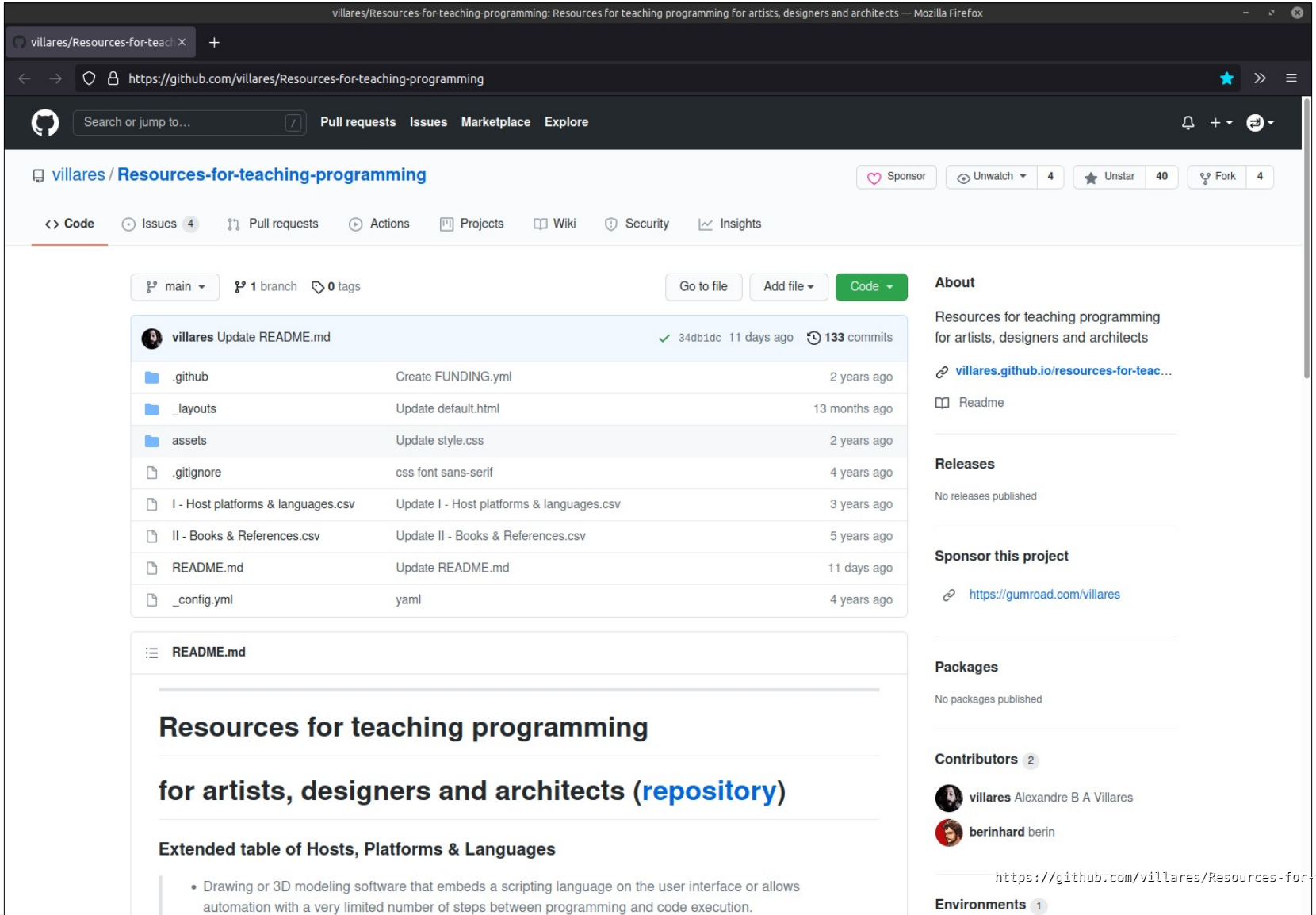
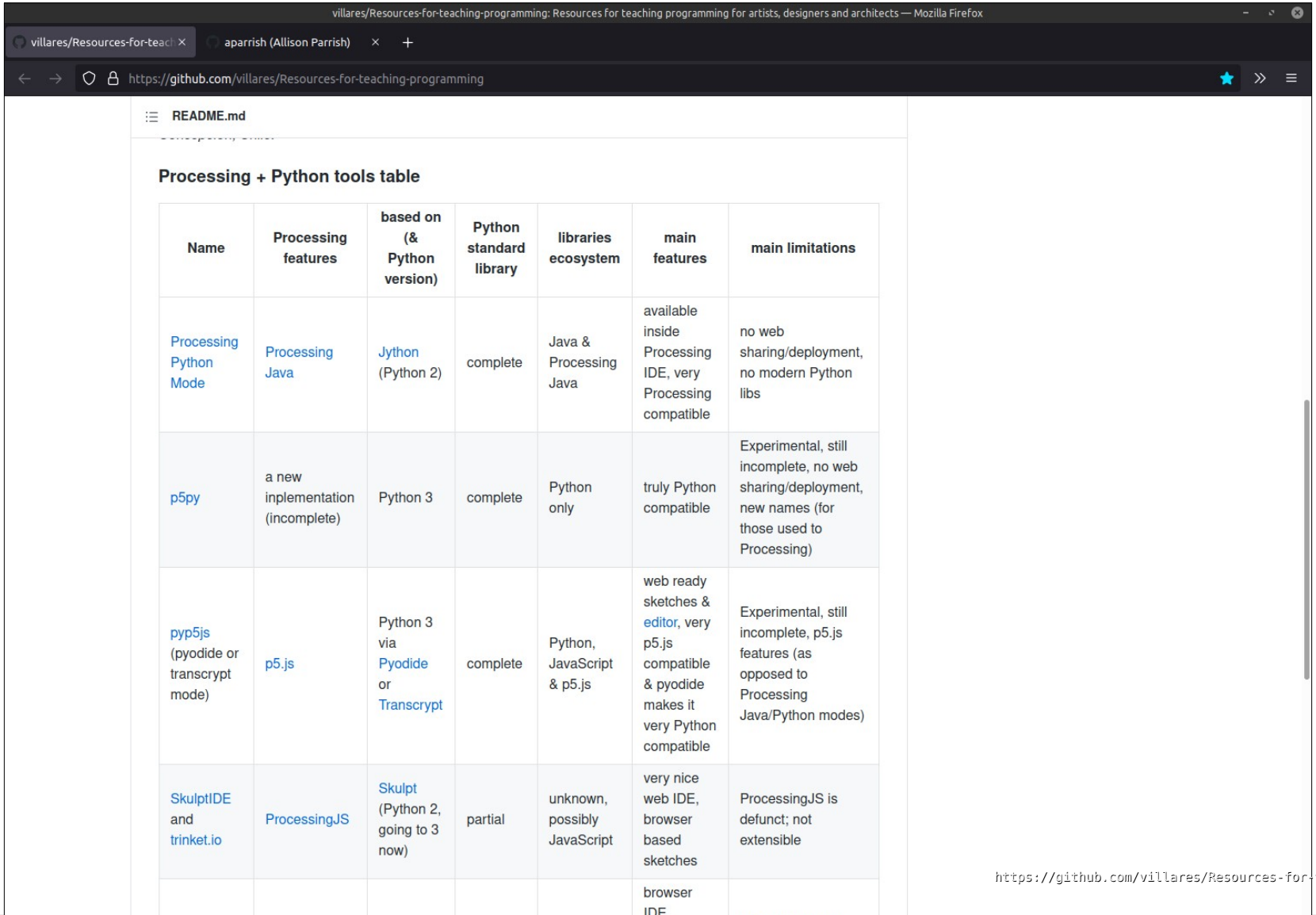
thonny + py5

a python 3 environment for processing

processing + py5 ecosystem



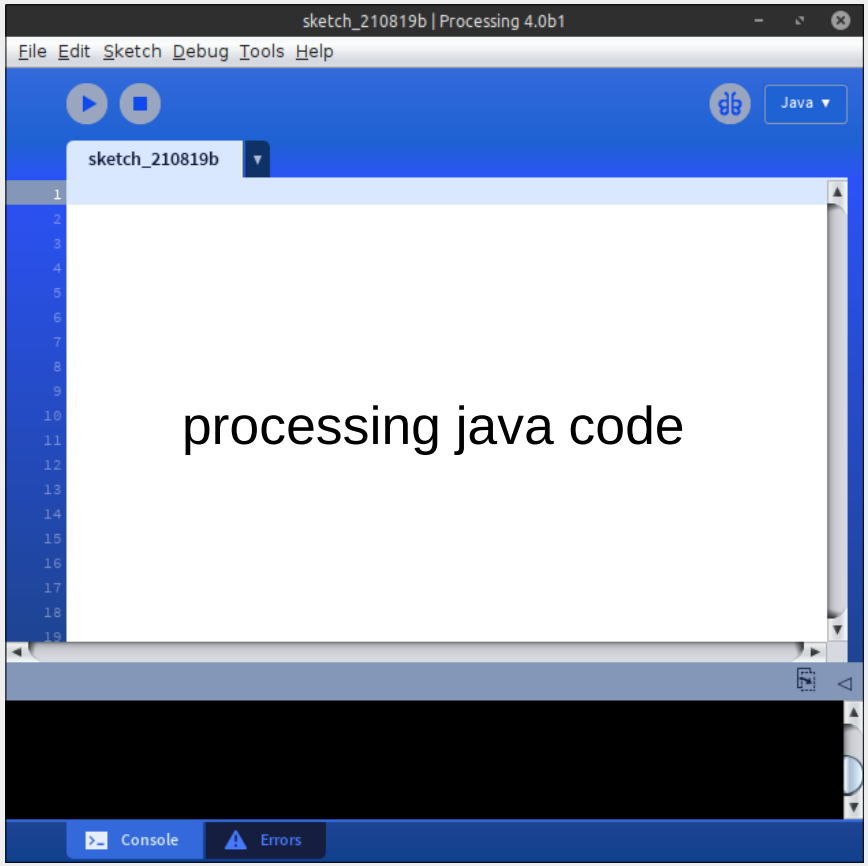
https://github.com/villares/Resources-for-teaching-programming

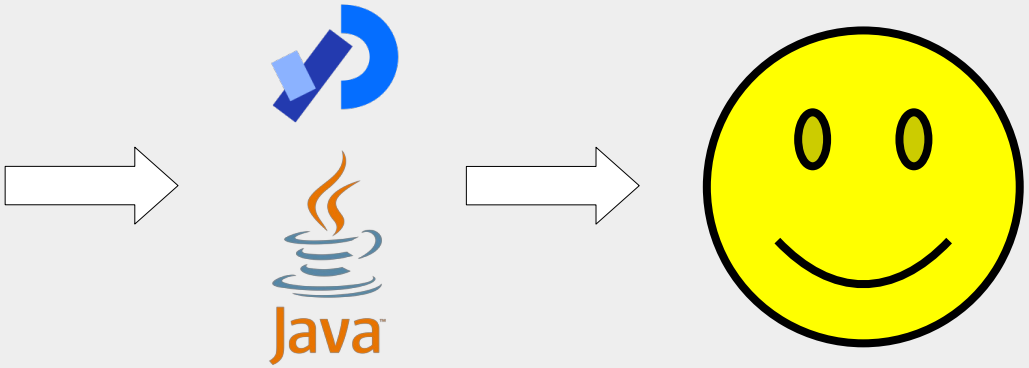


Processing + Python tools table

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 inplementation (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 transcrypt mode)	p5.js	Python 3 via Pyodide or Transcrypt	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	

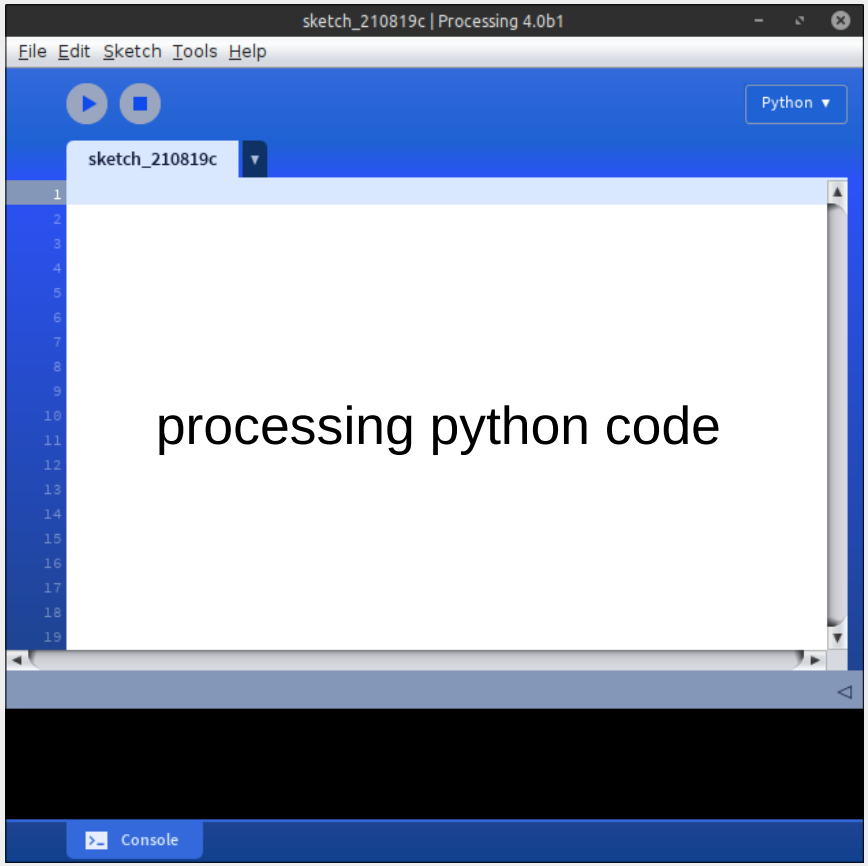
Processing

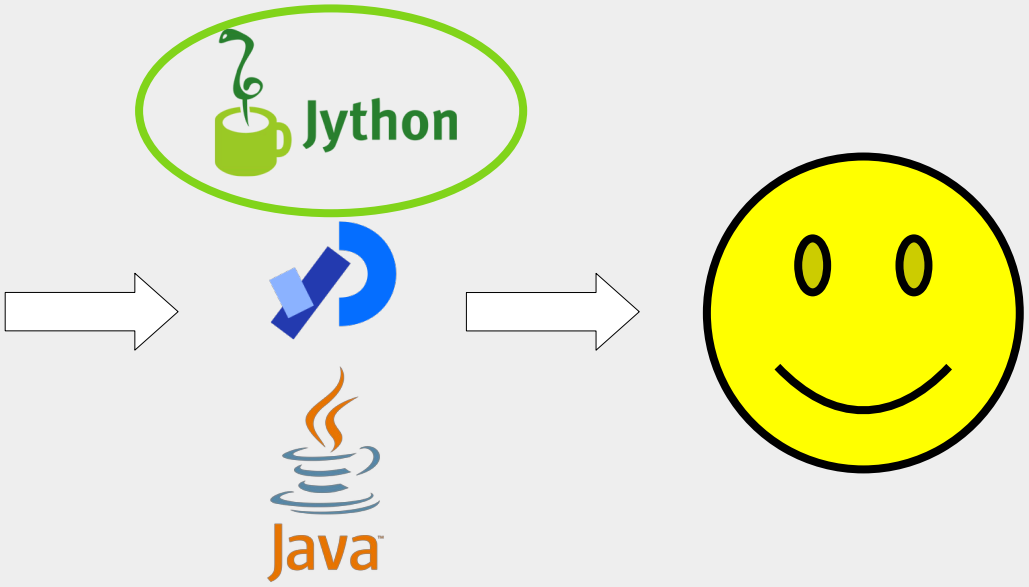




https://processing.org/

Processing
Python

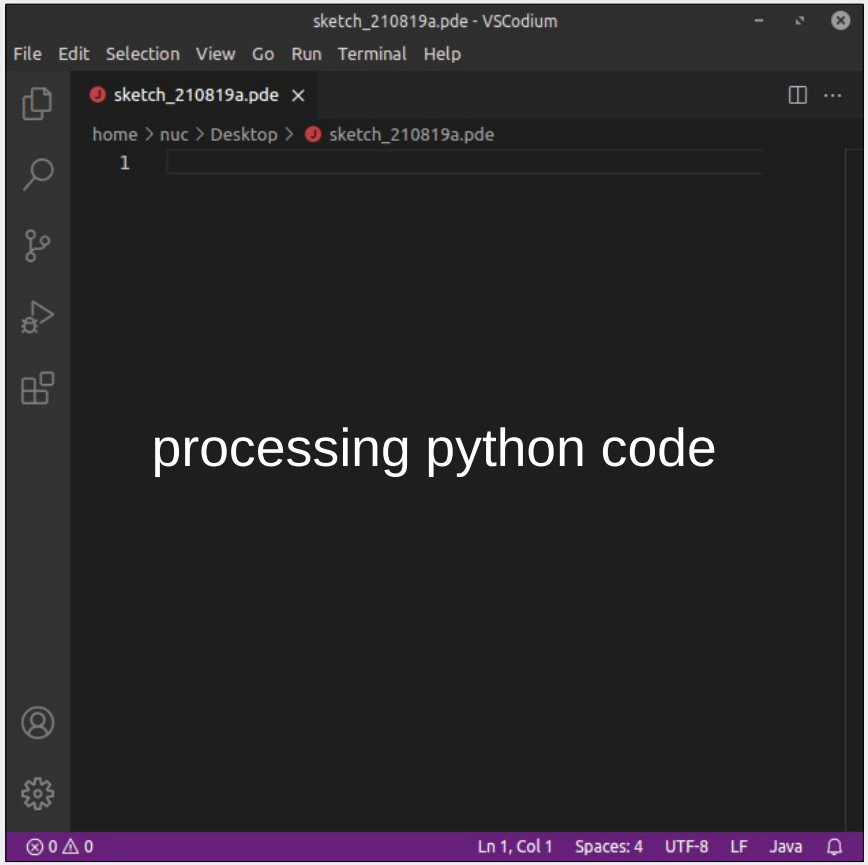


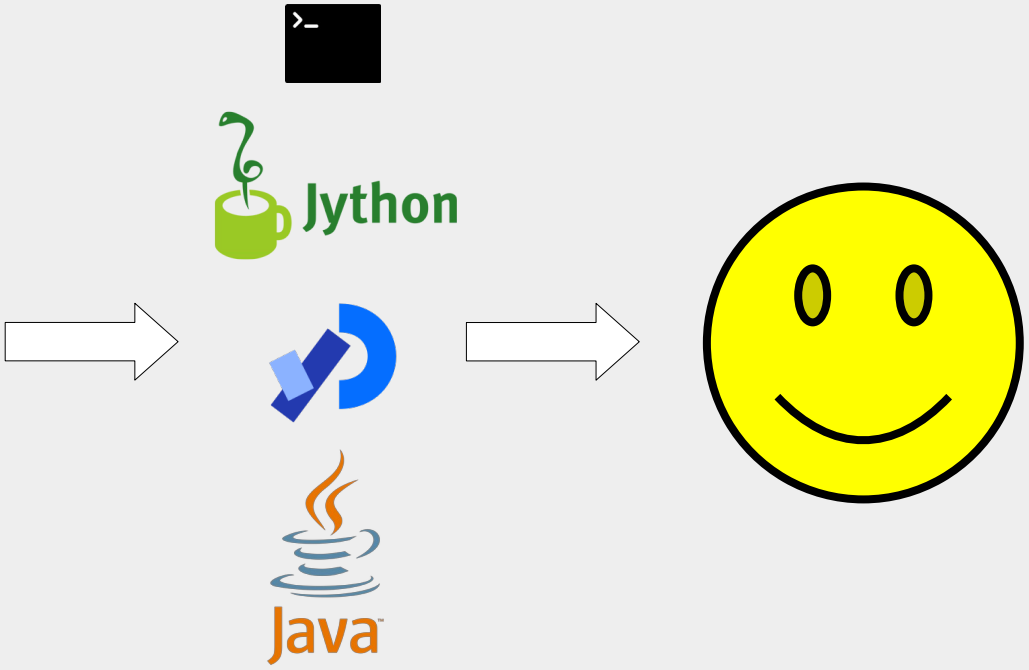


jython

* 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)

Processing Python + >_

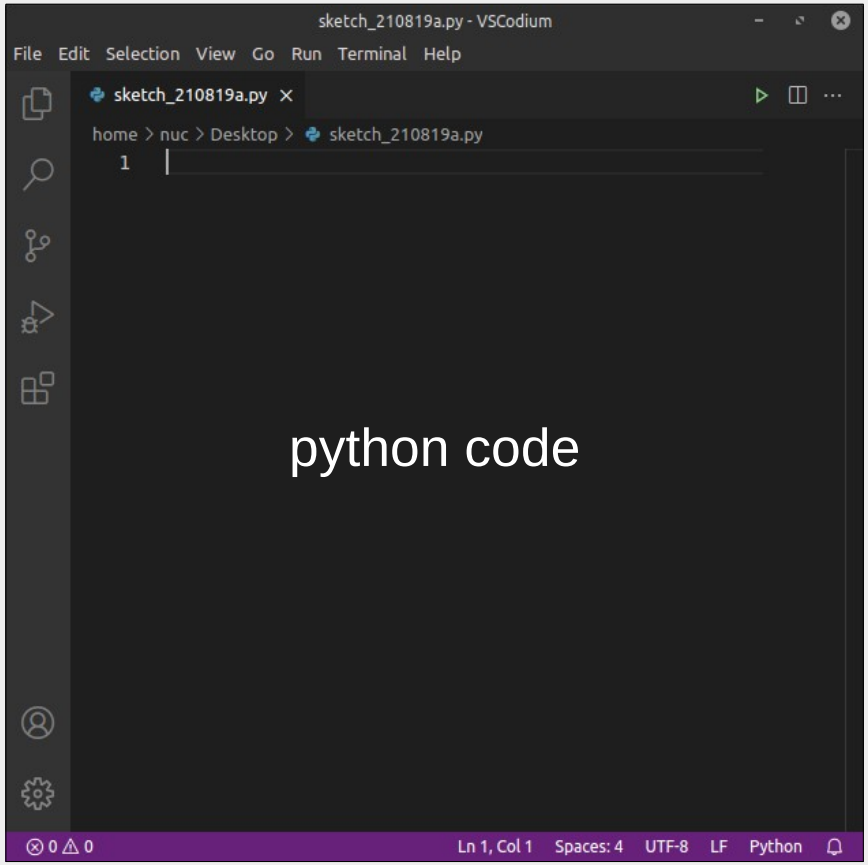


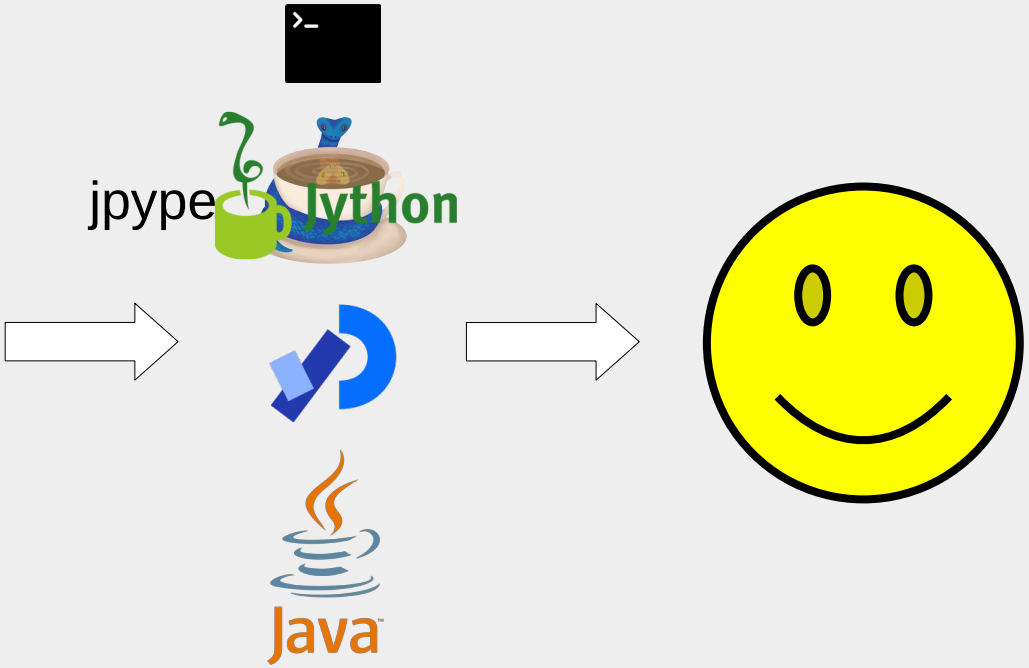


https://py.processing.org/tutorials/command-line/

py5

py5


sketch_210819a.py - VSCodium
File Edit Selection View Go Run Terminal Help
sketch_210819a.py
home > nuc > Desktop > sketch_210819a.py
1
python code
Ln 1, Col 1 Spaces: 4 UTF-8 LF Python

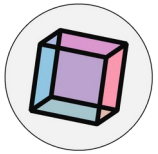

jpype
Jython
Java

jpype

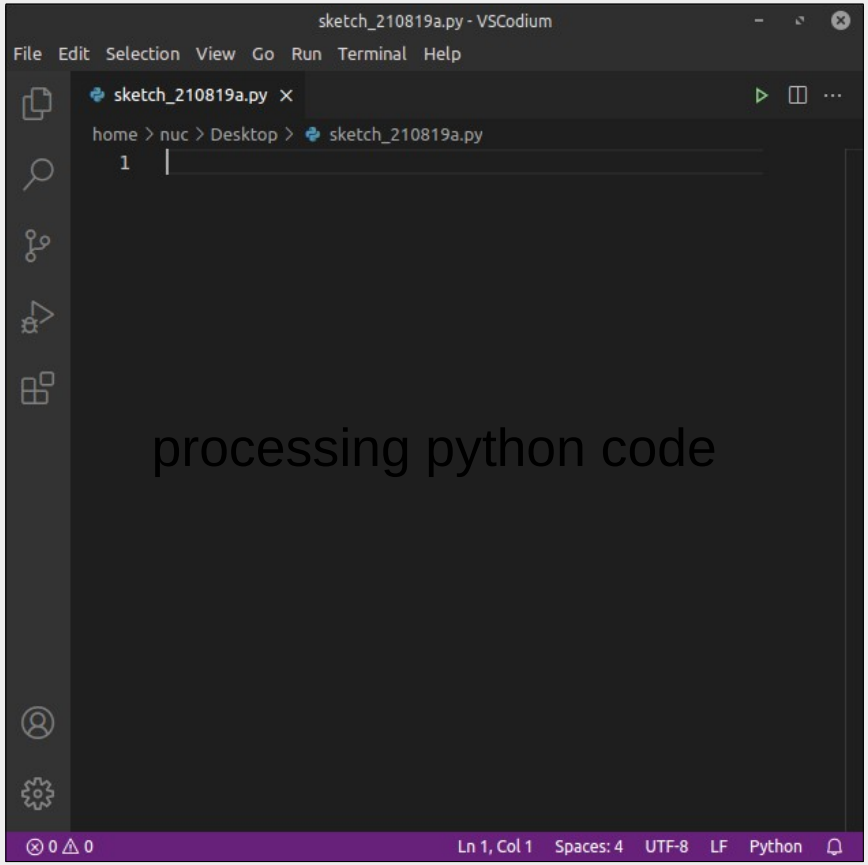
* provides full access to java libraries from within cpython

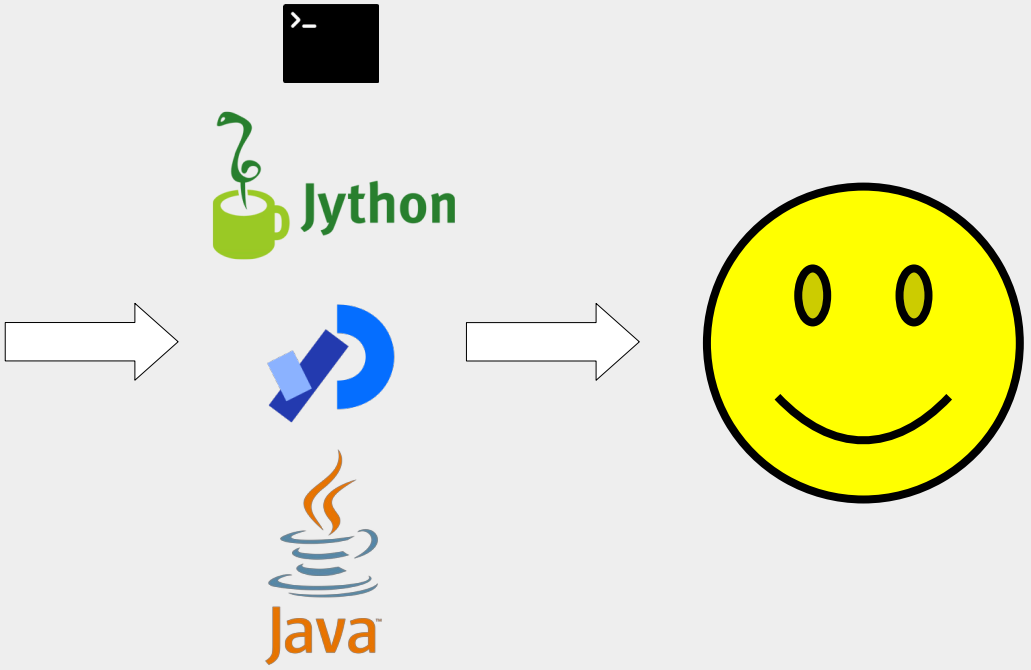
* supports python 3

* supports 3rd-party python libraries with c extensions

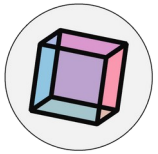


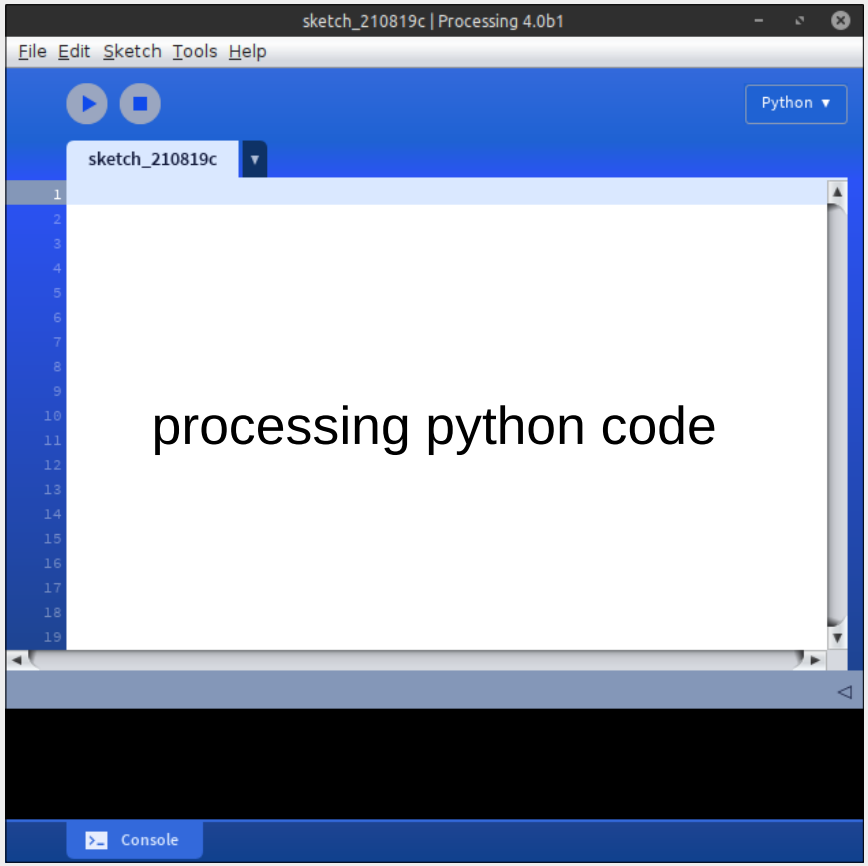
py5 + Th

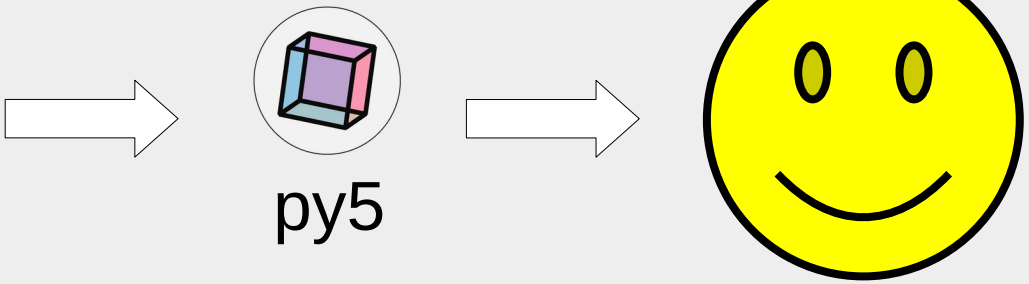


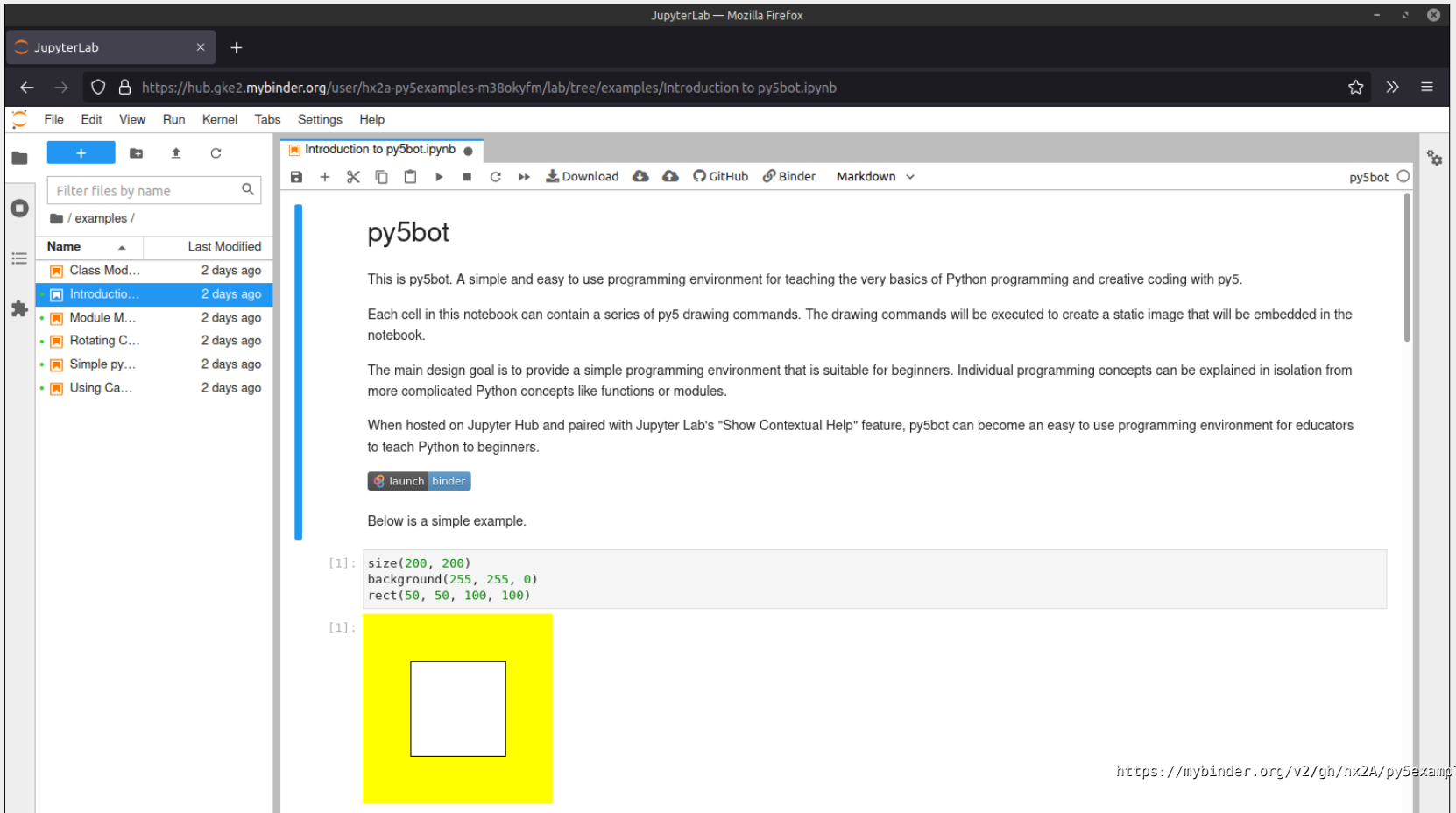


https://thonny.org/

Processing Python + 







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

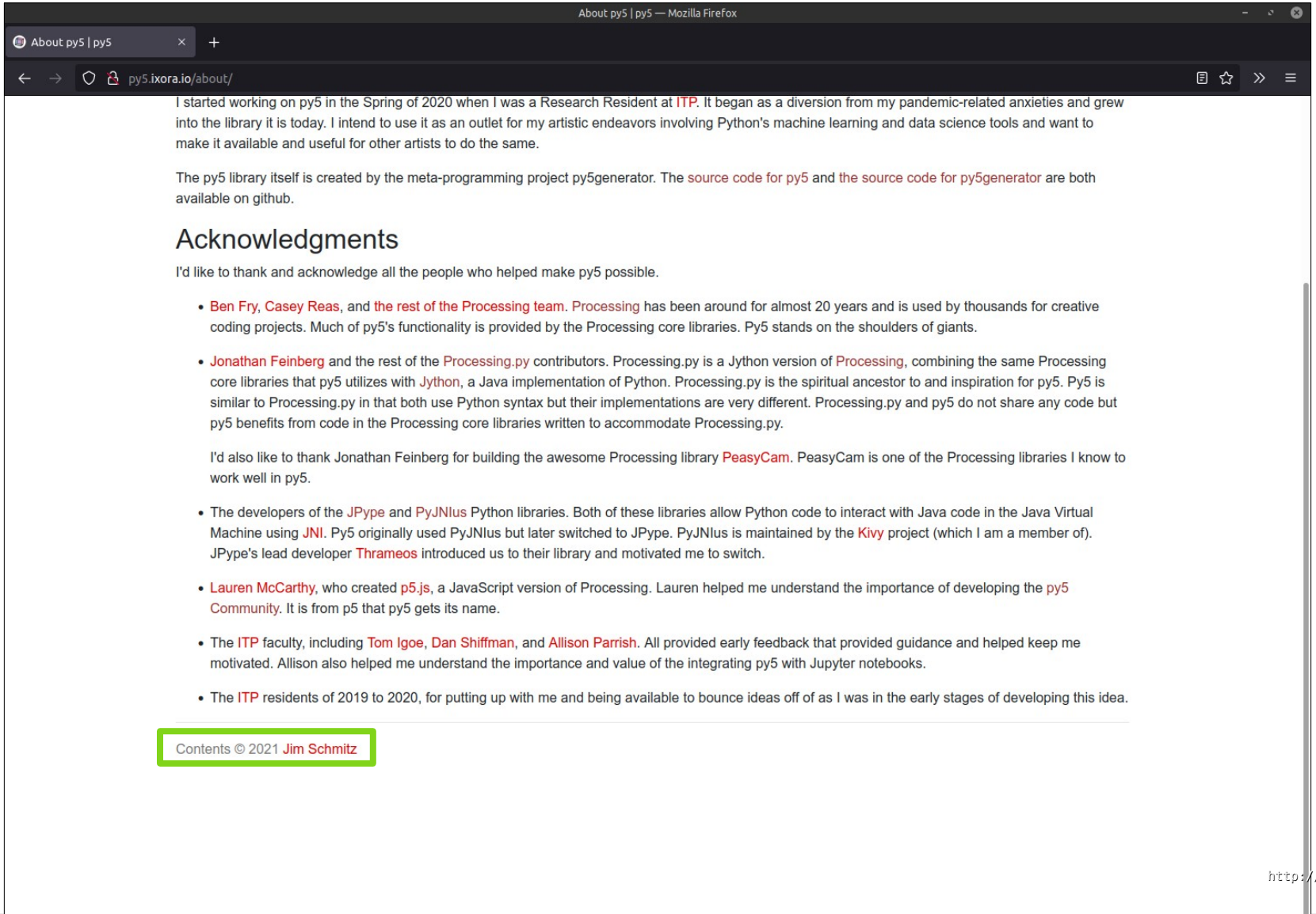
py5

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; ...



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.

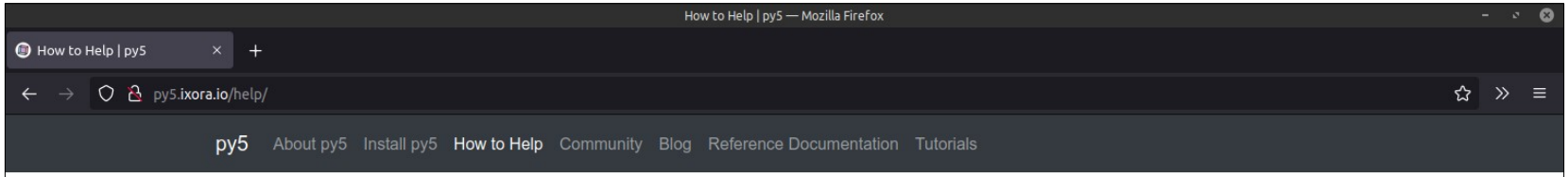
The py5 library itself is created by the meta-programming project py5generator. The source code for py5 and the source code for py5generator are both available on github.

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.



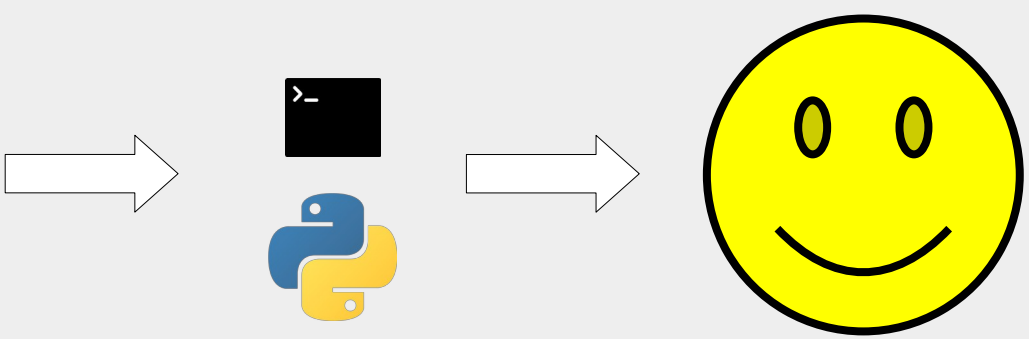
py5.ixora.io

github.com/hx2A/py5generator

p5

p5(py)

p5py



https://github.com/p5py/p5

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