

thonny + py5

a python 3 environment for processing

processing + py5 ecosystem

villares/Resources-for-teaching-programming: Resources for teaching programming for artists, designers and architects — Mozilla Firefox

villares/Resources-for-teach x +

https://github.com/villares/Resources-for-teaching-programming

Search or jump to... Pull requests Issues Marketplace Explore

villares / **Resources-for-teaching-programming** Sponsor Unwatch 4 Unstar 40 Fork 4

<> Code Issues 4 Pull requests Actions Projects Wiki Security Insights

main 1 branch 0 tags Go to file Add file Code

villares Update README.md ✓ 34db1dc 11 days ago 133 commits

.github	Create FUNDING.yml	2 years ago
._layouts	Update default.html	13 months ago
.assets	Update style.css	2 years ago
.gitignore	css font sans-serif	4 years ago
I - Host platforms & languages.csv	Update I - Host platforms & languages.csv	3 years ago
II - Books & References.csv	Update II - Books & References.csv	5 years ago
README.md	Update README.md	11 days ago
._config.yml	yaml	4 years ago

☰ README.md

Resources for teaching programming

for artists, designers and architects (repository)

Extended table of Hosts, Platforms & Languages

- Drawing or 3D modeling software that embeds a scripting language on the user interface or allows automation with a very limited number of steps between programming and code execution.

About
Resources for teaching programming for artists, designers and architects
villares.github.io/resources-for-teaching-programming/
Readme

Releases
No releases published

Sponsor this project
<https://gumroad.com/villares>

Packages
No packages published

Contributors 2

- villares** Alexandre B A Villares
- bernhard** berin

<https://github.com/villares/Resources-for-teaching-programming>

Environments 1

villares/Resources-for-teaching-programming: Resources for teaching programming for artists, designers and architects — Mozilla Firefox

villares/Resources-for-teach x aparrish (Allison Parrish) x +

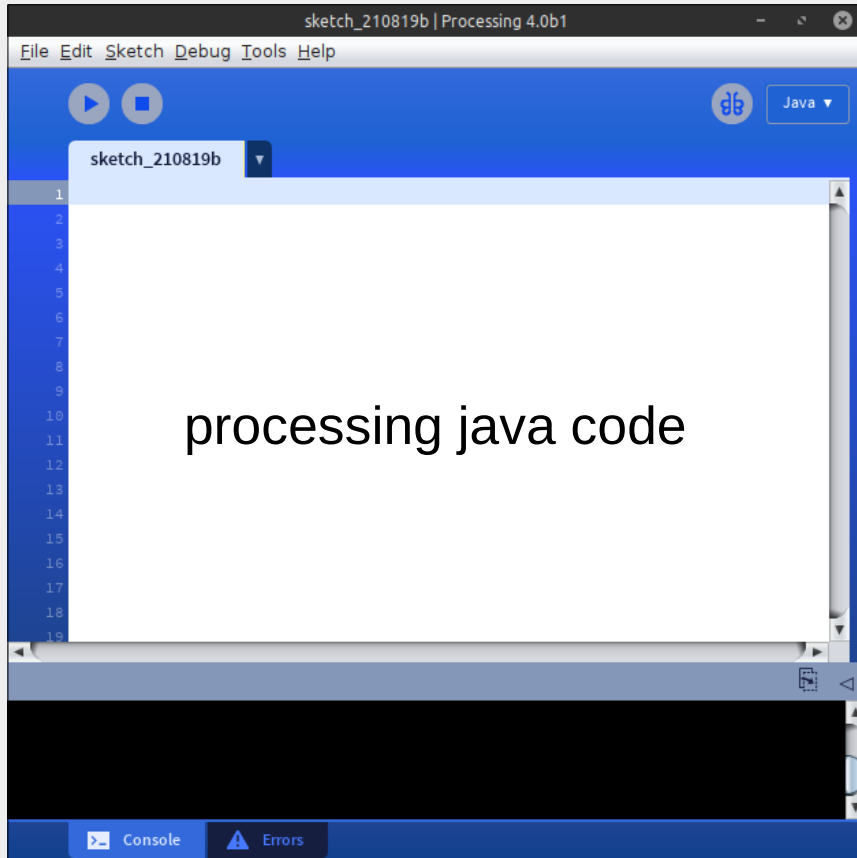
https://github.com/villares/Resources-for-teaching-programming

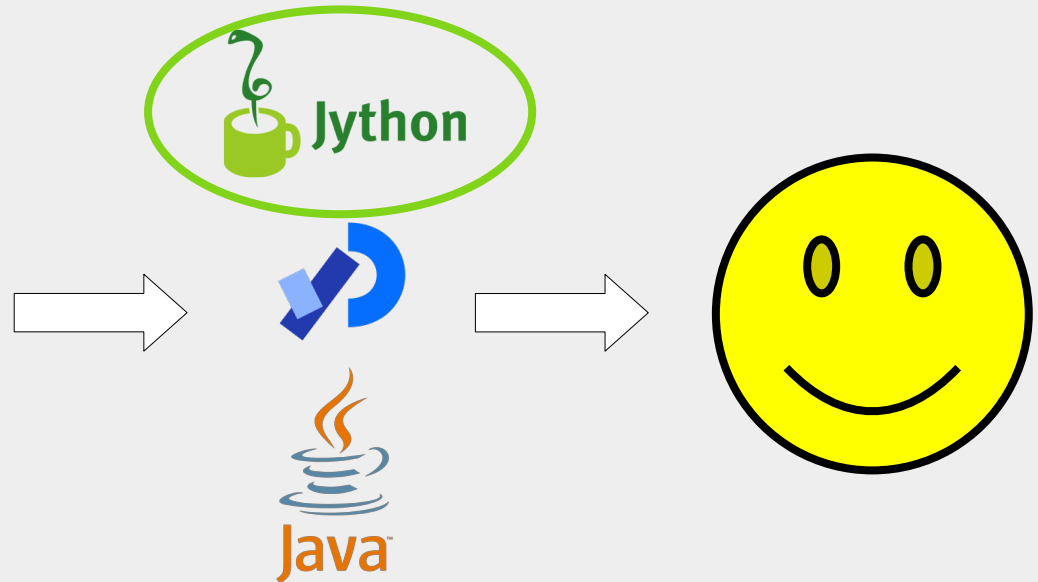
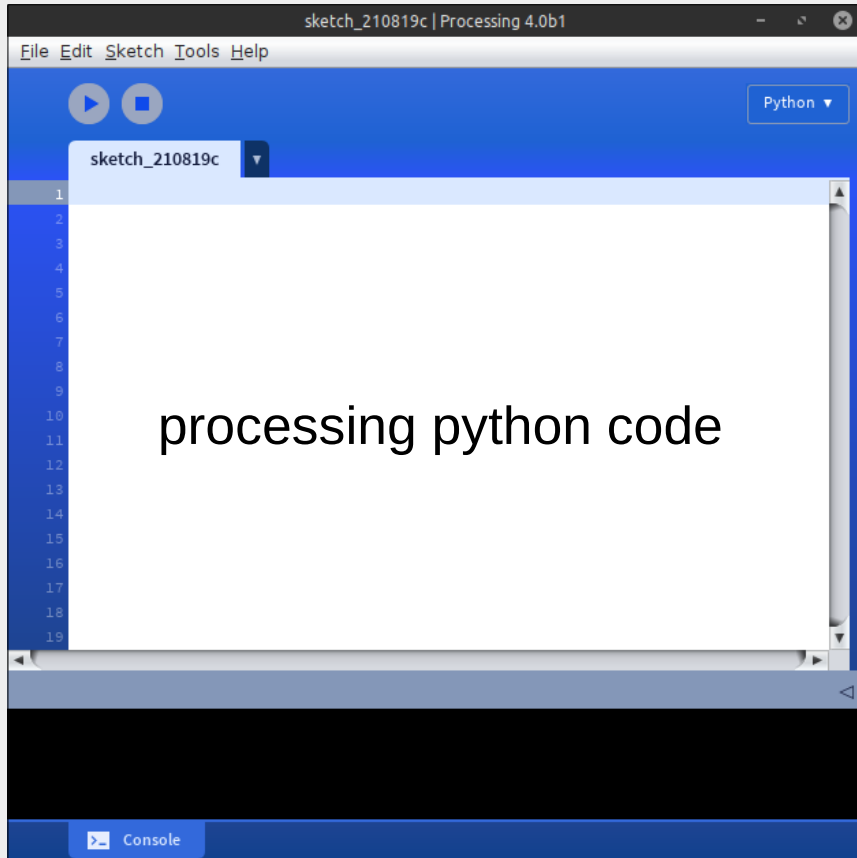
README.md

Processing + Python tools table

Name	Processing features	based on (& Python version)	Python standard library	libraries ecosystem	main features	main limitations
Processing Python Mode	Processing Java	Jython (Python 2)	complete	Java & Processing Java	available inside Processing IDE, very Processing compatible	no web sharing/deployment, no modern Python libs
p5py	a new implementation (incomplete)	Python 3	complete	Python only	truly Python compatible	Experimental, still incomplete, no web sharing/deployment, new names (for those used to Processing)
pyp5js (pyodide or transcript mode)	p5.js	Python 3 via Pyodide or Transcript	complete	Python, JavaScript & p5.js	web ready sketches & editor , very p5.js compatible & pyodide makes it very Python compatible	Experimental, still incomplete, p5.js features (as opposed to Processing Java/Python modes)
SkulptIDE and trinket.io	ProcessingJS	Skulpt (Python 2, going to 3 now)	partial	unknown, possibly JavaScript	very nice web IDE, browser based sketches	ProcessingJS is defunct; not extensible
					browser IDE	

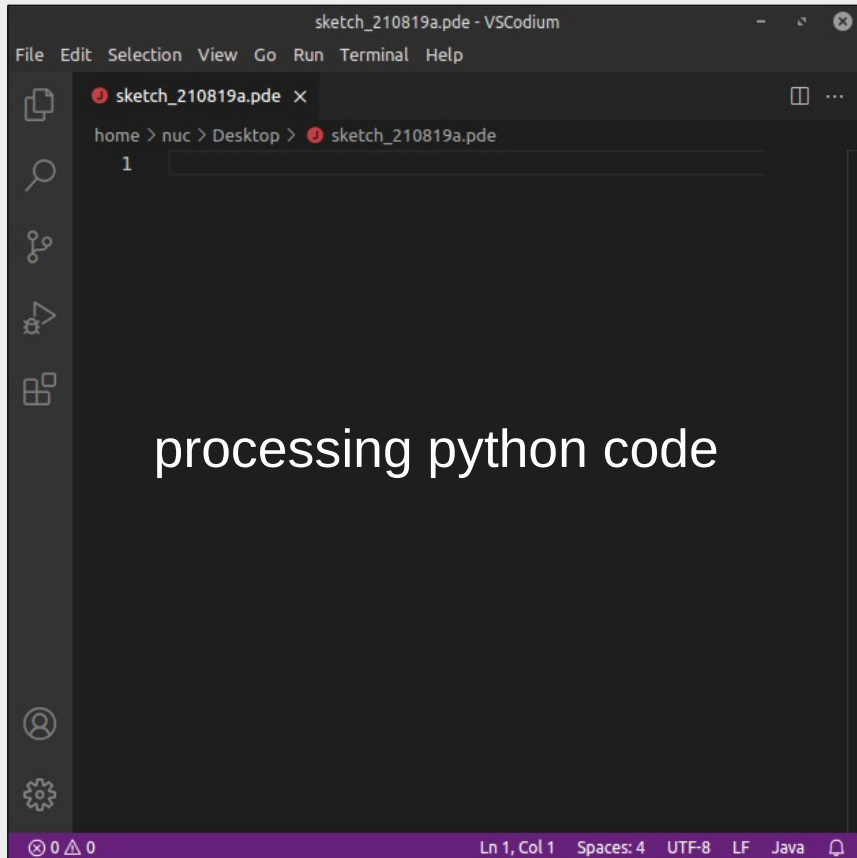
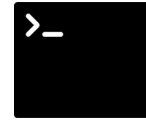
https://github.com/villares/Resources-for-teaching-programming



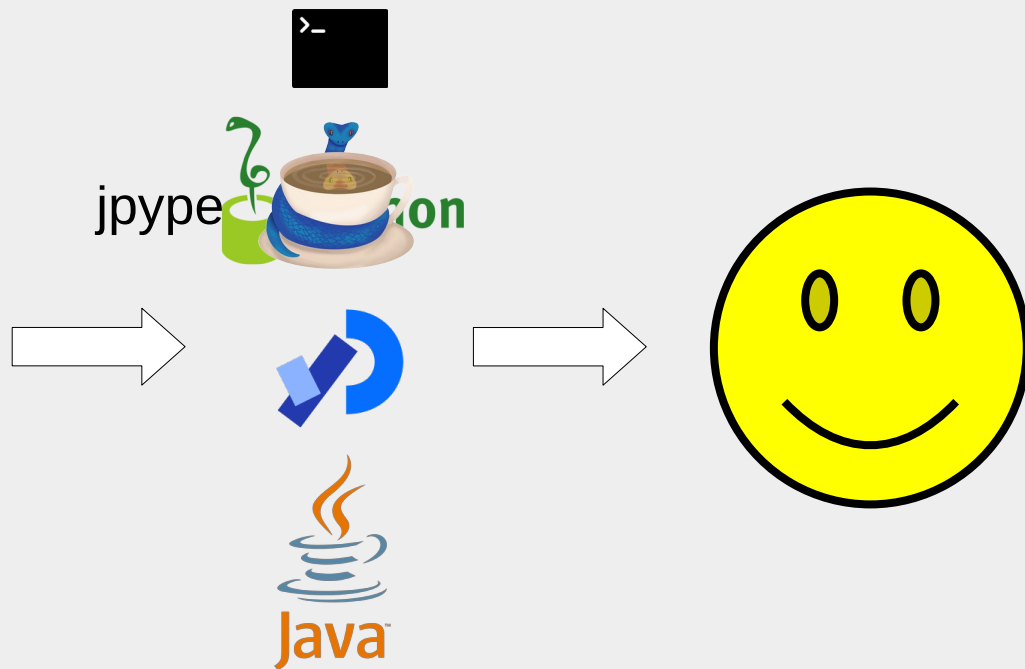
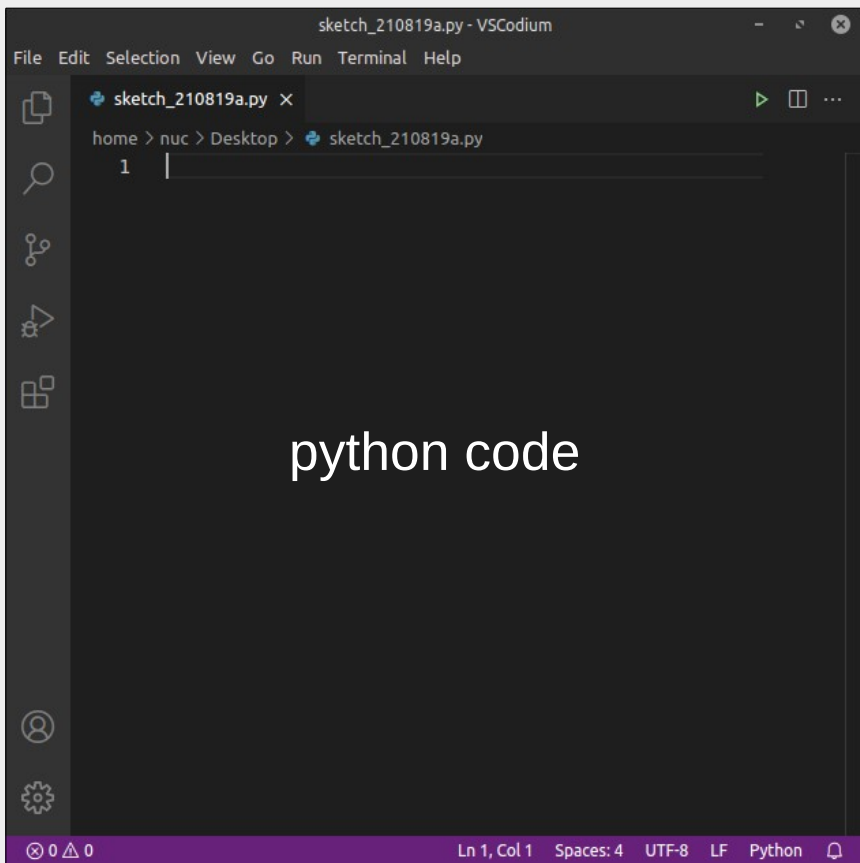
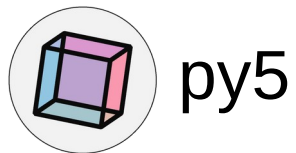


jython

- * is not cpython (reference implementation)
- * supports python 2.7
- * does not support 3rd-party python libraries with c extensions
- * hence "python mode" for processing (mrfeinberg.com)

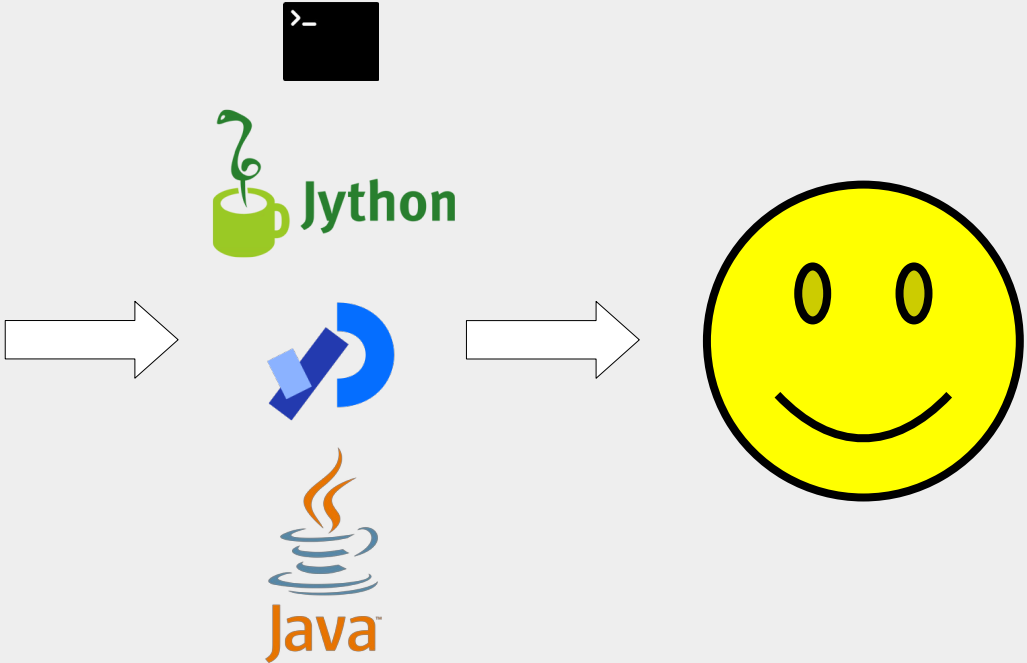
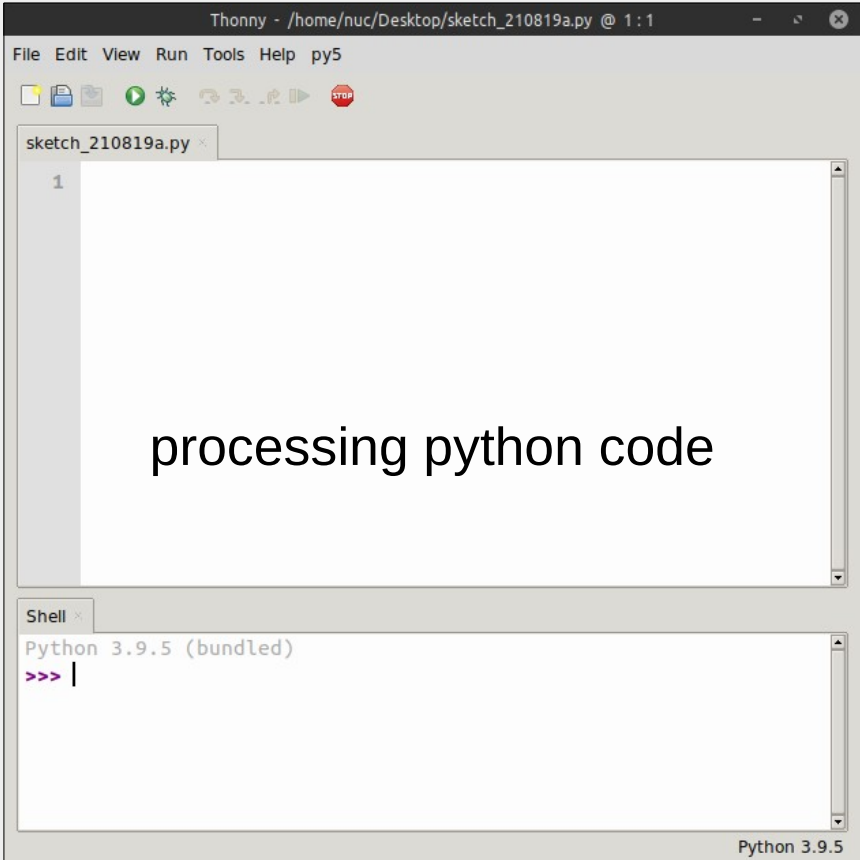


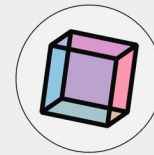
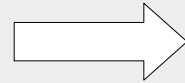
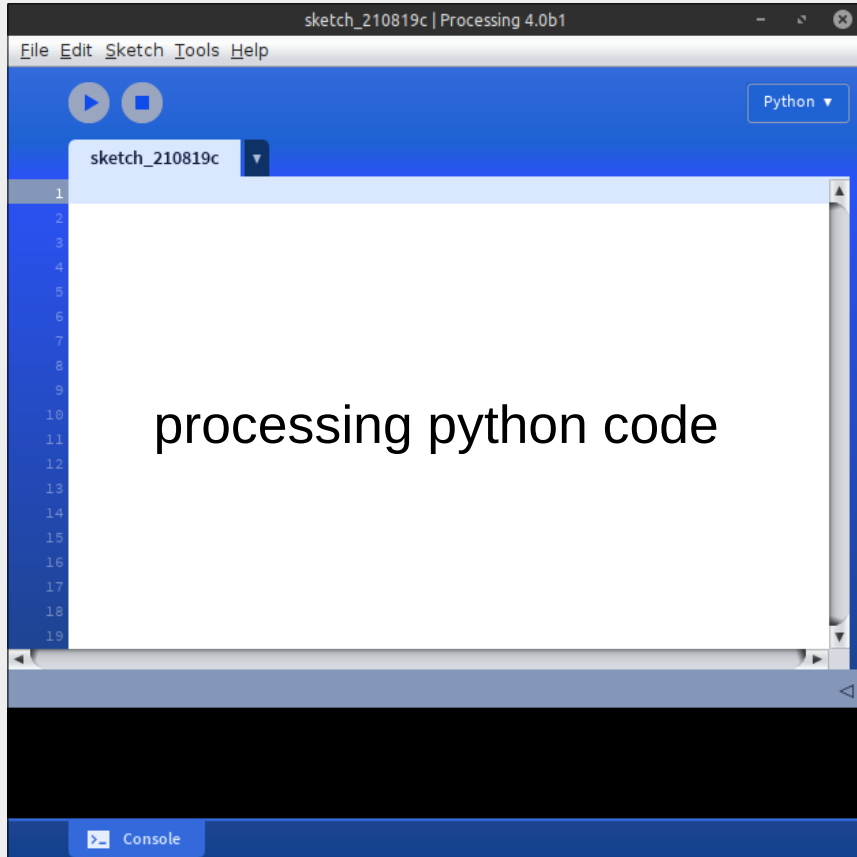
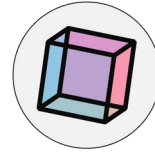
py5



jpype

- * provides full access to java libraries from within cpython
- * supports python 3
- * supports 3rd-party python libraries with c extensions





py5





JupyterLab — Mozilla Firefox

JupyterLab

https://hub.gke2.mybinder.org/user/hx2a-py5examples-m38okyfm/lab/tree/examples/Introduction%20to%20py5bot.ipynb

File Edit View Run Kernel Tabs Settings Help

Introduction to py5bot.ipynb

Download GitHub Binder Markdown

py5bot

This is py5bot. A simple and easy to use programming environment for teaching the very basics of Python programming and creative coding with py5.

Each cell in this notebook can contain a series of py5 drawing commands. The drawing commands will be executed to create a static image that will be embedded in the notebook.

The main design goal is to provide a simple programming environment that is suitable for beginners. Individual programming concepts can be explained in isolation from more complicated Python concepts like functions or modules.

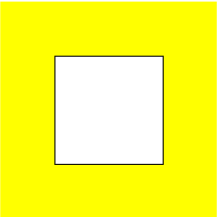
When hosted on Jupyter Hub and paired with Jupyter Lab's "Show Contextual Help" feature, py5bot can become an easy to use programming environment for educators to teach Python to beginners.

[launch binder](#)

Below is a simple example.

```
[1]: size(200, 200)
background(255, 255, 0)
rect(50, 50, 100, 100)
```

```
[1]:
```



<https://mybinder.org/v2/gh/hx2A/py5examples/HEAD?urlpath=lab>

```
# py5
```

```
processing.py (processing python mode) is the spiritual ancestor to and  
inspiration for py5
```

```
py5 is similar to processing python mode in that both use python syntax,  
but their implementations are very different; they do not share any  
code, but py5 benefits from code in the processing core libraries  
written to accommodate processing python mode
```

"my goal is to create a new version of processing that fits into the larger python ecosystem. python syntax is a byproduct of that, but it isn't the core goal. this also isn't an attempt to replace or compete with any other version of processing. py5 isn't and can't be a drop-in replacement for processing.py, even without the camel case / snake case differences. i made different design choices in pursuit of a different goal, resulting in different libraries that work differently"

--jim schmitz

numpy arrays to adjust pixels (np_pixels); matplotlib charts or pil images to py5image objects (convert_image); built in support for line_profiler for performance tuning; ...

About py5 | py5 — Mozilla Firefox

py5.ixora.io/about/

I started working on py5 in the Spring of 2020 when I was a Research Resident at ITP. It began as a diversion from my pandemic-related anxieties and grew into the library it is today. I intend to use it as an outlet for my artistic endeavors involving Python's machine learning and data science tools and want to make it available and useful for other artists to do the same.

The py5 library itself is created by the meta-programming project py5generator. The [source code for py5](#) and the [source code for py5generator](#) are both available on github.

Acknowledgments

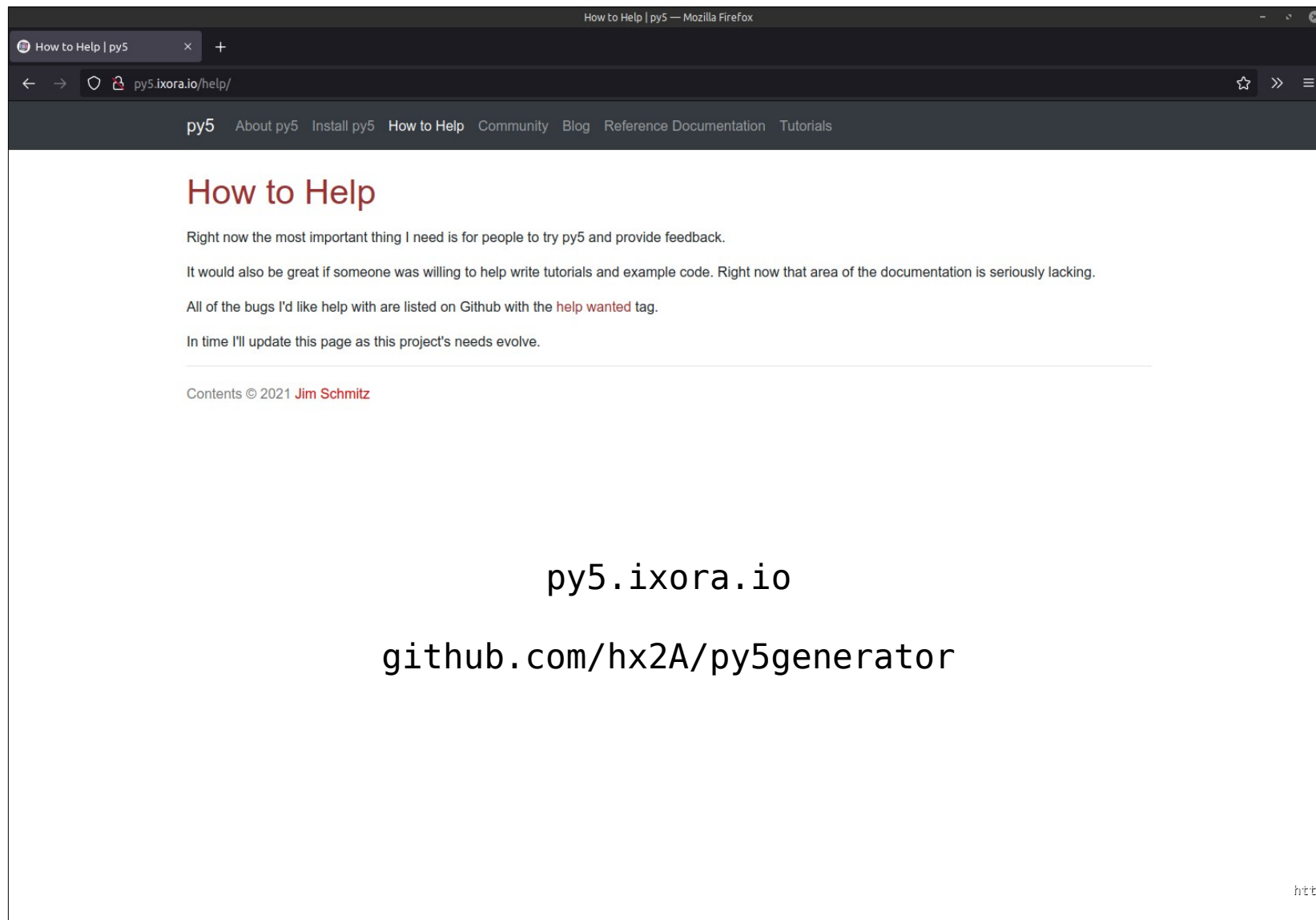
I'd like to thank and acknowledge all the people who helped make py5 possible.

- [Ben Fry](#), [Casey Reas](#), and [the rest of the Processing team](#). Processing has been around for almost 20 years and is used by thousands for creative coding projects. Much of py5's functionality is provided by the Processing core libraries. Py5 stands on the shoulders of giants.
- [Jonathan Feinberg](#) and the rest of the [Processing.py](#) contributors. Processing.py is a Jython version of Processing, combining the same Processing core libraries that py5 utilizes with Jython, a Java implementation of Python. Processing.py is the spiritual ancestor to and inspiration for py5. Py5 is similar to Processing.py in that both use Python syntax but their implementations are very different. Processing.py and py5 do not share any code but py5 benefits from code in the Processing core libraries written to accommodate Processing.py.

I'd also like to thank Jonathan Feinberg for building the awesome Processing library [PeasyCam](#). PeasyCam is one of the Processing libraries I know to work well in py5.

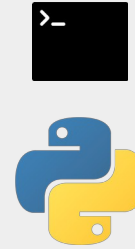
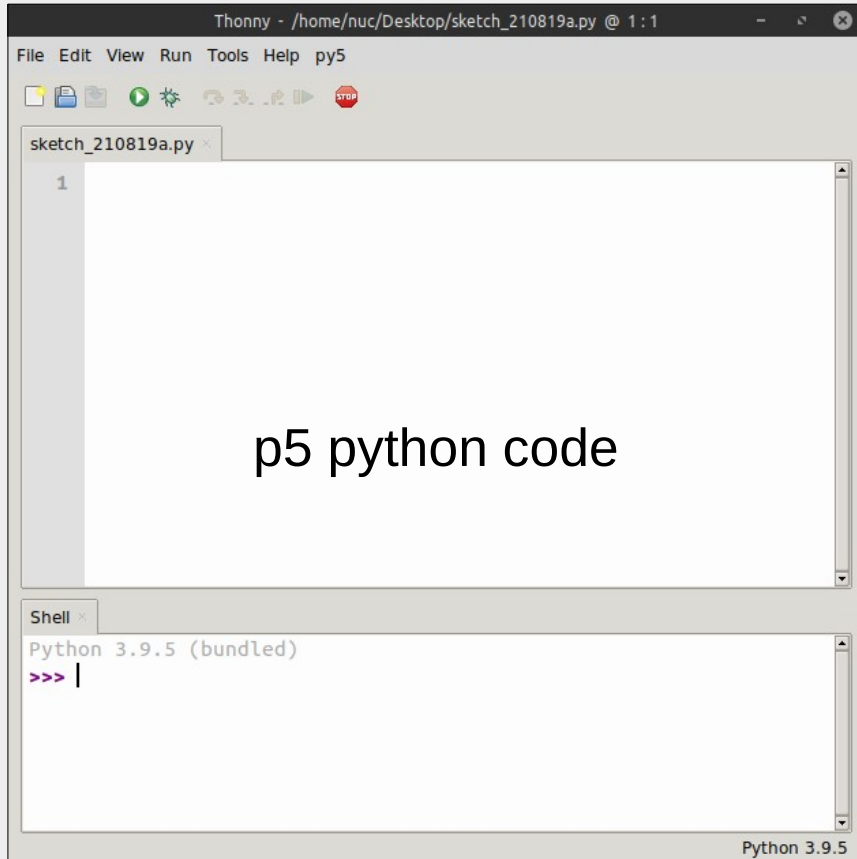
- The developers of the [JPype](#) and [PyJNIus](#) Python libraries. Both of these libraries allow Python code to interact with Java code in the Java Virtual Machine using [JNI](#). Py5 originally used PyJNIus but later switched to JPype. PyJNIus is maintained by the [Kivy](#) project (which I am a member of). JPype's lead developer [Thrameos](#) introduced us to their library and motivated me to switch.
- [Lauren McCarthy](#), who created [p5.js](#), a JavaScript version of Processing. Lauren helped me understand the importance of developing the [py5 Community](#). It is from p5 that py5 gets its name.
- The ITP faculty, including [Tom Igoe](#), [Dan Shiffman](#), and [Allison Parrish](#). All provided early feedback that provided guidance and helped keep me motivated. Allison also helped me understand the importance and value of the integrating py5 with Jupyter notebooks.
- The ITP residents of 2019 to 2020, for putting up with me and being available to bounce ideas off of as I was in the early stages of developing this idea.

Contents © 2021 [Jim Schmitz](#)



p5

p5 (py)



p5

- * python interpreter (reference implementation, cpython)
- * p5 is a native python **port** of the processing api
- * p5.readthedocs.io

session outline

`github.com/tabreturn/cc-fest-py5`

session outline

- * mybinder

- * thonny

 - * setting up thonny and thonny-py5mode plugin

 - * py5 examples

- * q & a

`github.com/tabreturn/cc-fest-py5`

end of slides