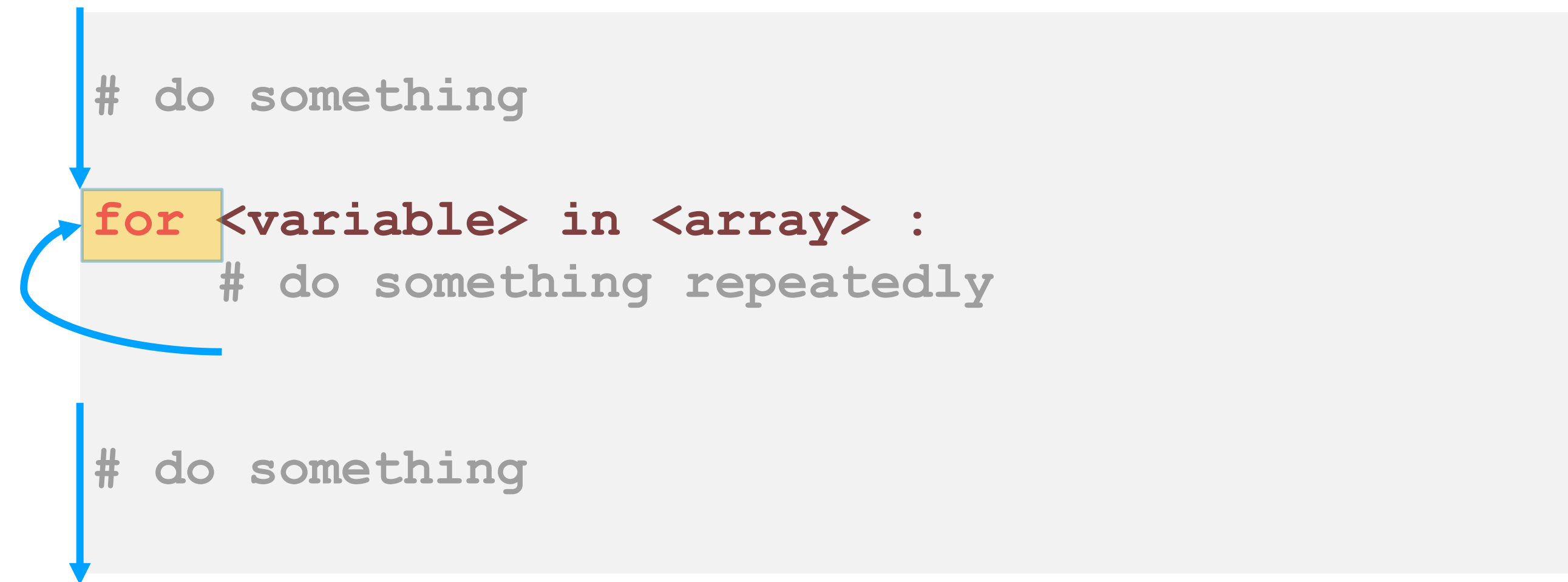


Loop
statement

Loops

The `for` statement allows us to execute some lines of code *for* several times, typically for all items in an array-like thing (lists, tuples, images)

```
# do something
for <variable> in <array> :
    # do something repeatedly
# do something
```



For-loops

Example list: `range(start, stop, step)`

```
for i in range(0, 5):  
    print(i)
```

0
1
2
3
4

```
▶ animal_set = ["Cat", "Dog", "Mouse"]  
  
for animal in animal_set:  
    print(animal)
```

Cat
Dog
Mouse

For-loops

- Indentation *means* combining operations to a block

```
for i in range(0, 5):  
print(i)
```

```
File "<ipython-input-15-59c457ae0ac9>", line 3  
print(i)  
  ^
```

IndentationError: expected an indented block

Don't forget
to indent!

- Colon necessary

```
for i in range(0, 5)  
print(i)
```

```
File "<ipython-input-13-23157c0ed137>", line 2  
for i in range(0, 5)  
  ^
```

SyntaxError: invalid syntax

Don't forget
the colon!

Generating arrays within for-loops

There is a long and a short way for creating arrays with numbers.

```
# we start with an empty list
numbers = []

# and add elements
for i in range(0, 5):
    numbers.append(i * 2)

print(numbers)
```

```
numbers = [i * 2 for i in range(0, 5)]
print(numbers)

[0, 2, 4, 6, 8]
```

Generating arrays within for-loops

Also a combination with the if-statement is possible

```
# we start with an empty list
numbers = []

# and add elements
for i in range(0, 5):
    # check if the number is odd
    if i % 2:
        numbers.append(i * 2)

print(numbers)
```

[2, 6]

```
numbers = [i * 2 for i in range(0, 5) if i % 2]
print(numbers)
```

[2, 6]

Listing files in a folder

Common use-case: do something with all *image* files in a folder

```
for file in file_list:  
    if file.endswith(".tif"):  
        print(file)
```

```
banana0002.tif  
banana0003.tif  
banana0004.tif  
banana0005.tif  
banana0006.tif
```

```
image_file_list = [file for file in file_list if file.endswith(".tif")]  
image_file_list
```

```
['banana0002.tif',  
'banana0003.tif',  
'banana0004.tif',  
'banana0005.tif',  
'banana0006.tif',
```


While-loops

While loops keep executing indented code as long as a **condition** is met:

```
number = 1024  
  
while (number > 1):  
    number = number / 2  
    print(number)
```

Works the same as
with the **if**
statement

```
512.0  
256.0  
128.0  
64.0  
32.0  
16.0  
8.0  
4.0  
2.0  
1.0
```

Executing loops

Using the `break` statement, you can leave a loop

```
number = 1024

while (True):
    number = number / 2
    print(number)


    if number < 1:
        break
```

```
512.0
256.0
128.0
64.0
32.0
16.0
8.0
4.0
2.0
1.0
0.5
```

Skipping iterations

The `continue` statement allows to skip iterations

```
for i in range(0, 10):  
    if i >= 3 and i <= 6:  
        continue  
    print(i)
```



0
1
2
7
8
9

Functions

- In case repetitive tasks appear that cannot be handled in a loop, custom functions are the way to go.
- Functions allow to re-use code in different contexts.
- Indentation is crucial.
- Functions must be defined before called

Definition:

```
def sum_numbers(a, b):  
    result = a + b  
    return result
```

name (parameters)

body

return statement
(optional)

Call:

```
c = sum_numbers(4, 5)  
print(c)
```

9

```
sum_numbers(5, 6)
```

11

```
sum_numbers(3, 4)
```

7

Functions

In case repetitive tasks appear that cannot be handled in a loop, custom functions are the way to go.

Functions allow to re-use code in different contexts.

Indentation is crucial.

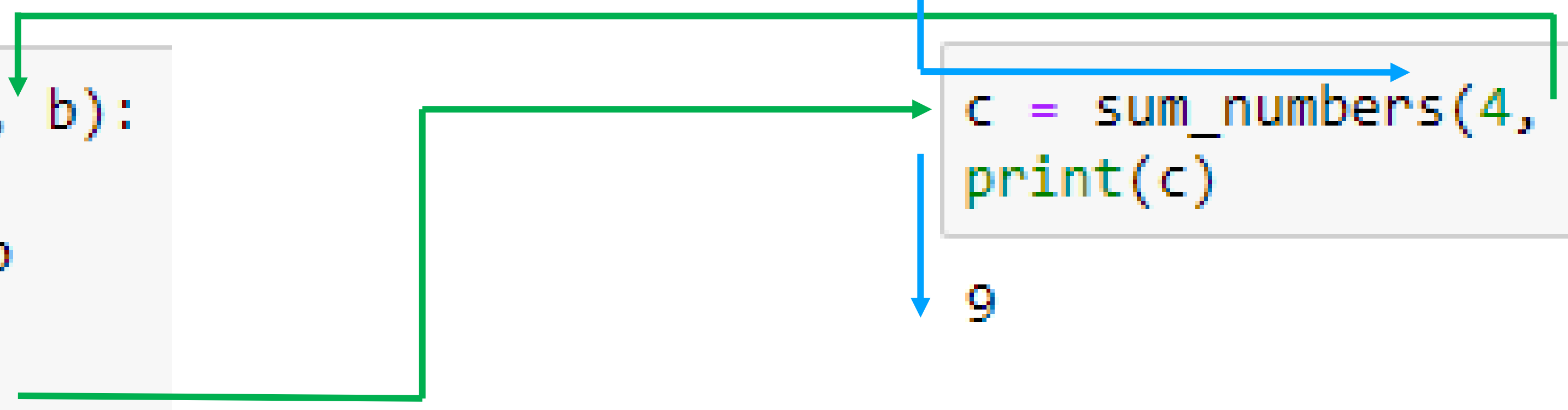
Functions must be defined before called

Definition

```
def sum_numbers(a, b):  
    result = a + b  
    return result
```

Call

```
c = sum_numbers(4, 5)  
print(c)  
9
```



Functions

Document your functions to keep track of what they do.

Describe what the functions does and what the parameters are meant to be

```
def square(number):  
    '''  
    Squares a number by multiplying it with itself and returns its result.  
    '''  
  
    return number * number
```

- You can then later print the *documentation* if you can't recall how a function works.

```
square?
```

Signature: square(number)

Docstring: Squares a number by multiplying it with itself and returns its result.

Libraries

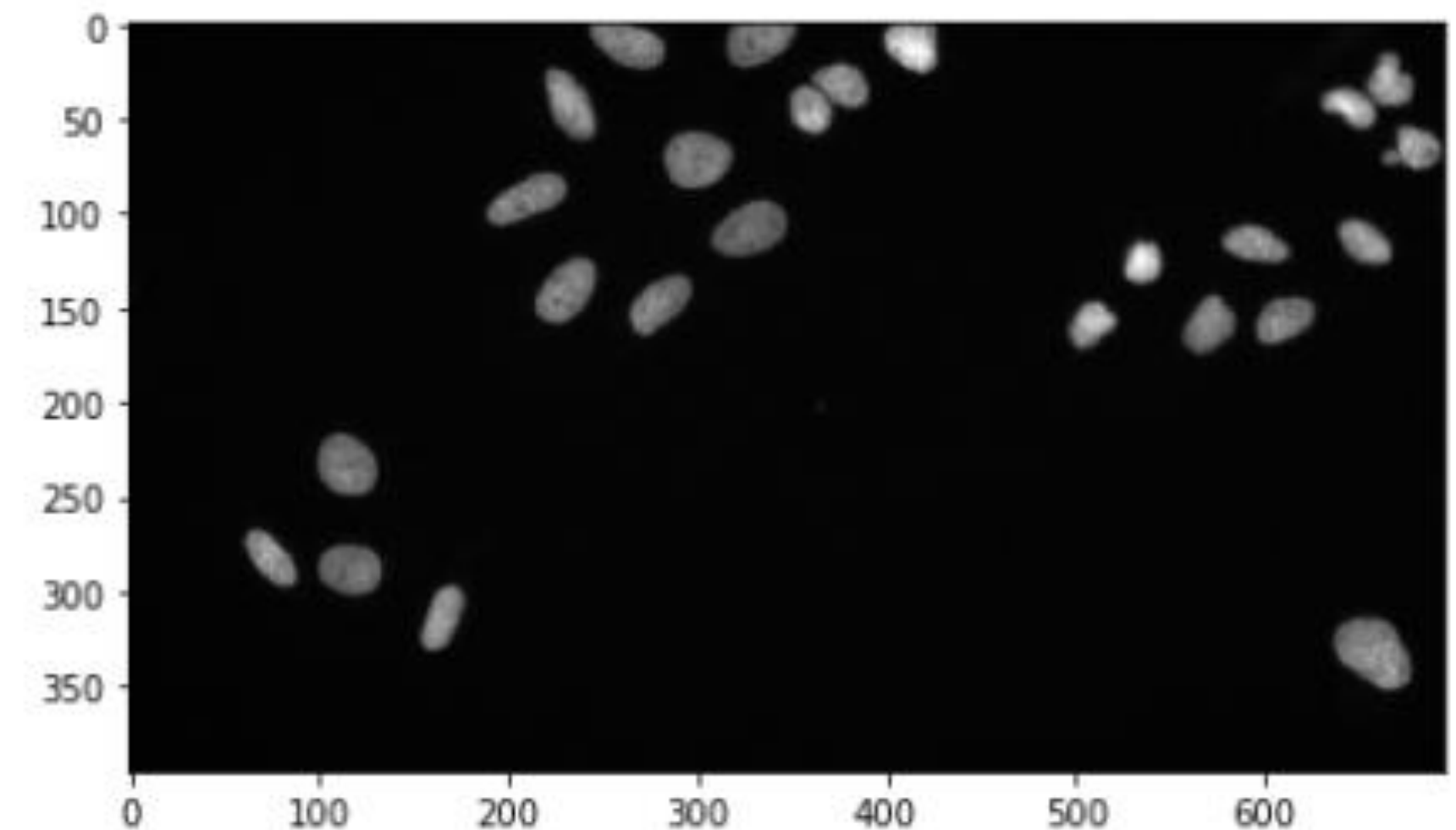
- The import statement allows to use functions provided by others.
- Commonly put at the beginning of a notebook or script to make sure everything is installed.

```
from skimage.io import imread
import matplotlib.pyplot as plt
import pyclesperanto_prototype as cle
```

After `cle` has
been imported...

... it can be used

```
cle.imshow(input_crop, plot=axes[1])
```



Libraries

For re-using functions between notebooks / projects, use libraries. -> Sustainability

Simple python libraries are .py files containing multiple functions.

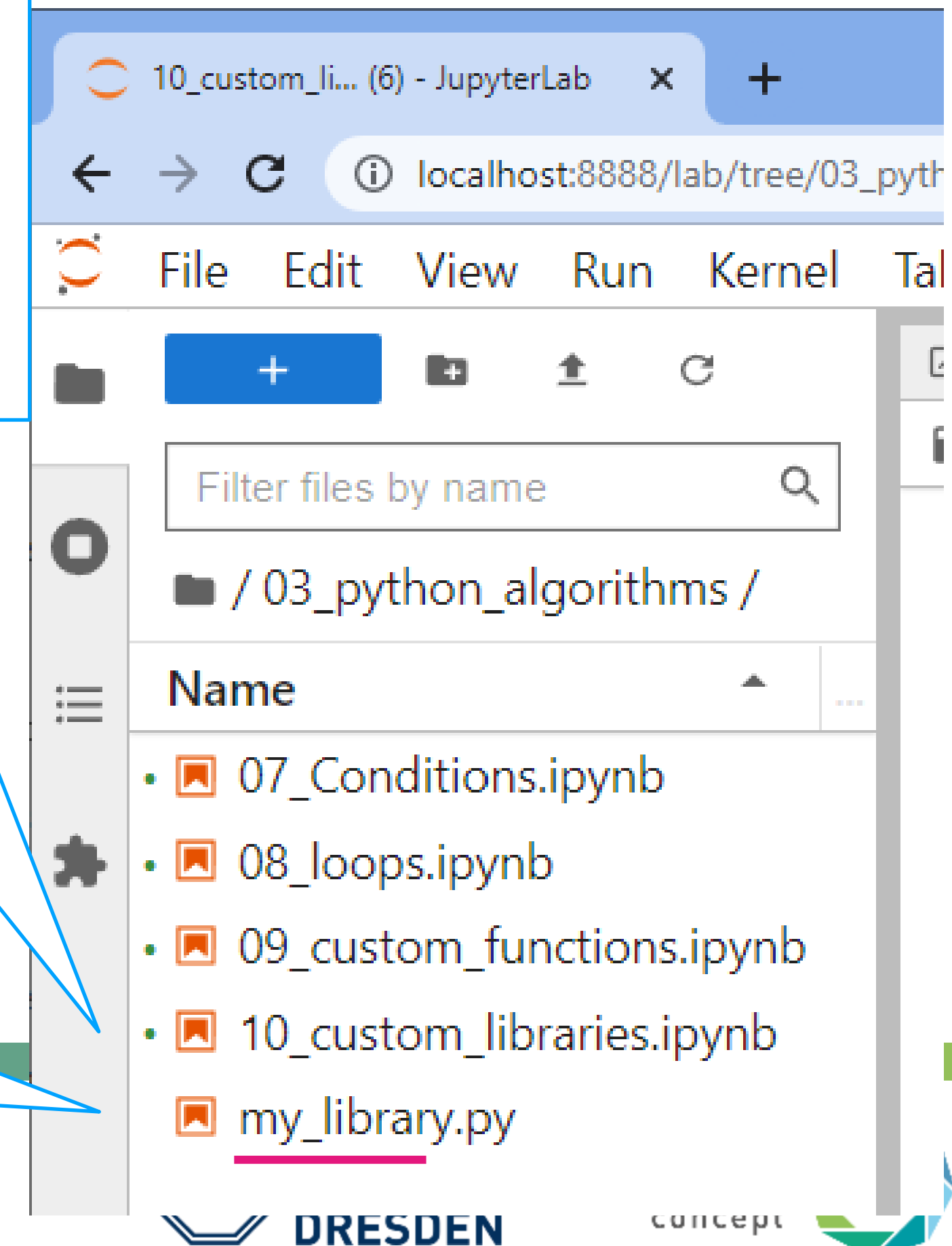
The **import** statement allows you to import python files from the same folder.

```
from my_library import square
```

```
square(5)
```

```
25
```

```
1 def wuzzle(number):  
2     ...  
3     The wuzzle function manipulates a number in a magic way and returns the result.  
4     ...  
5  
6     import math  
7     return math.sqrt(number * 1.2)  
8  
9 def square(number):  
10    ...  
11    Squares a number by multiplying it with itself and returns its result.  
12    ...  
13  
14    return number * number
```



10_custom_li... (6) - JupyterLab x +

localhost:8888/lab/tree/03_pyth

File Edit View Run Kernel Tal

Filter files by name

/ 03_python_algorithms /

Name

- 07_Conditions.ipynb
- 08_loops.ipynb
- 09_custom_functions.ipynb
- 10_custom_libraries.ipynb
- my_library.py

DRESDEN concept

Outlook: The power of Python

With Python, you can automate many tedious tasks. Example: Downloading files from the owncloud.

Build a login form

```
server_widget = widgets.Text(value='https://cloudstore.zih.tu-dresden.  
username_widget = widgets.Text(description='Username:')  
password_widget = widgets.Password(description='Password')  
  
widgets.VBox([server_widget, username_widget, password_widget])
```

Server	<input type="text" value="https://cloudstore.zih.tu-dresden.c"/>
Username:	<input type="text" value="roha044c"/>
Password	<input type="password"/>

Log in

```
oc = owncloud.Client(server_widget.value)  
oc.login(username_widget.value, password_widget.value)
```

List all files in the owncloud

```
# enter a folder on the owncloud drive that exists.  
remote_folder = "/"  
  
for f in oc.list(remote_folder):  
    print (f.path)
```

```
/BiAPoL/  
/Documents/  
/Nextcloud Manual.pdf  
/Nextcloud intro.mp4  
/Nextcloud.png  
/Photos/  
/Projects/  
/Shared/  
/Software/
```

Download a file

```
# enter the source file here  
remote_source_file = '/Nextcloud Manual.pdf'  
# enter the destination  
local_file = 'Nextcloud Manual.pdf'  
  
oc.get_file(remote_path=remote_source_file,  
            local_file=local_file)
```

True

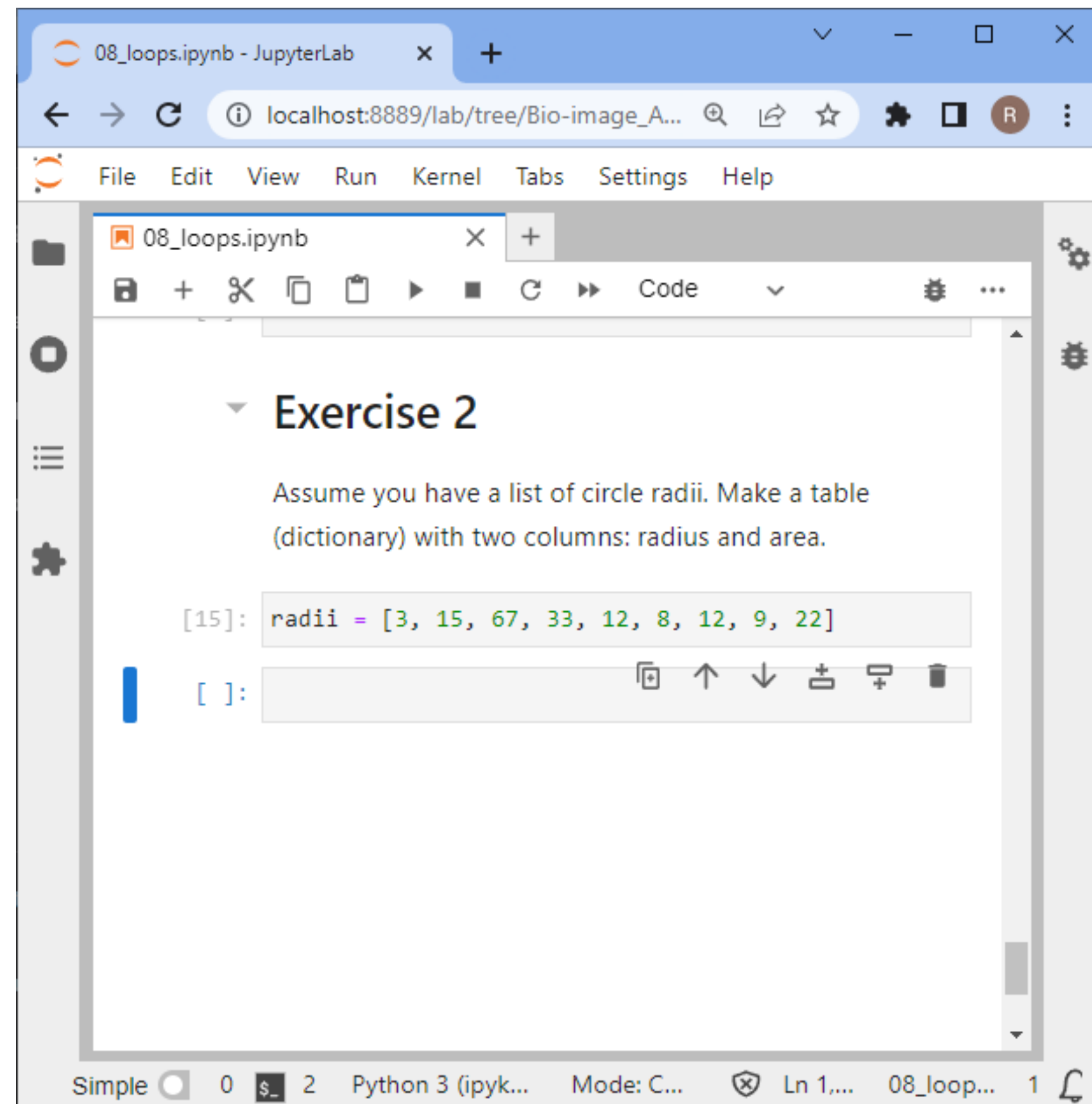
The PDF is now located in the same folder as this notebook.

Take care: You can also **DELETE** all files in an owncloud folder using similar code

NEVER save your password in Python code!

Outlook: The power of AI

Feel free to use artificial intelligence during the exercises. Play with it, learn how to *exploit it* best. Consider: It *lies* from time to time and during the exam it can't help you.



08_loops.ipynb - JupyterLab

localhost:8889/lab/tree/Bio-image_A...

File Edit View Run Kernel Tabs Settings Help

08_loops.ipynb

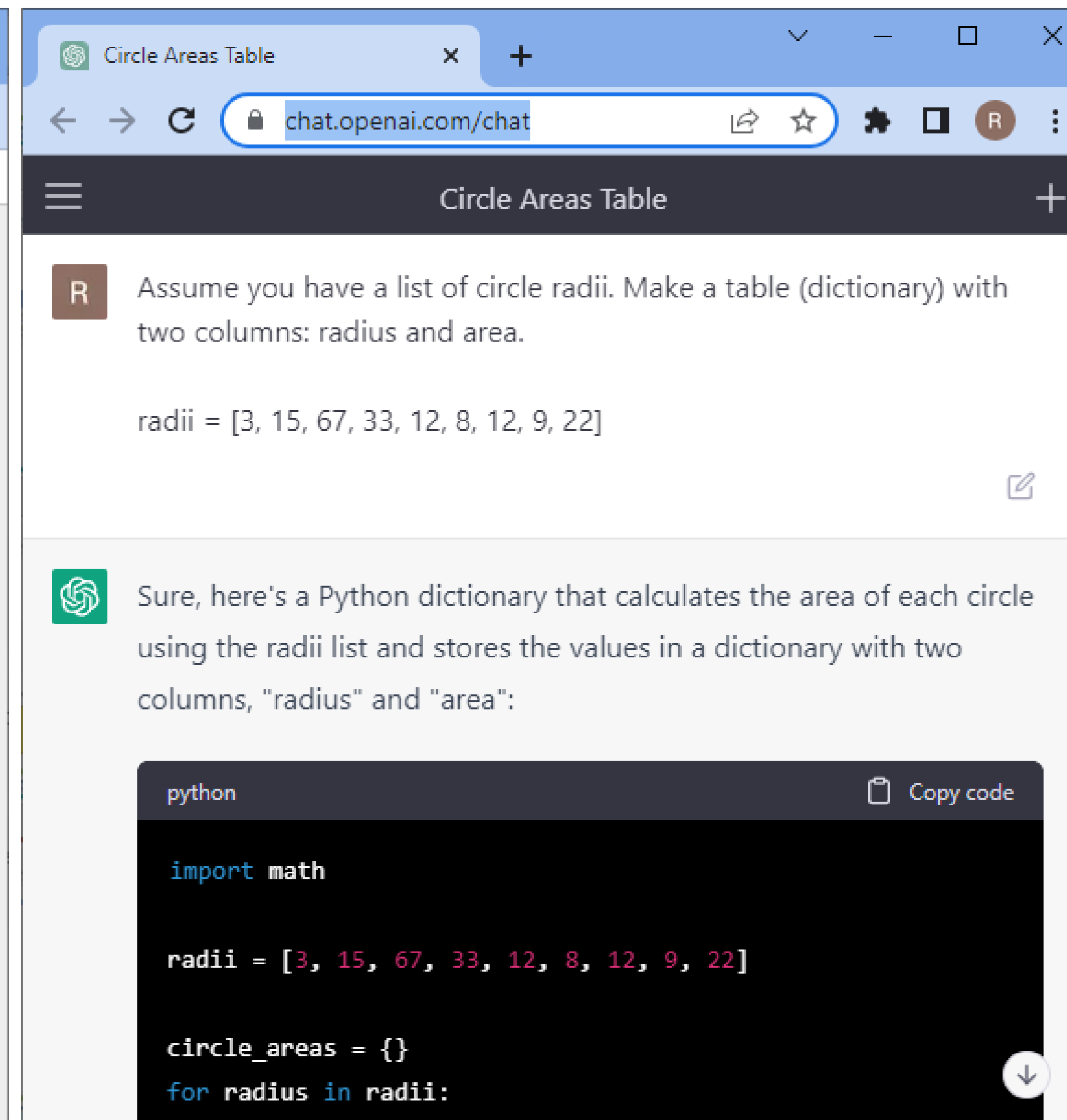
Exercise 2

Assume you have a list of circle radii. Make a table (dictionary) with two columns: radius and area.

```
[15]: radii = [3, 15, 67, 33, 12, 8, 12, 9, 22]
```

```
[ ]:
```

Simple 0 2 Python 3 (ipyk... Mode: C... Ln 1,000 08_loop... 1



Circle Areas Table

chat.openai.com/chat

Circle Areas Table

R Assume you have a list of circle radii. Make a table (dictionary) with two columns: radius and area.

```
radii = [3, 15, 67, 33, 12, 8, 12, 9, 22]
```

Sure, here's a Python dictionary that calculates the area of each circle using the radii list and stores the values in a dictionary with two columns, "radius" and "area":

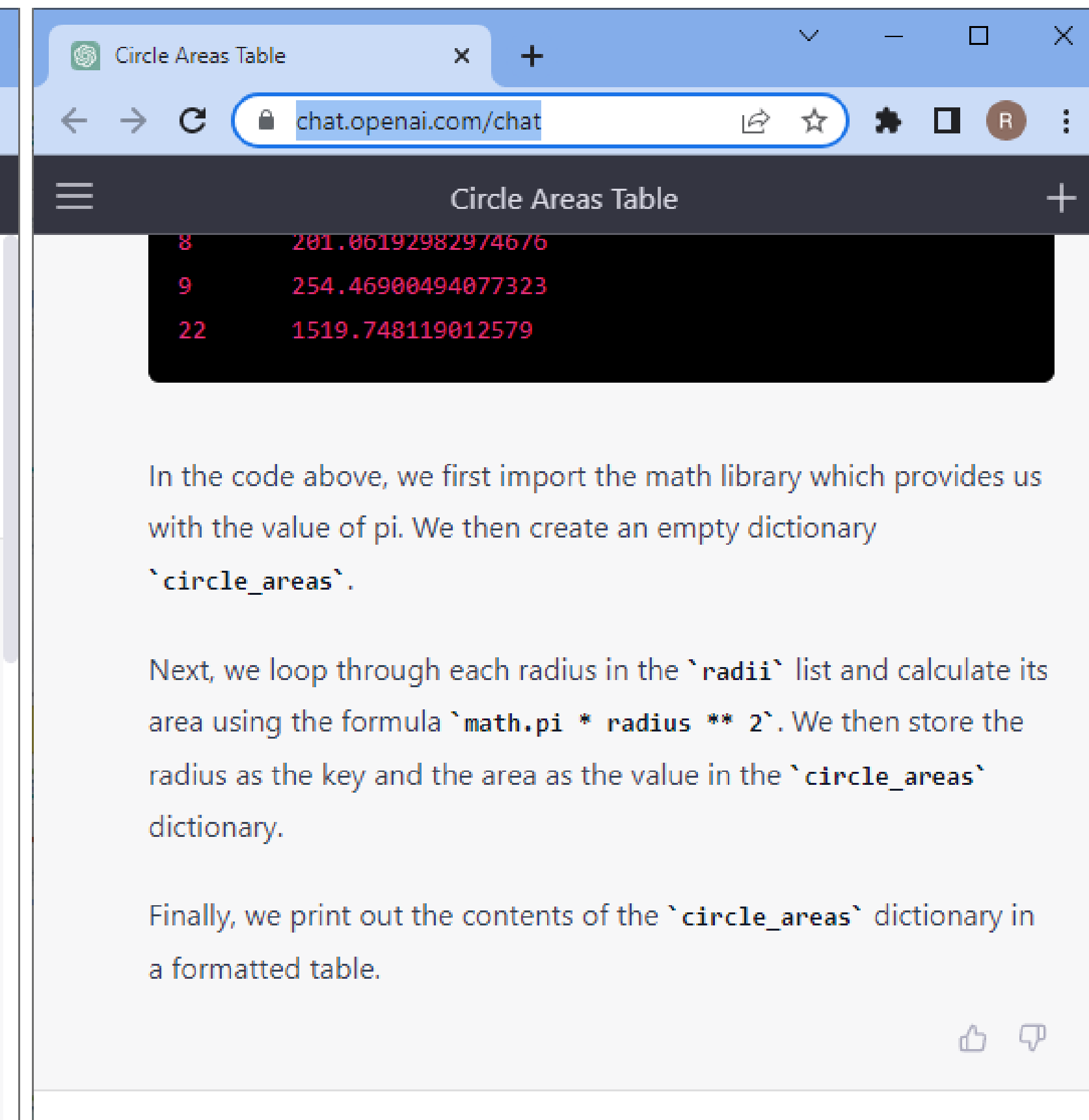
```
python Copy code
```

```
import math
```

```
radii = [3, 15, 67, 33, 12, 8, 12, 9, 22]
```

```
circle_areas = {}
```

```
for radius in radii:
```



Circle Areas Table

chat.openai.com/chat

Circle Areas Table

```
8 201.06192982974676
```

```
9 254.46900494077323
```

```
22 1519.748119012579
```

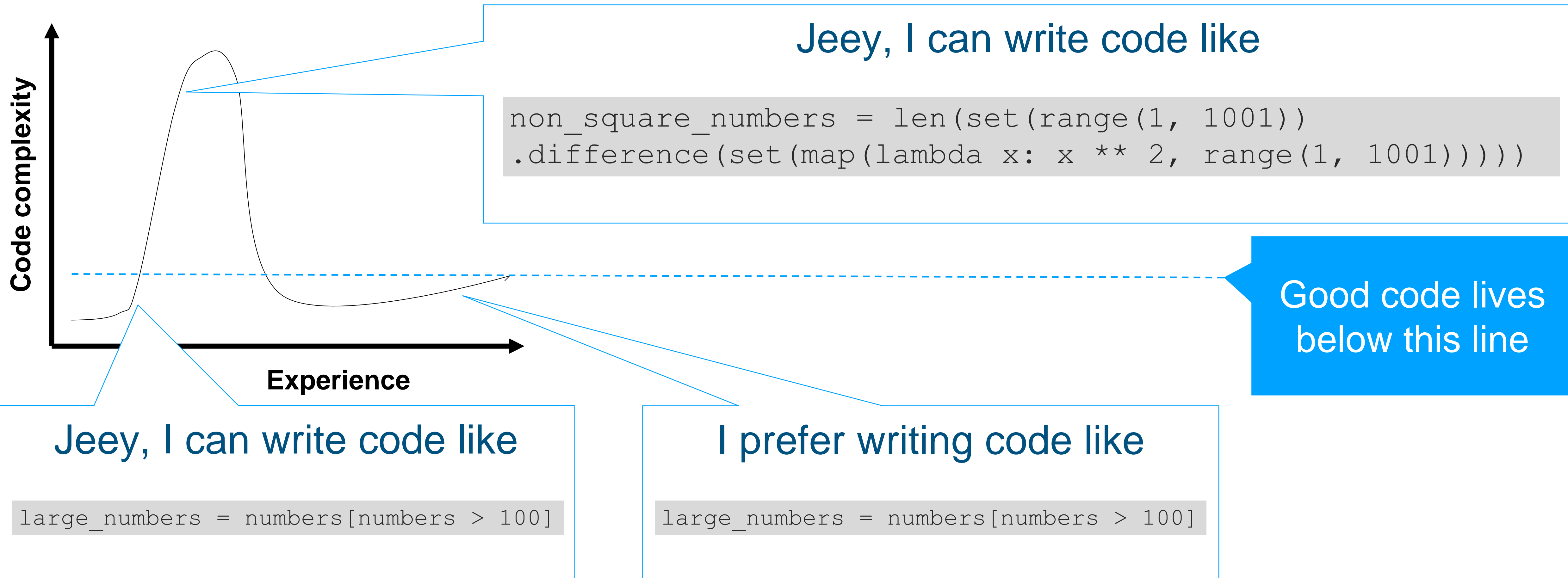
In the code above, we first import the math library which provides us with the value of pi. We then create an empty dictionary `circle_areas`.

Next, we loop through each radius in the `radii` list and calculate its area using the formula `math.pi * radius ** 2`. We then store the radius as the key and the area as the value in the `circle_areas` dictionary.

Finally, we print out the contents of the `circle_areas` dictionary in a formatted table.

Software quality indicators

Interpretation of “High quality code” changes with experience



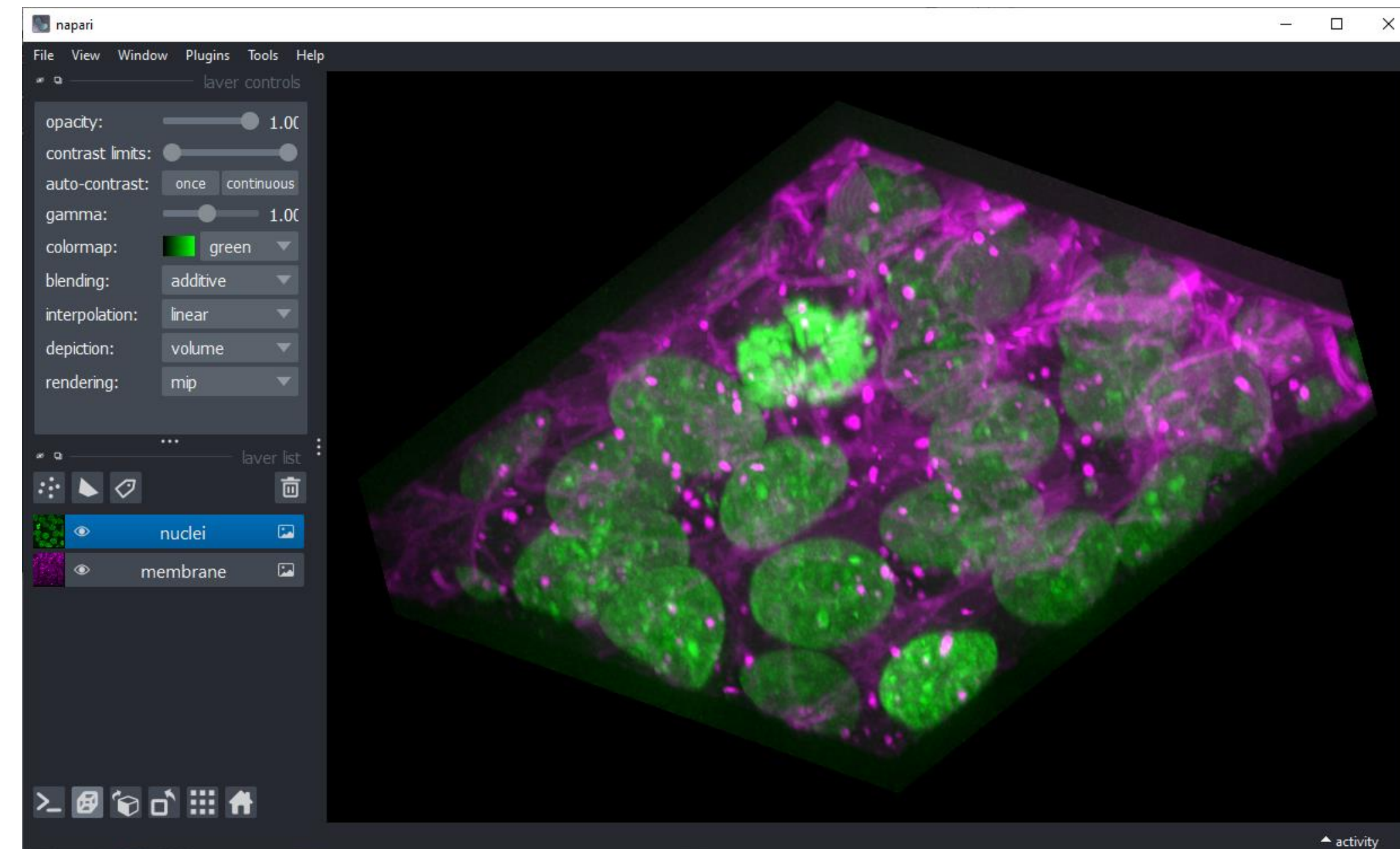
Summary

Today you learned:

- Document what *dependencies* you use!
- Sharing / licensing terminology
 - Levels of openness
 - Copyright holder / Author / Publisher / Licensee
 - FAIR principles (findable, accessible, interoperable, reusable)
- Python algorithms
 - Loops
 - Conditions
 - Functions
 - Libraries

Coming up next:

- Image processing
- Image filtering
- Napari

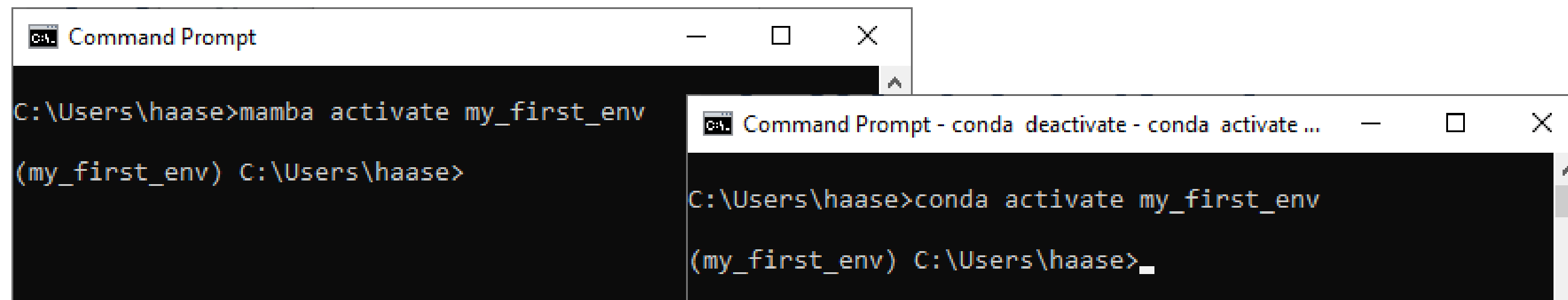


Exercises

- Check the `02_python_algorithms` folder online https://github.com/BiAPoL/Bio-image_Analysis_with_Python

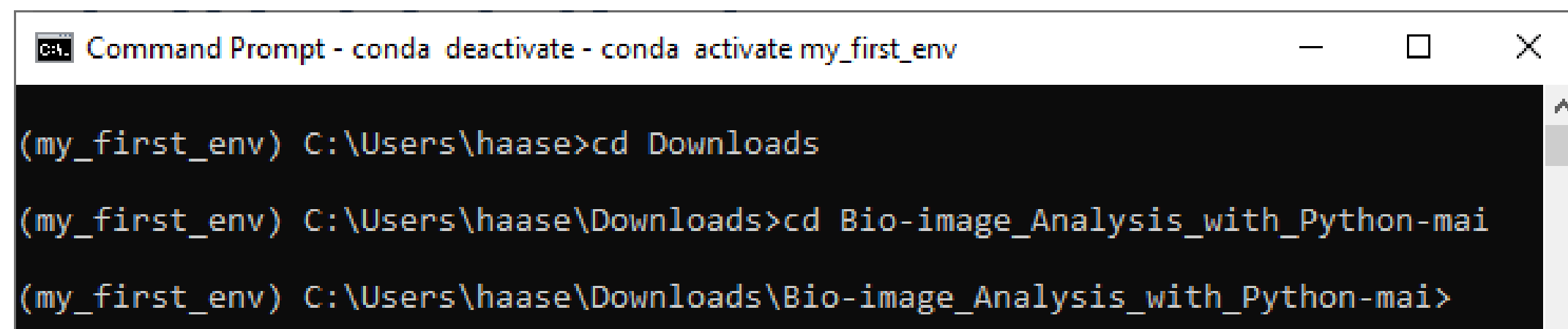
1. Start up a terminal
2. Activate the environment using `conda activate my_first_env`

4. Start up jupyter lab



```
C:\Users\haase>mamba activate my_first_env
(my_first_env) C:\Users\haase>
C:\Users\haase>conda activate my_first_env
(my_first_env) C:\Users\haase>_
```

3. Use the `cd` command to navigate to the exercise folder



```
(my_first_env) C:\Users\haase>cd Downloads
(my_first_env) C:\Users\haase\Downloads>cd Bio-image_Analysis_with_Python-mai
(my_first_env) C:\Users\haase\Downloads\Bio-image_Analysis_with_Python-mai>
```

