

# Hello World

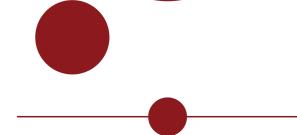


# Intro Kickstart-kursus i programmering 2024

Daniel Spikol

[ds@di.ku.dk](mailto:ds@di.ku.dk)

DIKU Københavns Universitet



KØBENHAVNS  
UNIVERSITET

# The Team

- Daniel Spikol
- Alexander Mittet
- Thea Nyks Meggele
- Jeppe Fræhr Møller
- Mathilde Kjer Parlo
- Andrea Sebro Rasmussen
- Casper Brahe Rützou
- Viktoria Silke Sofia Strandbygaard
- Lucas Østergaard Jarmer

# Inclusive DIKU

- Welcome to DIKU. Our goal is to promote excellence in computer science education and research.
- To strive for a respectful, inclusive, diverse environment and encourage open and critical academic discussion.
- We strive to create a welcoming and respectful environment.

## OPEN & ACCOMMODATING

Being open and respectful of differing opinions, viewpoints, and experiences



## INCLUSIVE

Focusing on what is best not just for us as individuals, but for the overall community

## CONSTRUCTIVE

Giving and accepting constructive feedback



## RESPONSIBLE

Accepting responsibility and apologizing to those affected by our mistakes, and learning from the experience

## RESPECTFUL

Respecting DIKU's facilities by following the current regulations of the Faculty of Science and DIKU



# **Velkommen!**

- Presentation of Activities
- Why and what is the Kickstart course?

# Overall Plan for the week

Day 1 Mon	6h 00m	Day 2 Tues	6h 00m	Day 3 Weds	6h 00m	Day 4 Thurs	6h 00m	Day 5 Fri	6h 00m
<p>09:00 Welcome and Introducti... 15m</p> <p>09:15 Greeting Activity 15m</p> <p>09:30 Team Building and Icebr... 1h 30m</p> <p>11:00 Installation, wifi, setup, ... 45m</p> <p>11:45 Lunch 45m</p> <p>12:30 Intro to Processing 10m</p> <p>12:40 Worksheet: #0 The First... 1h 00m</p> <p>13:40 Worksheet: #1 1h 20m</p>	15:00	<p>09:00 Hello and Intro to peer p... 15m</p> <p>09:15 Intro to setup/draw + gl... 15m</p> <p>09:30 Break 15m</p> <p>09:45 Worksheet #2 1h 00m</p> <p>10:45 Worksheet #3 1h 00m</p> <p>11:45 Lunch 45m</p> <p>12:30 Pair Programming Practi... 15m</p> <p>12:45 Worksheet: #3 1h 15m</p> <p>14:00 Break 15m</p> <p>14:15 Peer Feedback 30m</p> <p>14:45 Coding and end of day 15m</p>	15:00	<p>09:00 Intro talk on Conditionals 30m</p> <p>09:30 Break 10m</p> <p>09:40 Worksheet #4 and #5 1h 50m</p> <p>11:30 Peer Feedback Practice 15m</p> <p>11:45 Lunch 50m</p> <p>12:35 Projects and Brainstorm... 30m</p> <p>13:05 Project Time Brainstorm... 1h 55m</p>	15:00	<p>09:00 Creative Coding Valkari... 30m</p> <p>09:30 Worksheets #5-6 &amp; Proj... 1h 00m</p> <p>10:30 Break 15m</p> <p>10:45 Worksheets 1h 15m</p> <p>12:00 Lunch 1h 00m</p> <p>13:00 Project Time and/or Wor... 20m</p> <p>13:20 Project Time and/or Wor... 40m</p> <p>14:00 break 10m</p> <p>14:10 Project Time and/or Wor... 50m</p>	15:00	<p>09:00 Summary of week 45m</p> <p>09:45 Project time 2h 15m</p> <p>12:00 Lunch 45m</p> <p>12:45 Final Project Prep 45m</p> <p>13:30 Presentations 1h 15m</p> <p>14:45 Survey 15m</p>	15:00
<p>+ Add block</p>		<p>+ Add block</p>		<p>+ Add block</p>		<p>+ Add block</p>		<p>+ Add block</p>	

# Monday's Plan

- kl. 9:00-9:45 → Welcome and Introduction
- kl. 10:00-11:15 → Team Building Activities, Mini-Lecture, and Coding
- KL 11:15-12:00 → Processing Introduction
- kl. 12:00 - Lunch → Options (Bio Kantine...)
- kl. 13:00-13:30 → Aud.06 Intro to Processing
- kl. 13:30-15:00 → Team rooms - Intro to Processing
- kl. 14:45-15:00 → Check-in with your Teams

# Intended Fun Outcomes

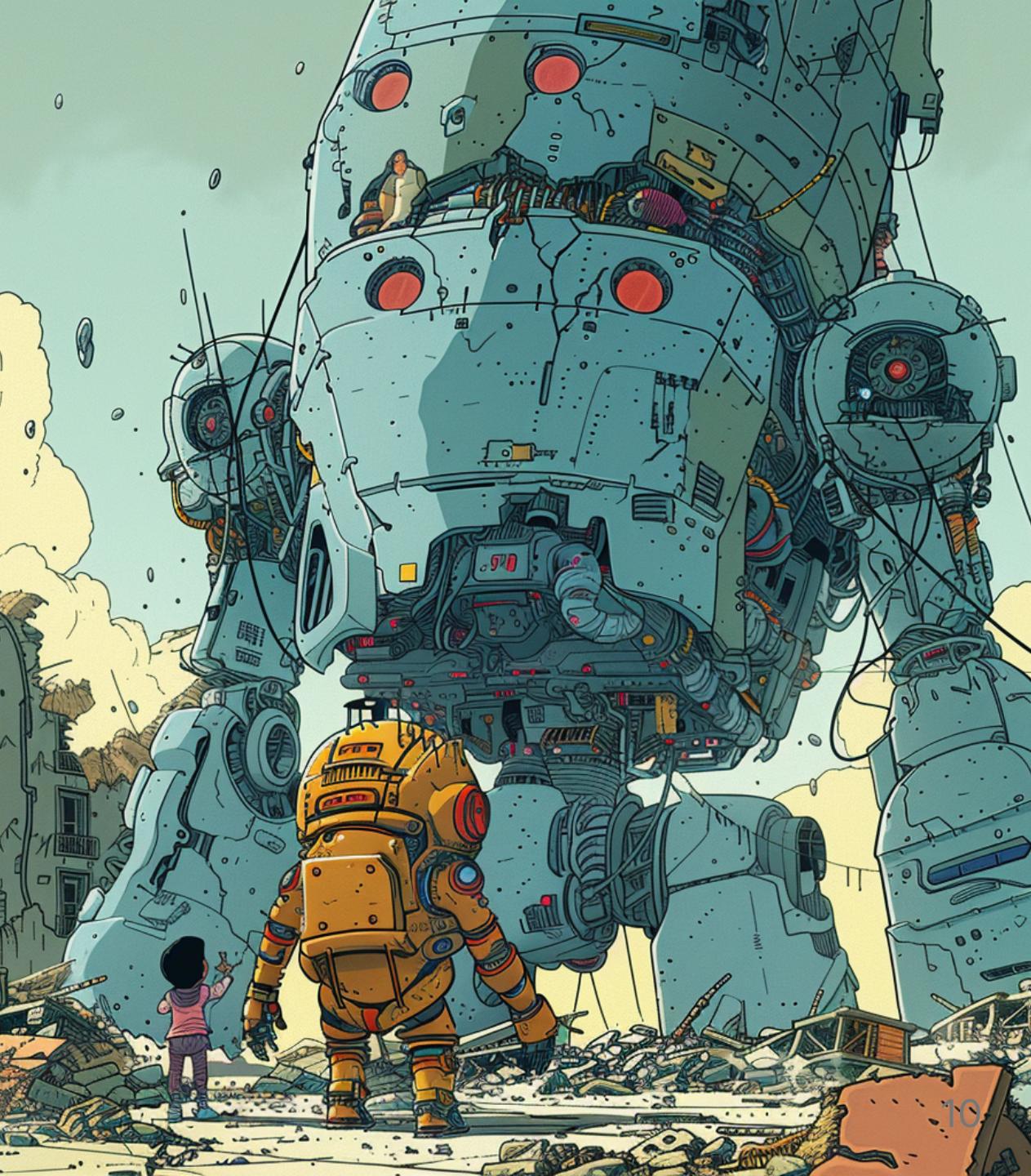
- Why IFOs and not Intended Learning Outcomes
  - Intended Learning Outcomes (ILOs) define what a learner will have acquired and will be able to do upon completing their courses and studies.

# Intended Fun Outcomes

- Meet people and collaborate
- Explore programming with a focus on creating games
- Have fun and take chances through creativity
- Take control of your education

# Meet people and collaborate

- Design Thinking
- Collaboration
- Reframing



# Teams and Instructors

- **Team Alpha:** Alexander and Thea → room
- **Team Bravo:** Mathilde and Jeppe → room
- **Team Charlie:** Casper and Andrea → room
- **Team Delta:** Viktoria and Lucas → room

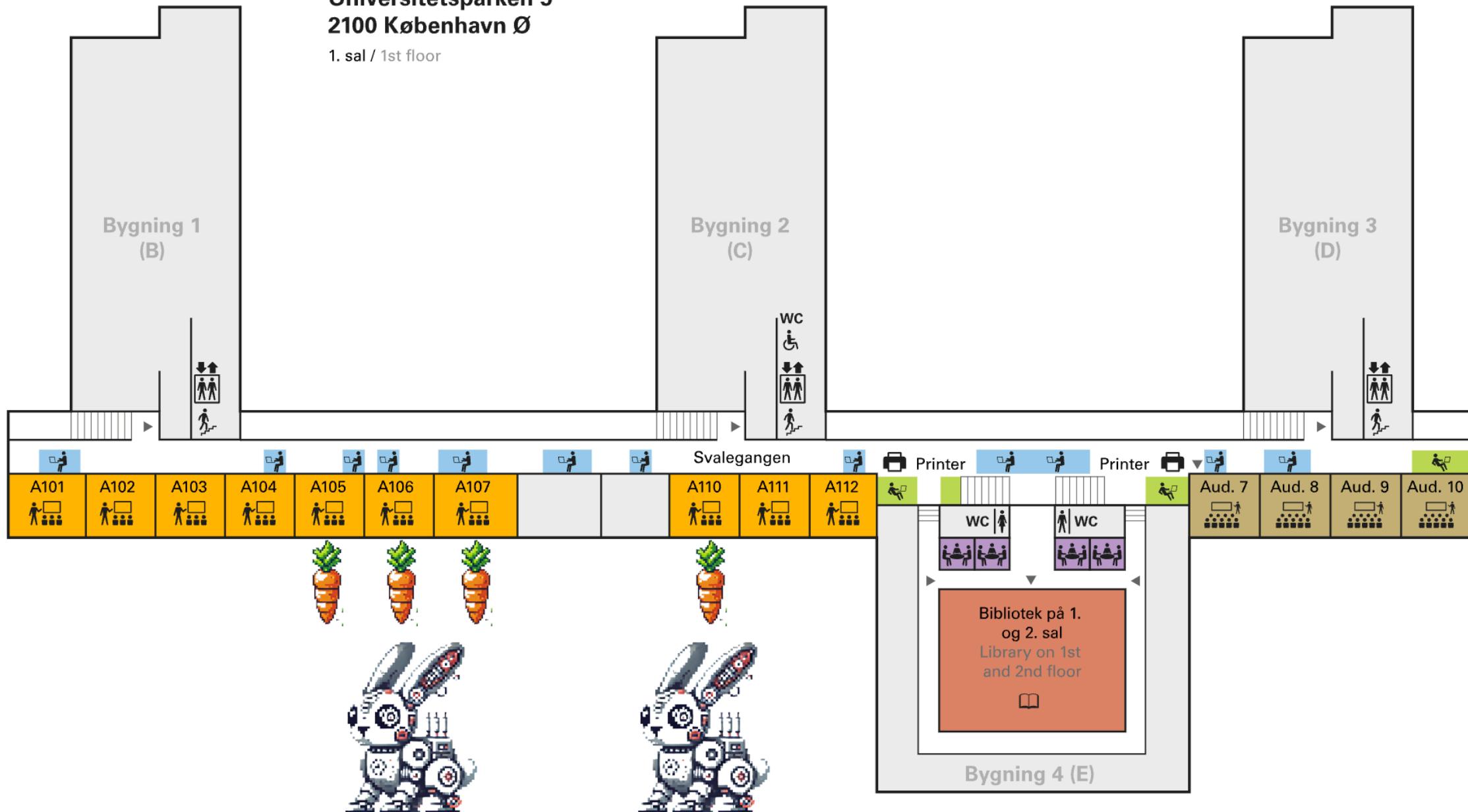
# Challenge Time

- Design Thinking
- Collaboration
- Reframing

# HCØ Workrooms

H.C. Ørsted Instituttet  
Universitetsparken 5  
2100 København Ø

1. sal / 1st floor



# Time for a Team Challenge



# Marshmallow Challenge Reflection



- [TED TALK Tom Wujec](#)
- Constant prototyping as a problem-solving method.
- **Getting the Design Right and the Right Design** → Buxton, William (2007). *Sketching user experiences: getting the design right and the right design*. Amsterdam: Elsevier/Morgan Kaufmann

# Internet Access

## Aktiver KU-bruger

- Log på [mit.ku.dk](https://mit.ku.dk) med NemID og find midlertidig pinkode. Gå til [kunet.dk](https://kunet.dk) og tryk "Første gang du logger på".
- Log på med den midlertidige kode og CPR nummer uden bindestreg.
- Aflæs KU brugernavn (fx abc123) og angiv password.

## Eduroam WiFi

- Kræver KU bruger og password
- Log på med brugernavn: [abc123@ku.dk](mailto:abc123@ku.dk)

## Alternativt: KU Guest WiFi

- Opret 24 timers konto ved at indtaste navn, mail, mobilnummer.
- Modtag kode via SMS.

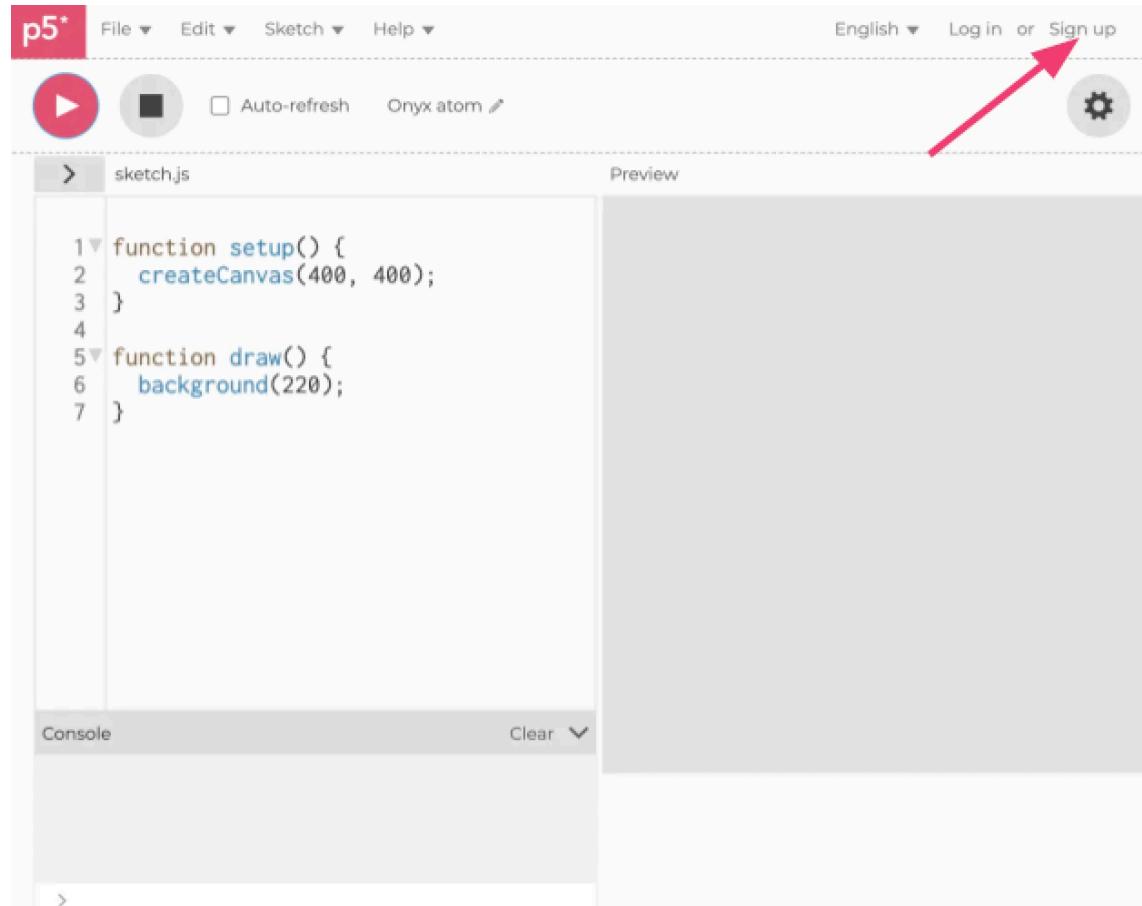
## Installation af p5\*js

<https://p5js.org/>



# Web Editor Steps

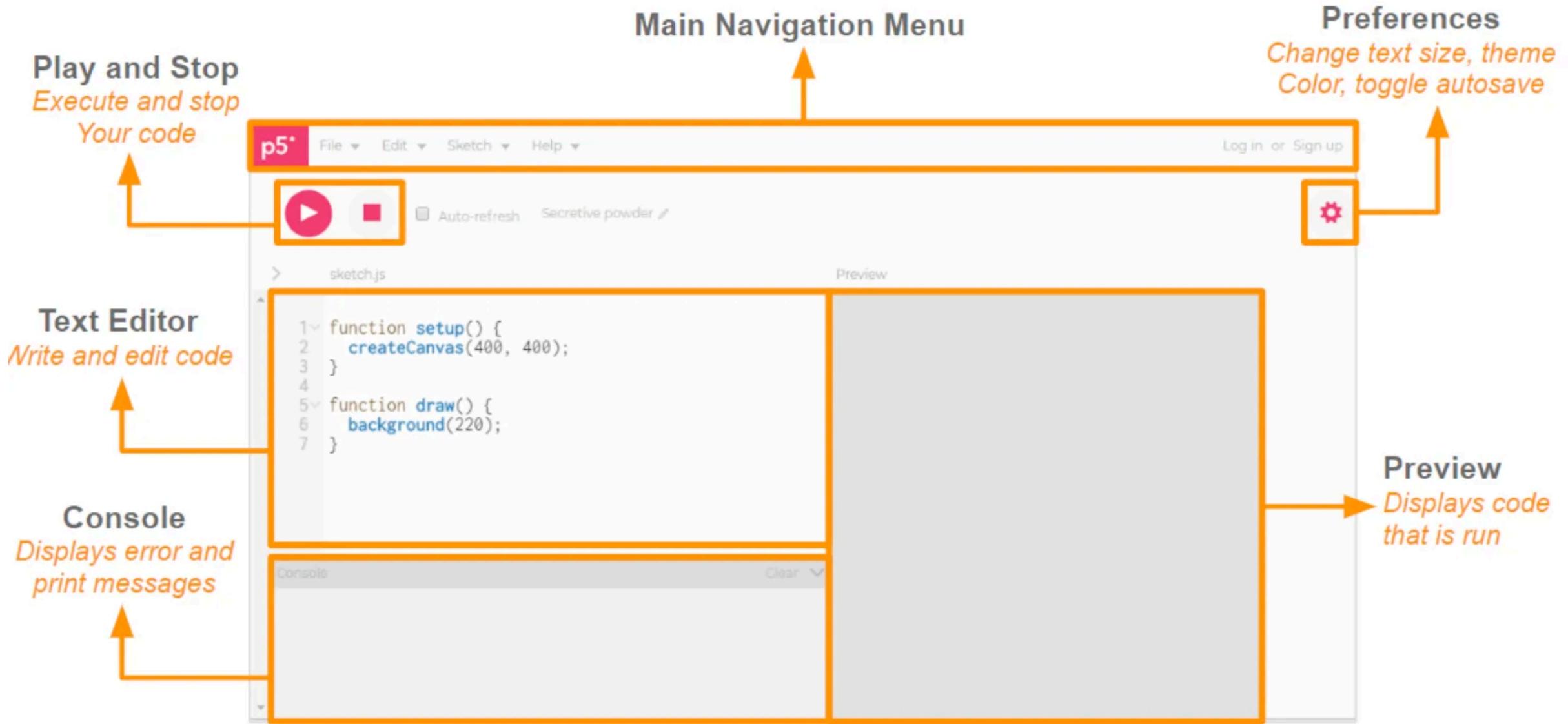
1. Access to the Internet  and web browser 
2. The p5.js Web Editor is a website where programmers can write, test, share, or remix p5.js programs without needing to download or configure a code editor on a computer. A code editor makes writing and reading code easier by organizing and coloring text in a way that helps programmers see different parts of the code.
3. See this [Tutorial](#) for getting up to speed if you want.

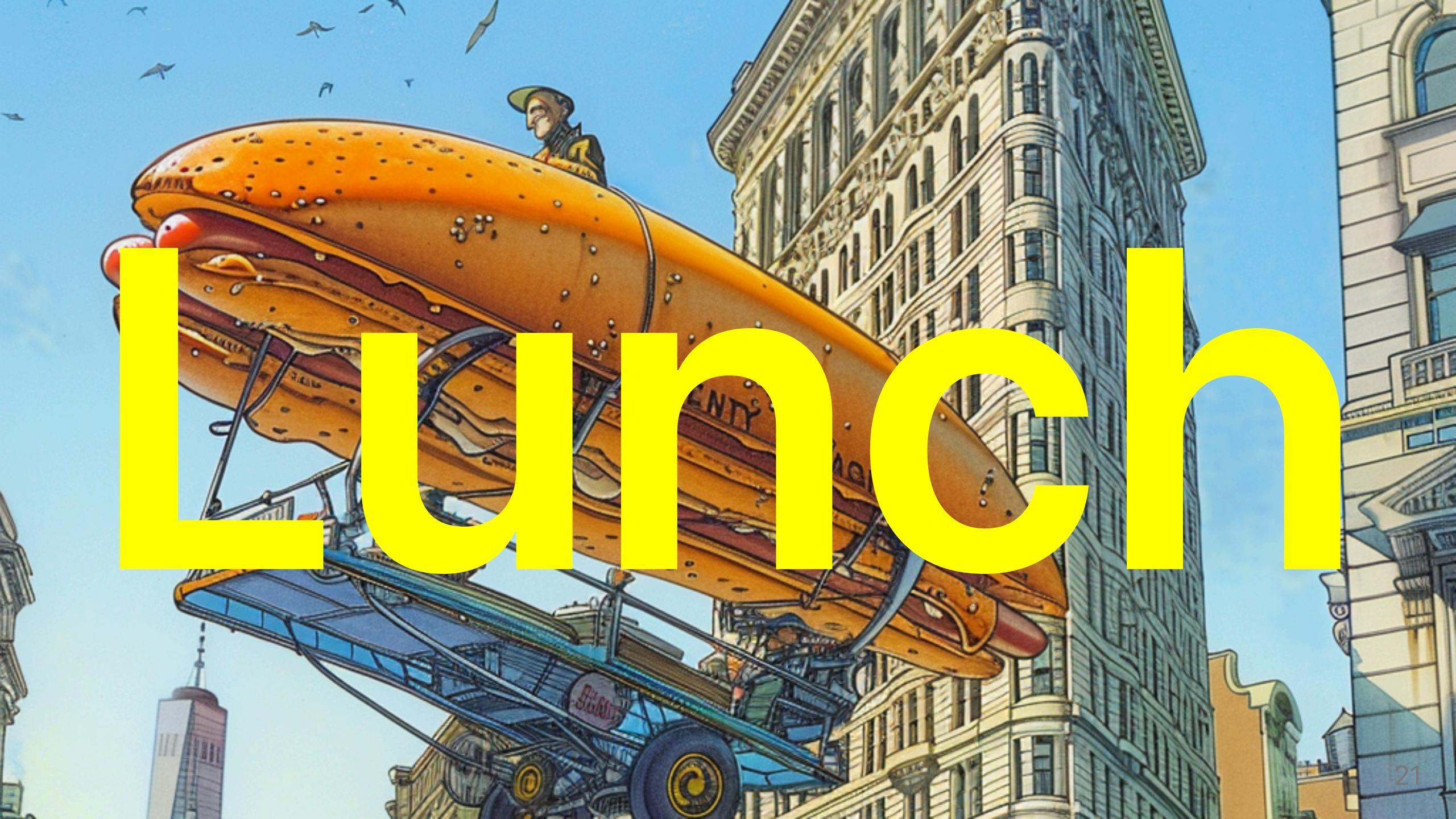


## P5 Sign up

1. You will want to save and share your files
2. You can sign-up with your Email, Google, or GitHub

# p5.js Web Editor Interface

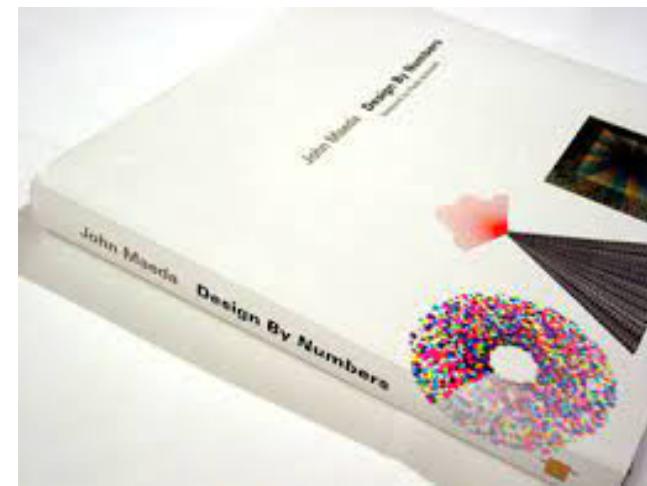


A surreal illustration featuring a massive sandwich flying through a city skyline. The sandwich is enormous, with a golden-brown bun, meat, cheese, and vegetables, riveted together. A small figure wearing a hat and vest sits atop the sandwich, which is suspended by mechanical arms from a large blue mechanical device. In the background, there are tall, detailed buildings, including one with arched windows and another with a prominent tower. The sky is blue with a few birds. The word "Lunch" is overlaid in large, bold, yellow letters.

Lunch

# Processing History

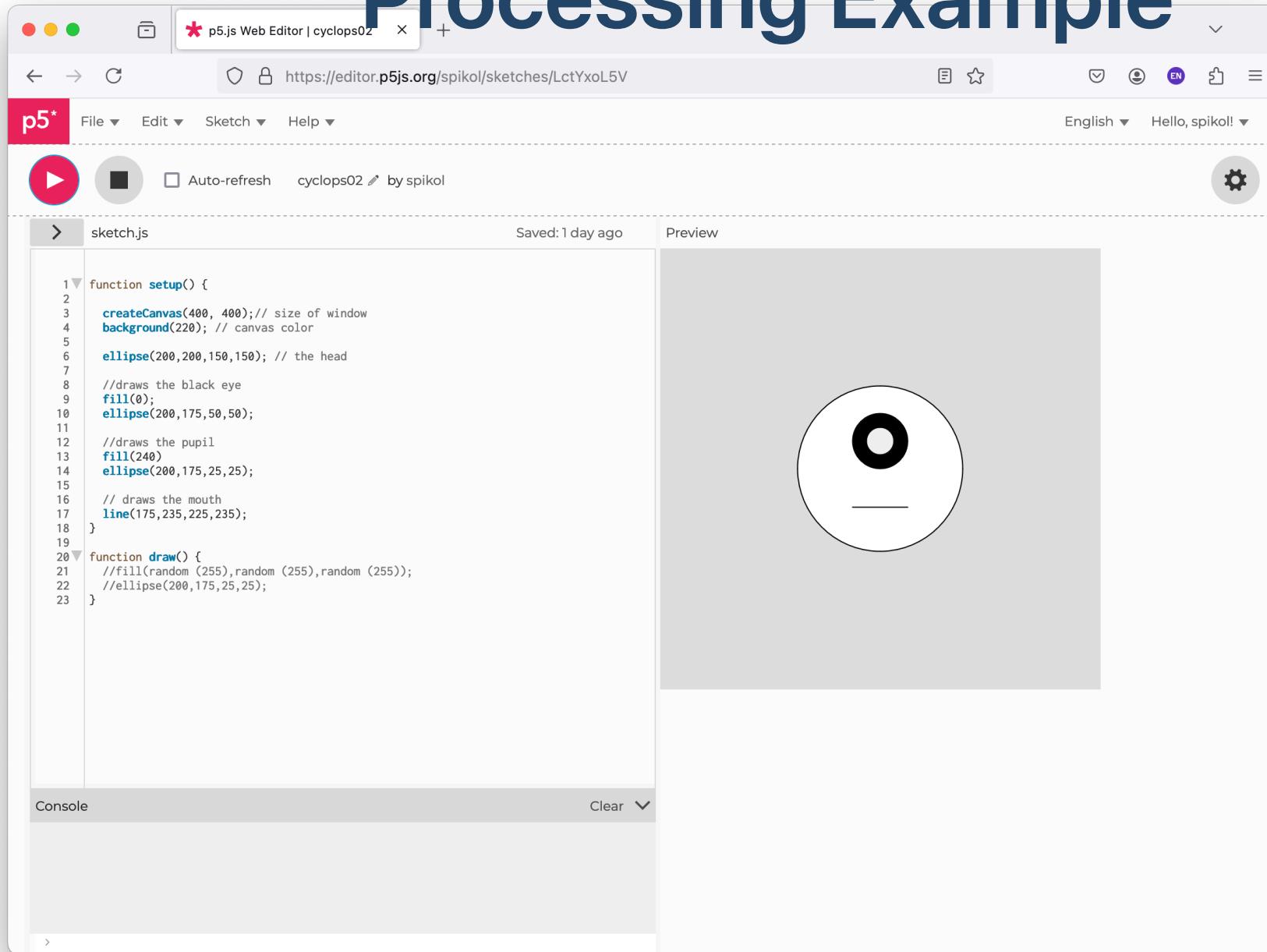
- M.I.T. Media Lab Casey Reas & Ben Fry
- Design By Numbers - Maeda, John. - Design by numbers / John Maeda.. - 1999. - ISBN: 0262133547
- Creative Programming



# What is Processing

- Processing is a flexible software sketchbook and a language for learning code. Since 2001, Processing has promoted software literacy within the visual arts and visual literacy within technology. Thousands of students, artists, designers, researchers, and hobbyists use Processing for learning and prototyping.
- Processing Development Environment → PDE
- Programs are called Sketches
- The PDE consists of a simple text editor for writing code, a message area, a text console, tabs for managing files, a toolbar with buttons for common actions, and a series of menus.

# Processing Example



The screenshot shows the p5.js Web Editor interface. At the top, the title bar reads "p5\* p5.js Web Editor | cyclops02" and the address bar shows the URL "https://editor.p5js.org/spikol/sketches/LctYxoL5V". The editor window has a red header bar with "File", "Edit", "Sketch", and "Help" menus. Below the header is a toolbar with a play button, a square button, and an "Auto-refresh" checkbox. The main workspace is titled "sketch.js" and contains the following code:

```
1 function setup() {
2   createCanvas(400, 400); // size of window
3   background(220); // canvas color
4
5   ellipse(200, 200, 150, 150); // the head
6
7   //draws the black eye
8   fill(0);
9   ellipse(200, 175, 50, 50);
10
11  //draws the pupil
12  fill(240)
13  ellipse(200, 175, 25, 25);
14
15  // draws the mouth
16  line(175, 235, 225, 235);
17
18 }
19
20 function draw() {
21   //fill(random(255), random(255), random(255));
22   //ellipse(200, 175, 25, 25);
23 }
```

The preview area on the right shows a gray square containing a white circle with a black outline and a black dot in the center. A thin horizontal line extends from the bottom of the circle.

# Processing Parts 1

## 1. Basics of Processing IDE:

- **Interface:** The Processing IDE has a simple interface. There's an area for writing code, buttons for running and stopping sketches, and a message area below for error messages and other notifications.
- **Sketch:** In Processing, each project is called a "sketch". A sketch is a combination of code, data, and output.

## 2. Language:

Processing uses a variant of the Java language. It's designed to be beginner-friendly, with simpler functions and setup to create visual and interactive projects quickly.

# Structure of a Basic Sketch:

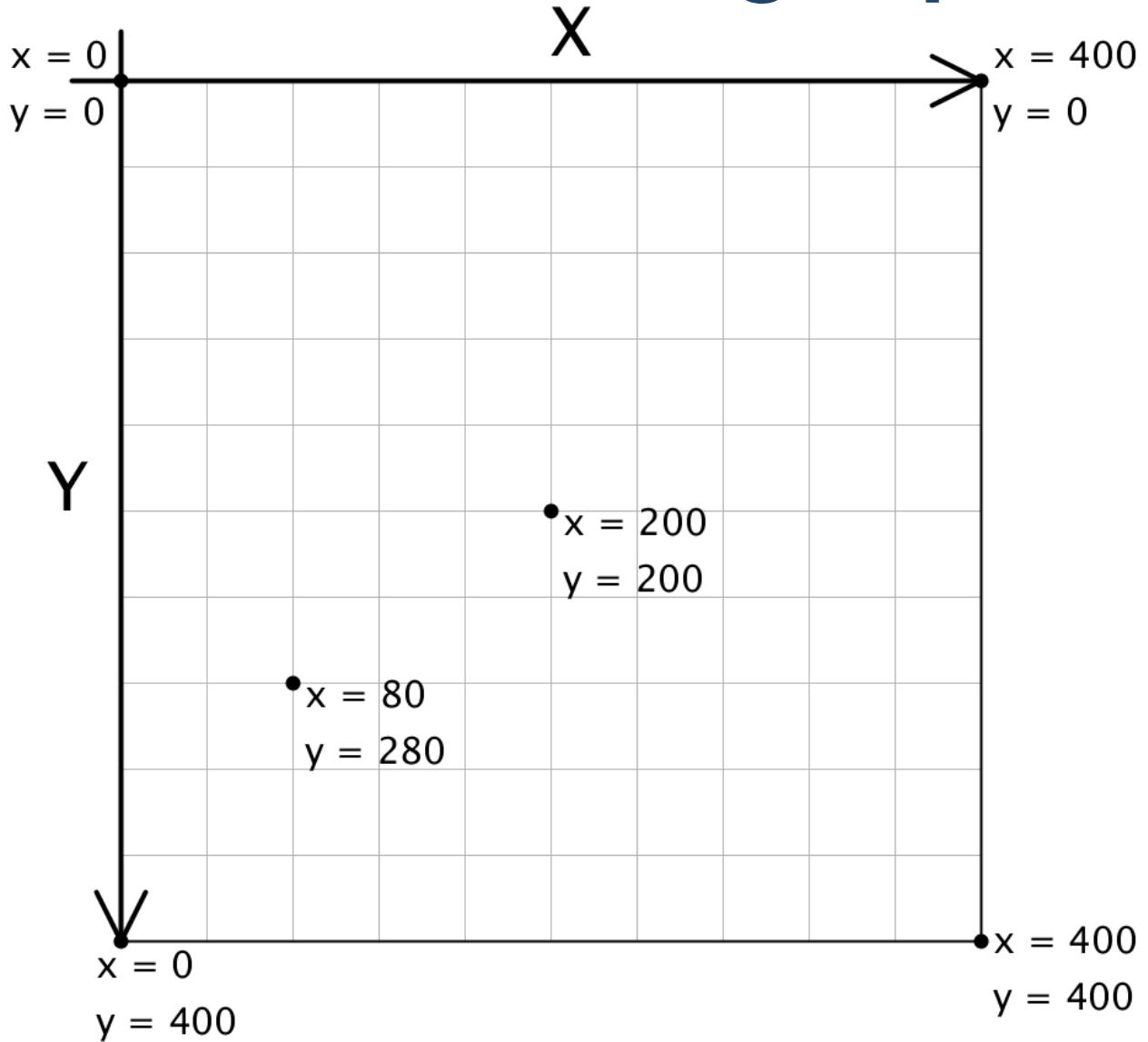
- **setup() function:** This is executed once when the sketch starts. It's commonly used to define initial environment properties such as screen size and background color.
- **draw() function:** This runs repeatedly after setup(). It's used for continuously running code, such as animation or checking for input.

**4. Running the Sketch:** The sketch is compiled and run when you click the "Run" button (or press Ctrl+R/Cmd+R). The visual output is displayed in a separate window.

# Enough Lecture: Today's IFOs

- You will get a sense of the PDE shortly
- How to **draw and color** with Processing
- What are **Variables** as simple as they get

# Drawing in p5



# Cartesian Coordinates in p5.js

## Canvas Grid:

- Origin  $(0, 0)$  is at the top-left corner.
- X-axis increases to the right.
- Y-axis increases downward.

## Using Coordinates:

- Position shapes using  $(x, y)$  coordinates.

## Example coordinates:

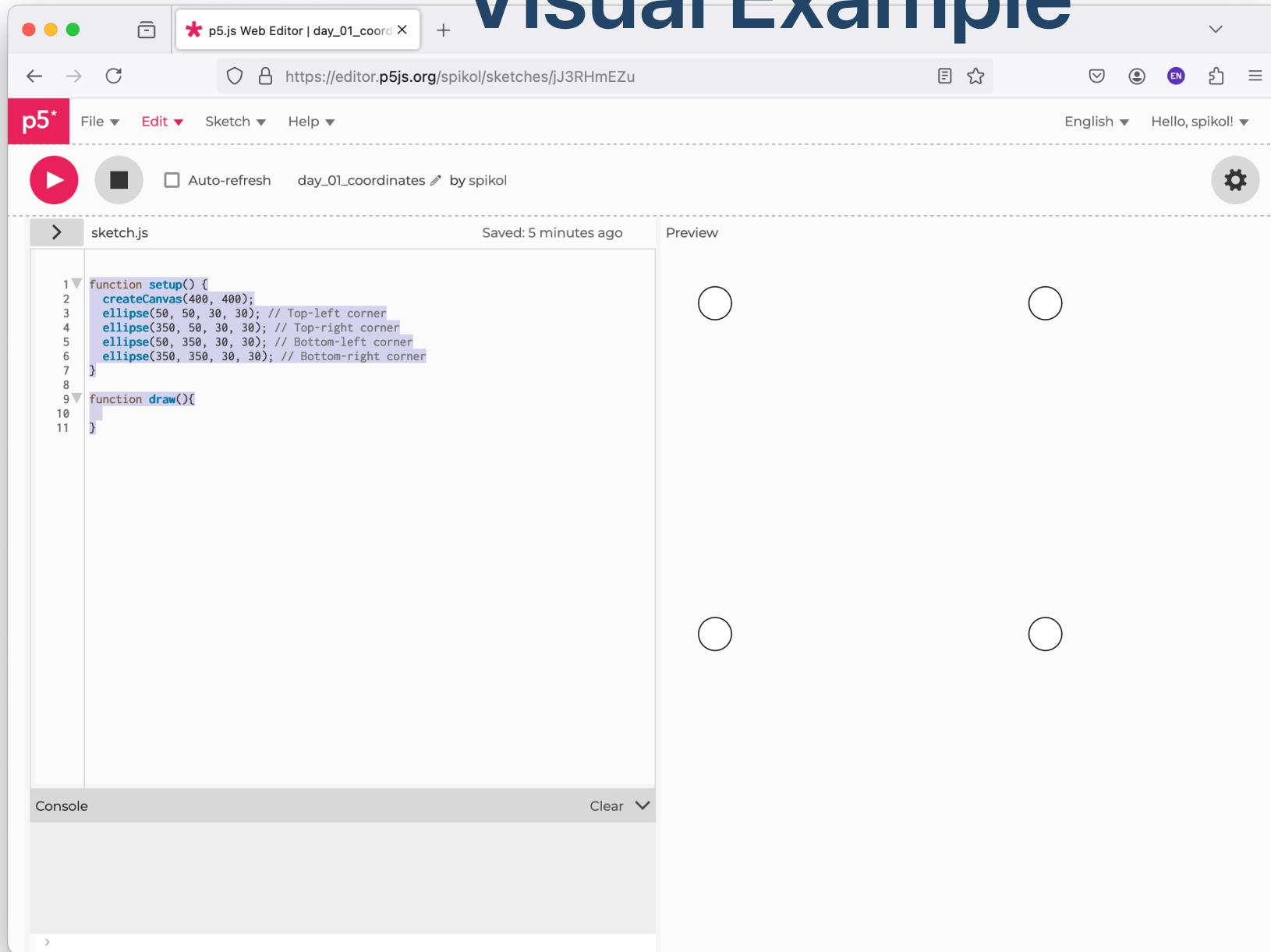
- $(0, 0)$  - Top-left corner.
- $(400, 400)$  - Bottom-right corner (for a 400x400 canvas).

# Code Example

```
function setup() {
  createCanvas(400, 400);
  ellipse(50, 50, 30, 30); // Top-left corner
  ellipse(350, 50, 30, 30); // Top-right corner
  ellipse(50, 350, 30, 30); // Bottom-left corner
  ellipse(350, 350, 30, 30); // Bottom-right corner
}

function draw(){
}
```

# Visual Example



# Variables

- At its core, a **variable in programming** is like a container or a storage box that holds data.
- **Variables** are names that hold **values**.
- Imagine you have a box in real life where you put an apple; you might label that box "fruit" so you know it contains a fruit. In programming, the box is the variable, the apple is the data, and the label "fruit" is the variable's name.
- Keep in mind that, unlike real life, if you put an orange into your variable.

# Scoping Variables

- **Global Variables:**
  - Declared outside any function.
  - Accessible from any part of the program.
- **\*\*Local Scope**
  - Variables declared within function or blocks

