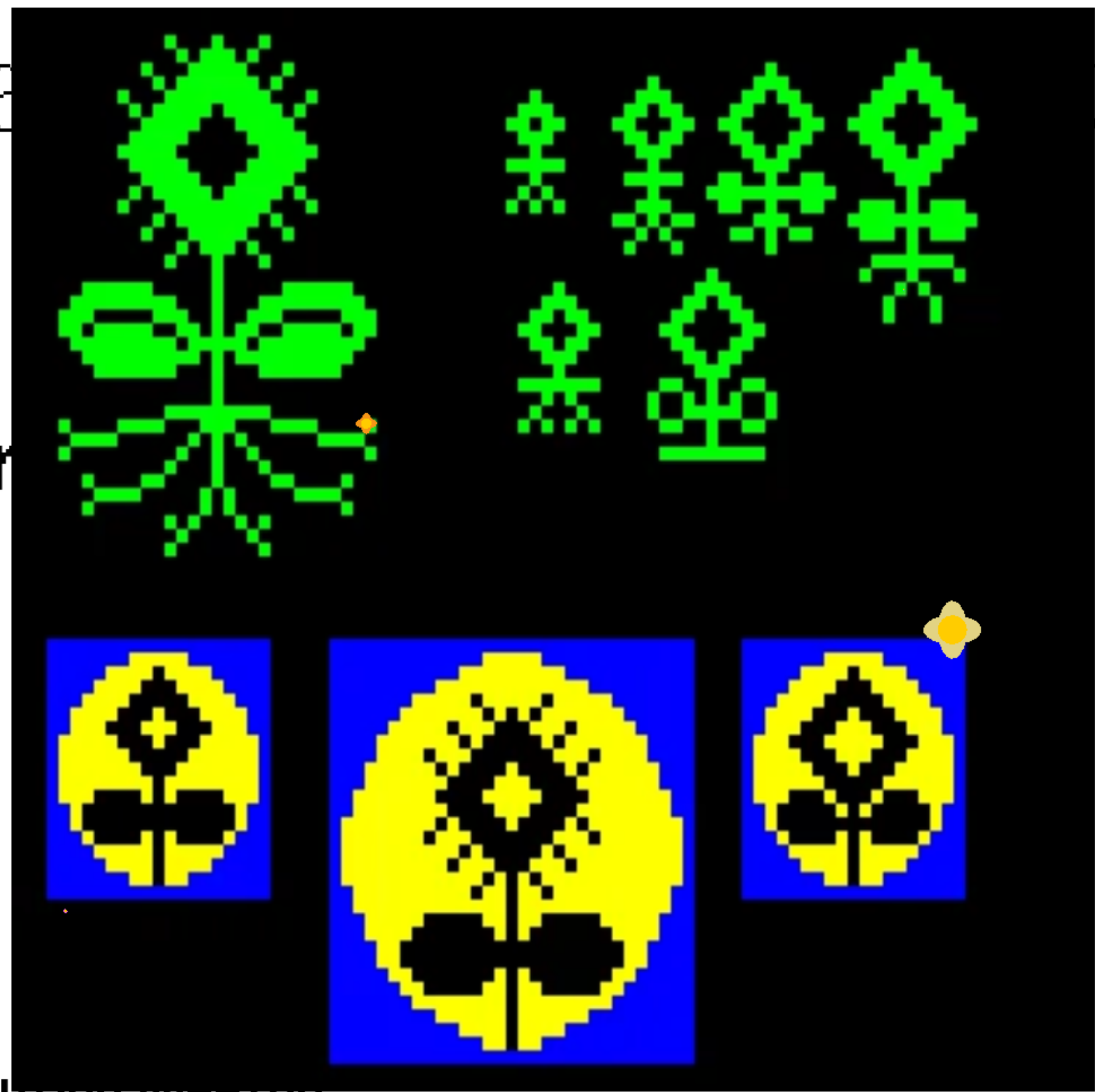


# L5: a permacomputing-inspired creative coding library

**Lee Tusman**

25: a piece  
creative

Lee Tusman



red

What is L5?

A new (alpha) creative coding library in the family of Processing and p5, inspired by Permacomputing.

<https://L5lua.org>

What is

A new (alpha)  
of Processing



in the family  
of computing.

<https://L5lua.org>

# What is permacomputing?

**Permacomputing is both a concept and a community of practice oriented around issues of resilience and regenerativity in computer and network technology inspired by permaculture.**

What is

# permacomputing

Welcome to the [Permacomputing](#) wiki!



Permacomputing is both a concept and a community of practice oriented around issues of resilience and regenerativity in computer and network technology inspired by permaculture.

☺ \*☆ -☆:\*

There are huge environmental and societal issues in today's computing, and permacomputing specifically wants to challenge them in the same way as permaculture has challenged industrial agriculture. With that said, permacomputing is an anti-capitalist political project. It is driven by several strands of anarchism, decoloniality, intersectional feminism, post-marxism, degrowth, ecologism.

Permacomputing is also a utopian ideal that needs a lot of rethinking, rebuilding and technical design work to put in practice. This is why a lot of material on this wiki is highly technical.

Most importantly, there is no permacomputing kit to buy. See permacomputing as invitation to collectively and radically rethink computational culture. It is not a tech solution searching for a problem. You are free to start your own initiative and use the term permacomputing, however please make sure you understand the purpose and ethos of this project :)

Permacomp  
community  
resilience a  
network te

permacomputing wiki

1 a  
d issues of  
ter and  
culture.



LS is in the Processing family of  
languages

**Processing is a flexible software sketchbook and a  
language for learning how to code.**

LS is in  
language

Processing  
language fo

processing.org

The screenshot shows the Processing.org homepage. At the top, there are logos for Processing Foundation, Processing, p5.js, Processing Android, and Processing Python. Below these are navigation links: Processing, Download, Documentation, Learn, Forum, About, and Donate. A search bar is also present. The main content area features a large blue graphic of a pencil and a checkmark, with the text "Welcome to Processing!" and a paragraph describing Processing as a flexible software sketchbook and a language for learning how to code. Below this text are three buttons: Download, Reference, and Donate. At the bottom right of the main content area is an "Open Editor" button. Below the main content area is a section titled "Examples" with four small thumbnail images showing various Processing projects: a colorful gradient, a colorful grid, a dark scene with a light source, and a 3D rendered object.

city of

chbook and a

For years I've made my art with

p5.js

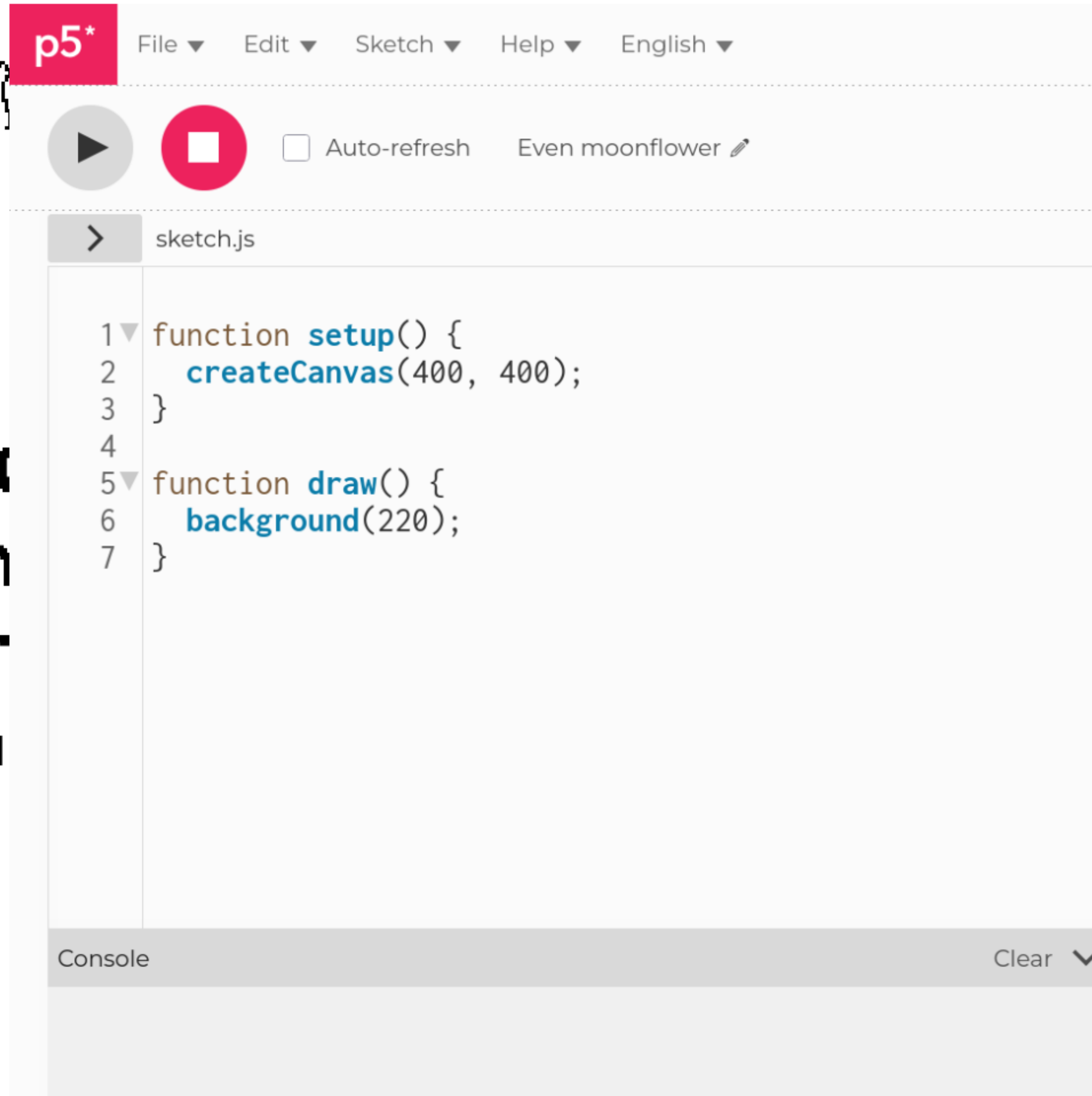
- \* p5.js is processing for the browser
- \* It's implemented in JavaScript
- \* p5.js prioritizes accessibility and community contribution

For year

p5.js

- \* p5.js is pro
- \* It's implem
- \* p5.js prior contributio

p5js.org



with

community

p5.js places a focus on access

**a commitment to only add new features that increase access (inclusion and accessibility)**

**<https://p5js.org/contribute/access>**

p5.js role

access

a commitment  
increase ac

es that  
bility)



<https://p5js.org/contribute/access>

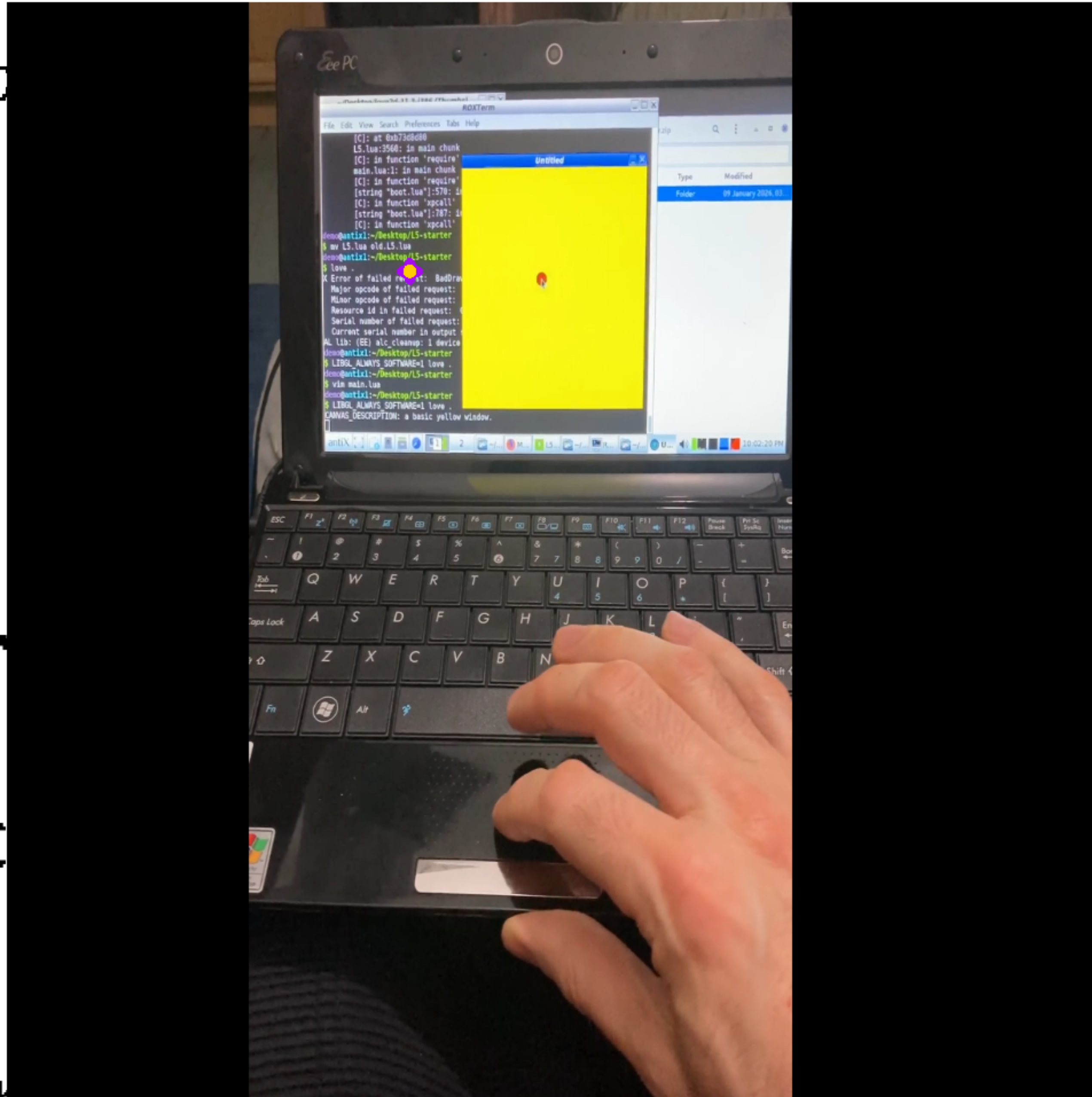
# What does L5 do differently?

- \* Considers access to include permacomputing principles
- \* Small, constrained, \*simple\*, runs on LOTS of hardware, fast, works offline
- \* built for the long term\*\*

\*\*note: L5 is in alpha, so we can only test longevity of older computers

What do

- \* Considers principles
- \* Small, compact hardware,
- \* built for t



you?

Computing  
LOTS of

\*\*note: L5 is in alpha, so we can only test longevity of other computers

# Responding to Permaconnecting

## Principles

- \* Exposes more seams of the underlying system
- \* Currently requires more complex install
- \* The functions work like p5.js, except when they work like Processing!
- \* The website is designed to work on slow connections
- \* And can be downloaded and run entirely offline

Responsible

Principles

- \* Exposes n
- \* Currently
- \* The funct.
- work like P
- \* The webs.
- connections
- \* And can b

## permacomputing/ principles

### Introduction

Contemporary permaculture is founded on three core ethics: Earth Care, People Care, and Fair Share. These ethics serve as a guiding compass for its design principles, co-creating a holistic framework for regenerative living. Similarly, permacomputing is built upon 10 principles that encourage and raise awareness about more sustainable digital practices.

Whether you are a tech specialist, someone who uses a computer for daily tasks, or deals with technology only occasion, there are steps that you or the group you are involved in can take to reduce the environmental and socio-economic impact of your digital activities. The following section explores the permacomputing principles, showing how and where to start.

Each principle is illustrated through:

- **Issue/background**
- **What can YOU do?** Strategies and interventions for all, from techie to casual user, or something in between
- **Principle in action & examples** Further reading, tools for deeper engagement, and examples of implementation

The community of permacomputing promotes a comprehensive approach to the design of human technology, taking into account social and ecological issues, encouraging resilience and supporting a fair coexistence. These design principles are meant to not to be just technical guidelines but rather represent strategies for positive changes.

**That said, permacomputing is not prescriptive;** it instead favors situatedness and an awareness of contextual diversity. We, the Permacomputing working group, hope these design principles can serve both as a guide for practice in specific situations and as a tool for identifying systemic issues in the relationship between computer technology and ecology.

Furthermore, the Permacomputing community is more than this site! It is inspired by—and builds upon—a diverse range of initiatives, research, projects, and bodies of knowledge. And just like we learn and expand concepts of permacomputing further, these principles will also continually be developed and refined. The text below is meant to serve as a starting point. :)

string

g system  
tall  
when they  
slow  
rely offline

# Simplicity Complexity and Scale

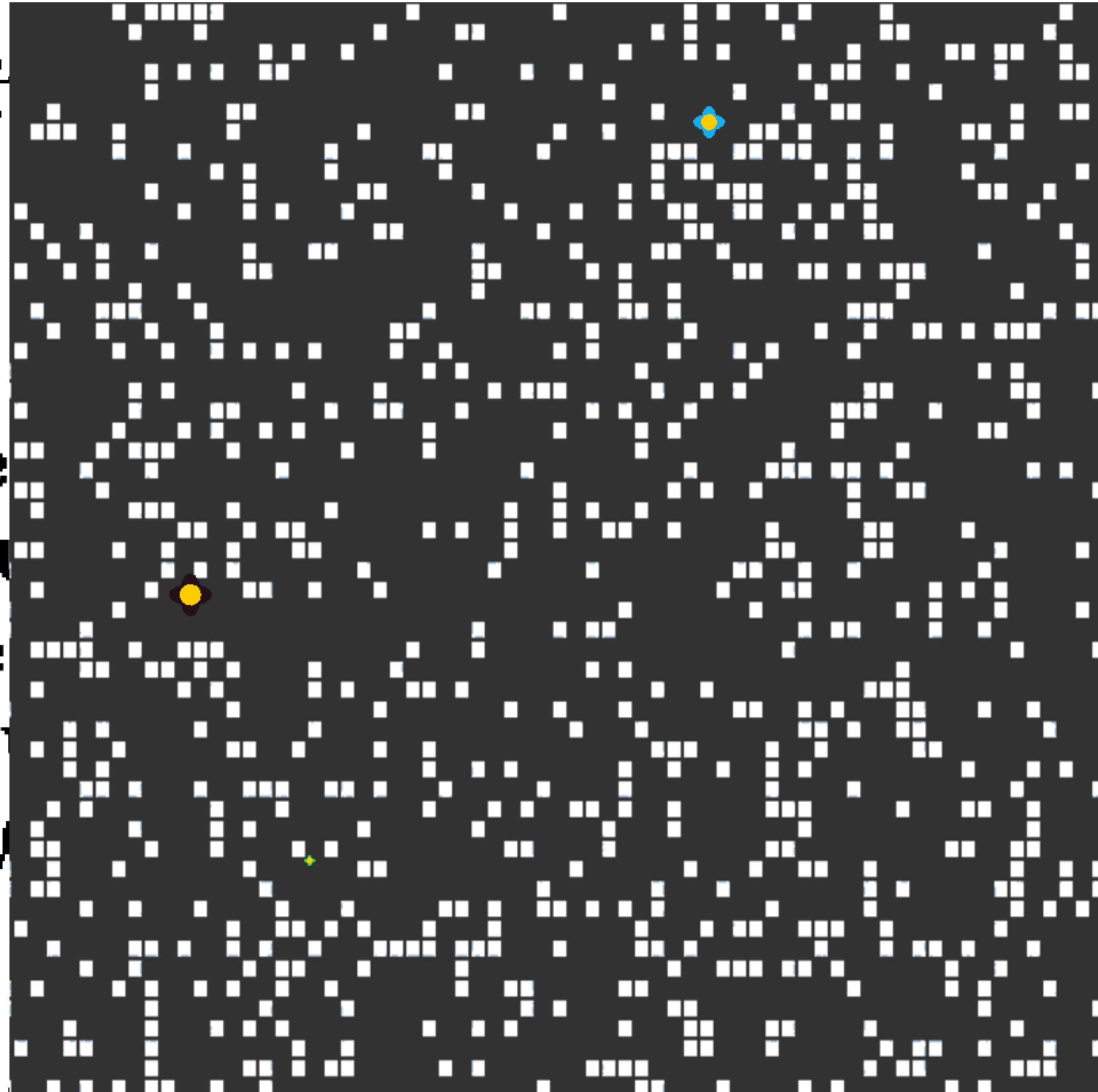
- \* graspable by an individual
- \* avoid pseudosimplicity
- \* reduce dependencies
- \* wisdom over accumulation
- \* creativity thrives within constraints.

<https://permacomputing.net/consider-carefully-the-interaction-between-simplicity-complexity-and-scale/>

Simplicity

Scale

- \* graspable
- \* avoid pseudo
- \* reduce de
- \* wisdom of
- \* creativity



<https://permacomputing.net/consider-carefully-the-interaction-between-simplicity-complexity-and-scale>

# Build on Solid Ground

- \* Lua is a Brazilian-born and led programming language
- \* It is lightweight, extremely cross-platform, fast, and has a fun, simple syntax
- \* very small

Build on

- \* Lua is a B language
- \* It is lightv and has a fi
- \* very smal



programming  
platform, fast,

L5 is built on the Love2d  
framework



- \* Love2d is a framework for making 2d games. It is open source, and has a wonderful community.
- \* small, well-documented
- \* The celebrated game Balatro was made with it

LS is ba

framework

- \* Love2d is open source
- \* small, well
- \* The celeb



d games. It is  
community.

made with it

love2d.org

# Not Doing

- \* **Constraint drives design**
- \* **what is necessary?**
- \* **consider dependencies**
- \* **consider hardware**
- \* **consider maintenance**

# Not Doing loadVideo()

Loads and returns a video file for simple audio/video playback.

Note: Currently L5 can only play ogv (Ogg Theora) video files. Use an external program such as [Handbrake](#) or [ffmpeg](#) (command line) to convert mp4, avi, mkv, and mov codecs first.

- ❖ **Constrain** *Differences from Processing and p5.js:* Processing requires a separate Video library but can work with more codecs. p5.js requires creating a DOM Video element separate from the canvas and playing once loaded asynchronously. L5's load and play functions are simple and synchronous.
- ❖ **what is not**
- ❖ **consider doing**
- ❖ **consider how**
- ❖ **consider not**

## Examples



# Compositing is Culture

- \* Framed in the wiki as **Almost Everything Has a Place**
- \* **situated in community**
- \* **support wilder and diverse practices and community**
- \* **builds off the learner-beginner-artist focus of Processing** ♦

# Community

- \* Framed in Place
- \* situated in
- \* support w
- community
- \* builds off
- Processing
- love2d.org



thing Has a  
and  
t focus of



# Documentation

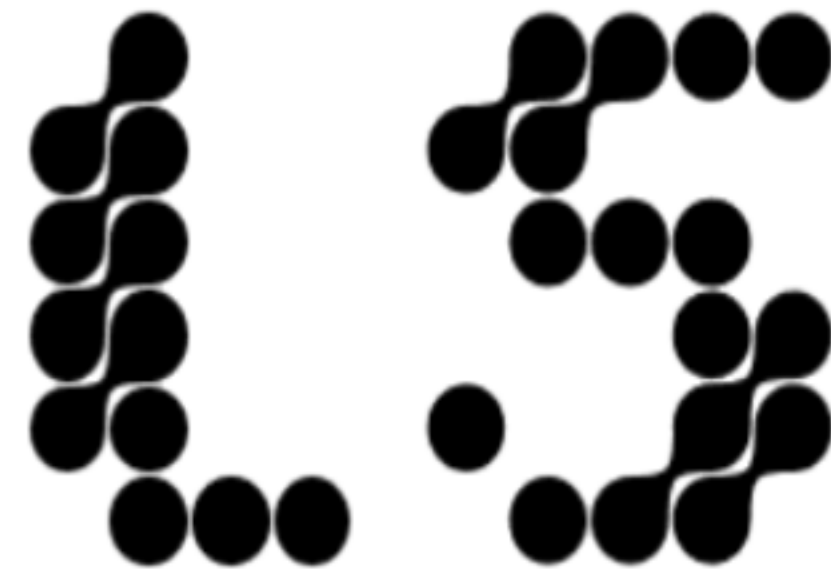
- \* **step-by-step install instructions, tutorials, examples, more than 200 reference pages**
- \* **also built with permacomputing principles: lightweight, fast, works on older computers and browsers.**

we are still working to improve install instructions



Document

Welcome to L5



- \* step-by-step examples, r
- \* also built lightweight browsers.

L5 is a fun, fast, cross-platform, and lightweight implementation of the Processing API in Lua. It is a free and open source coding library to make interactive artwork on the computer, aimed at artists, designers, and anyone that wants a flexible way to prototype art, games, toys, and other software experiments in code.

L5 is designed to work cross-platform, including on desktop, phone, and tablet. Beyond running fast on modern machines, L5 is optimized for older and lower-powered devices, minimizing resource usage to keep creative coding accessible to everyone. This helps with our goal of building resilient, long-lasting software projects. L5 is built in Lua, a robust but lightweight, long-running, lightning-fast, extensible language.

orials,  
ages  
ciples:  
puters and

we are still working to improve installation instructions

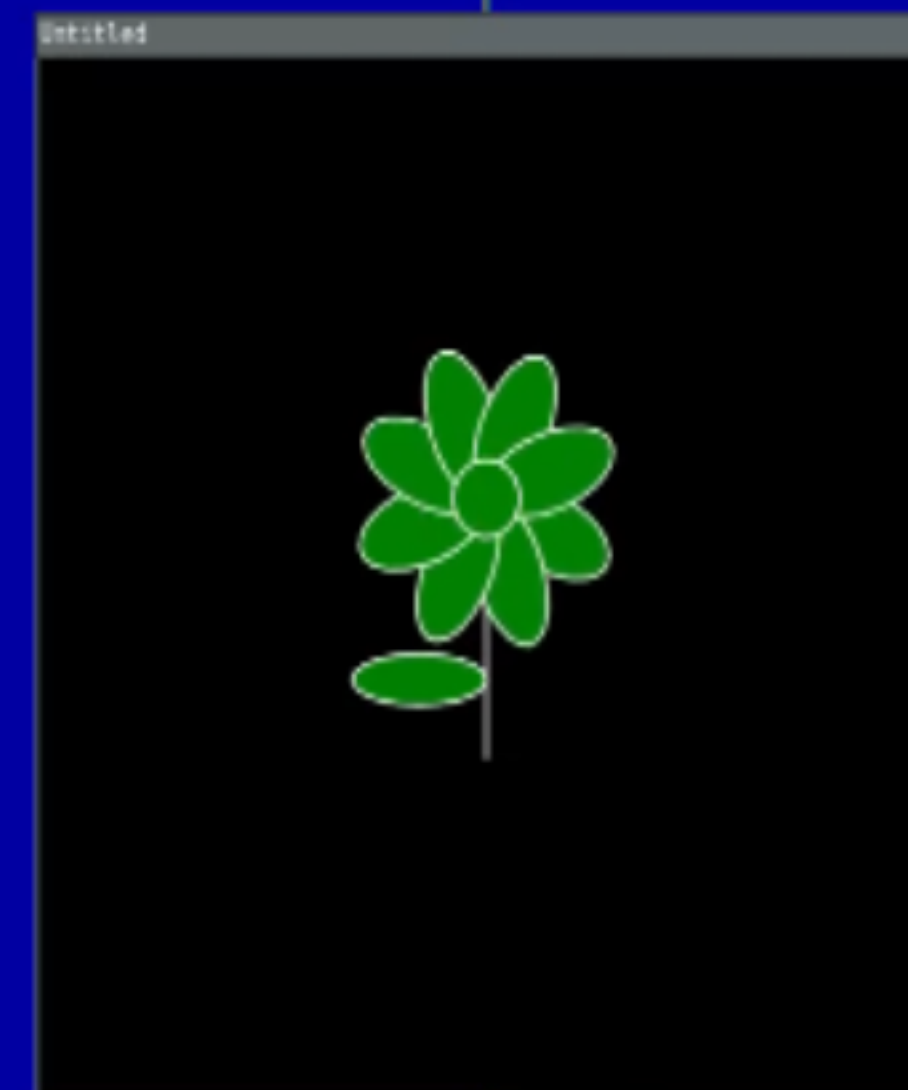
## A basic L5 sketch

```
require('L5')  
function setup()  
  fullScreen()  
  describe('draws black circles at mouse location')  
end  
function mouseDragged()  
  circle(mouseX, mouseY, random(30,300))  
end ♦
```

A basic

```
require('L5')
function setup()
  fullscreen()
end
function mousePressed()
end
```

```
1 require("L5")
2
3 -- Permacomputing flower logo recreation?
4
5 function setup()
6   size(400, 400)
7 end
8 function draw()
9   background(0)
10  fill('green')
11  stroke('white')
12
13  push()
14  translate(width/2, height/2 - 30)
15
16  -- Draw stem
17  line(0, 0, 0, 100)
18
19  rotate(radians(frameCount))
20
21  local numPetals = 8
22  local petalLength = 60
23  local petalWidth = 30
24
25  push()
26  for i = 0, numPetals - 1 do
27    push()
28    rotate((i / numPetals) * TWO_PI)
29    circle(mouseX, mouseY, petalWidth)
```



se location')

00))

THANK YOU!

Website: <https://L5lua.org>

Forum: <https://discourse.processing.org>  
(Choose L5 category)



We're looking for contributors! (documentation, testing, bug fixing, etc)