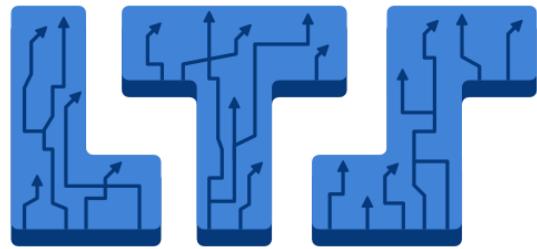


1 - Welcome

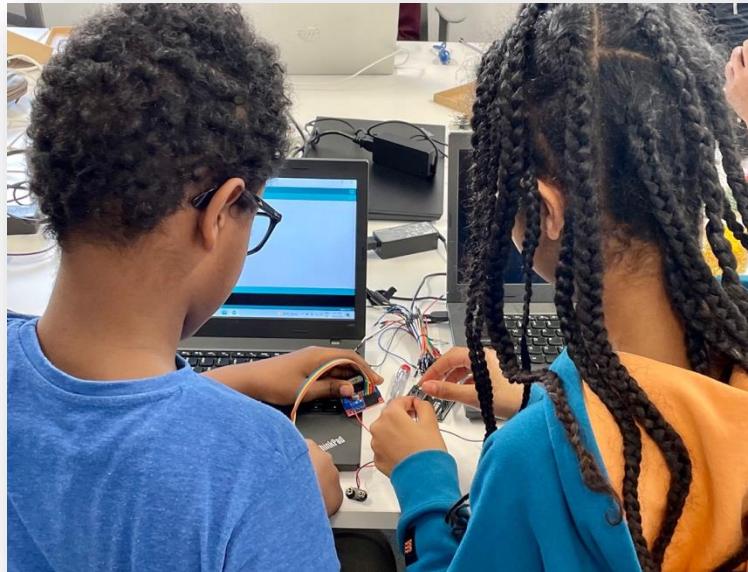




**Who are we.
What we will do.
Some Rules.**

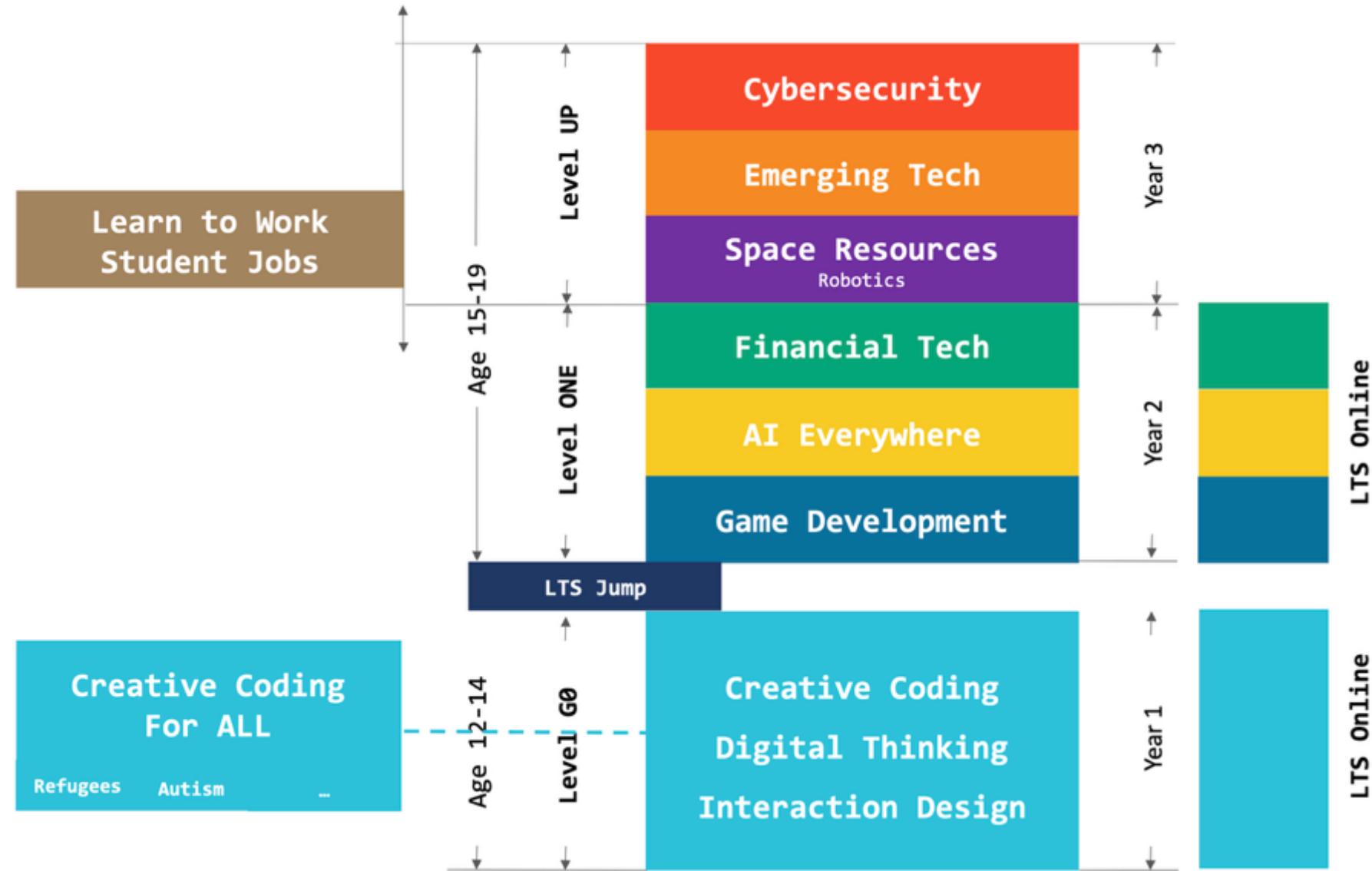


is an educational program created in 2016
to support the development of
[future] Digital Leaders

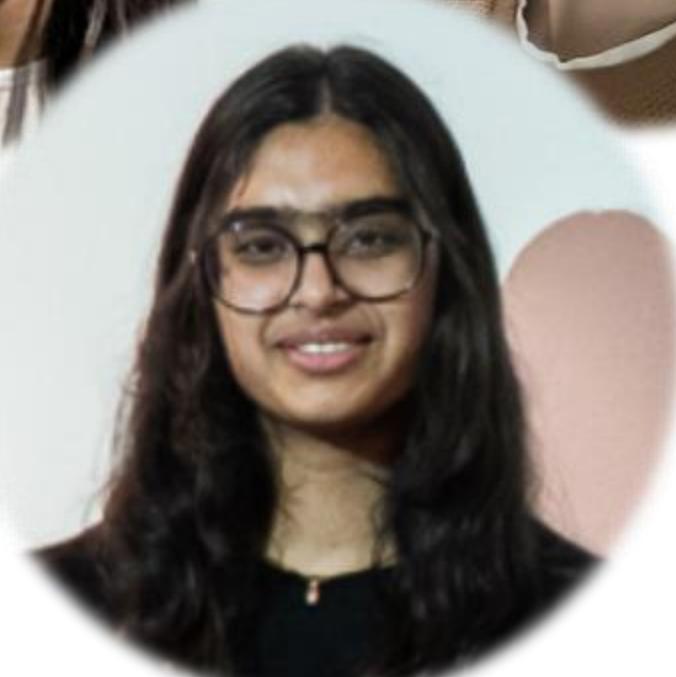


We are a non-profit organization with coaches coming from the tech industry, business and/or creative field. We offer **weekly classes** around **various topics and concepts in technology** and the **digital world** for **12-19 years old** students in different schools in Luxembourg.

Luxembourg Tech School LTS



Your LTS Coaches



What about you ?



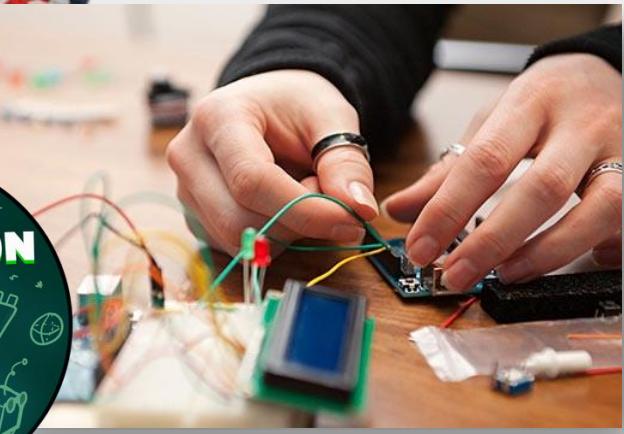
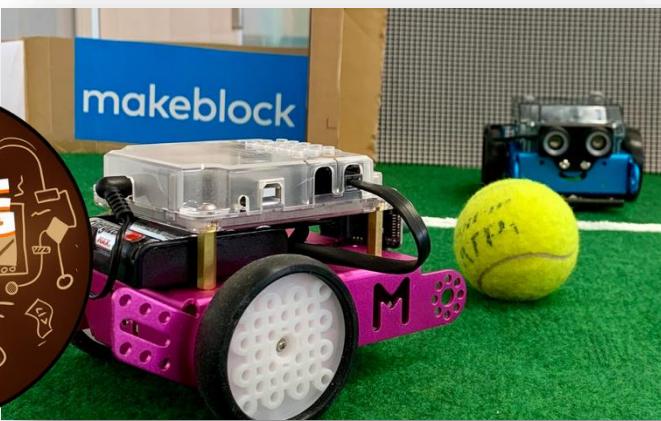
Circle Intro & Warm Up

What's your name?

How are you feeling today?



Level G0: Overview



Level G0: Weekly Classes

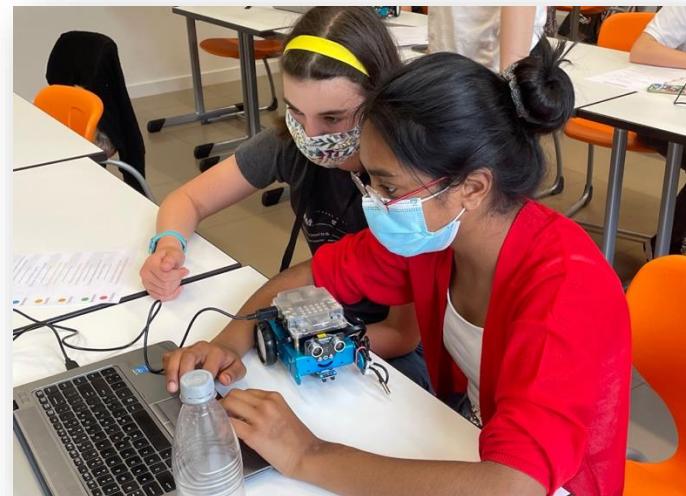
Watch & Listen

We show and explain you everything.



Try yourself

You code and create actively.



Show

Show us your projects.



Communication: Teams

- 1 Open Microsoft Teams



- 2 Look for your LTS Group



Communication: Teams

The screenshot shows the Microsoft Teams interface for the 'LTS1-LVLG0-1-EN-2020-2021' team. The left sidebar includes Activity, Chat, Teams (selected), Calls, Files, and Apps. The main area is the 'General' channel, which displays a 'Choose where you want to start' section with 'Upload Class Materials' and 'Set up Class Notebook' options. A large orange post from 'KAISER GOOSSENS Sara' dated 25/09/2020 18:59 welcomes users to the LTS Digital Creators group. The post details the class schedule (Tuesdays, 14:15 - 16:00 at Athénée Luxembourg), the meeting room (Room SJ08), and a reminder to bring laptops. Two blue callout boxes with black outlines are overlaid: one pointing to the 'Upload Class Materials' button labeled 'Course Material', and another pointing to the post itself labeled 'Announcements'.

Course Material

Announcements

Welcome!

LTS Digital Creators: Tuesdays, 14:15 - 16:00 at Athénée Luxembourg
Hello everyone LTS1-LVLG0-1-EN-2020-2021! This is your Digital Creators group of the Luxembourg Tech School.

We will start with our first class next Tuesday, 29 September at 14:15-15:45h
Where? Room SJ08 at Athénée Luxembourg, 24 Boulevard Pierre Dupong, Luxembourg
Please bring your laptop with you. If you have asked for an LTS laptop, we will bring it to the first class.

Looking forward to seeing you soon!

New conversation

Weekly Course Material

The screenshot shows a Microsoft Teams interface for a course named "LTS1-LVLG0-1-EN-2020...". The left sidebar includes icons for Activity, Chat, Teams, Assignments, and Apps. The main area displays the "General" tab of the course channel. A large black arrow points to the "Files" tab, which is highlighted with a blue border. Below the tabs are buttons for "+ New", "Upload", "Sync", "Copy link", "Download", and a three-dot menu. The "All Documents" dropdown is set to "General". The file list shows two items:

Name	Modified	Modified By
Class Materials		HOUGARDY Barbar...
WEEK01+02_Draw-with-p5	January 20	KAISER GOOSSENS...

Some general rules



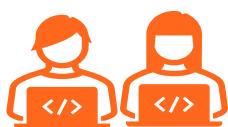
- **Phones goes silenced into the backpack.**

Not on the table, not in the pocket.



- **We start classes on time.**

If you can't come, please inform us before the class starts.



- **We take care of our laptops.**
- **We use our laptops to code.**

Please no games, videos, online chats during the class.



- **If we copy something from someone else, we mention it.**



- **Always ask.**

If there is anything you feel uncomfortable with or have questions, please ask.



- **We support each other.**

We're all here together to learn.

Paperwork



- Parental Media Agreement
- Parental Consent for Online Services



If you don't have them with you today, bring the signed papers back next week!



A short intro about coding

What is Code ?



```
if num_rocks < 31+self.wave:
    while True:
        x, y = randint(0, num_grid_cols-1), randint(1, num_grid_rows)
        if self.grid[y][x] == None:
            self.grid[y][x] = Rock(x, y)
            break
    else:
        game.play_sound("wave")
        self.wave += 1
        self.time = 0
        self.segments = []
        num_segments = 8 + self.wave // 4 * 2
        for i in range(num_segments):
            if DEBUG_TEST_RANDOM_POSITIONS:
                cell_x, cell_y = randint(1, 7), randint(1, 7)
            else:
                cell_x, cell_y = -1-i, 0
            health = [[1,1],[1,2],[2,2],[1,1]][self.wave % 4][i % 2]
            fast = self.wave % 4 == 3
            head = i == 0
            self.segments.append(Segment(cell_x, cell_y, health, fast, h
```



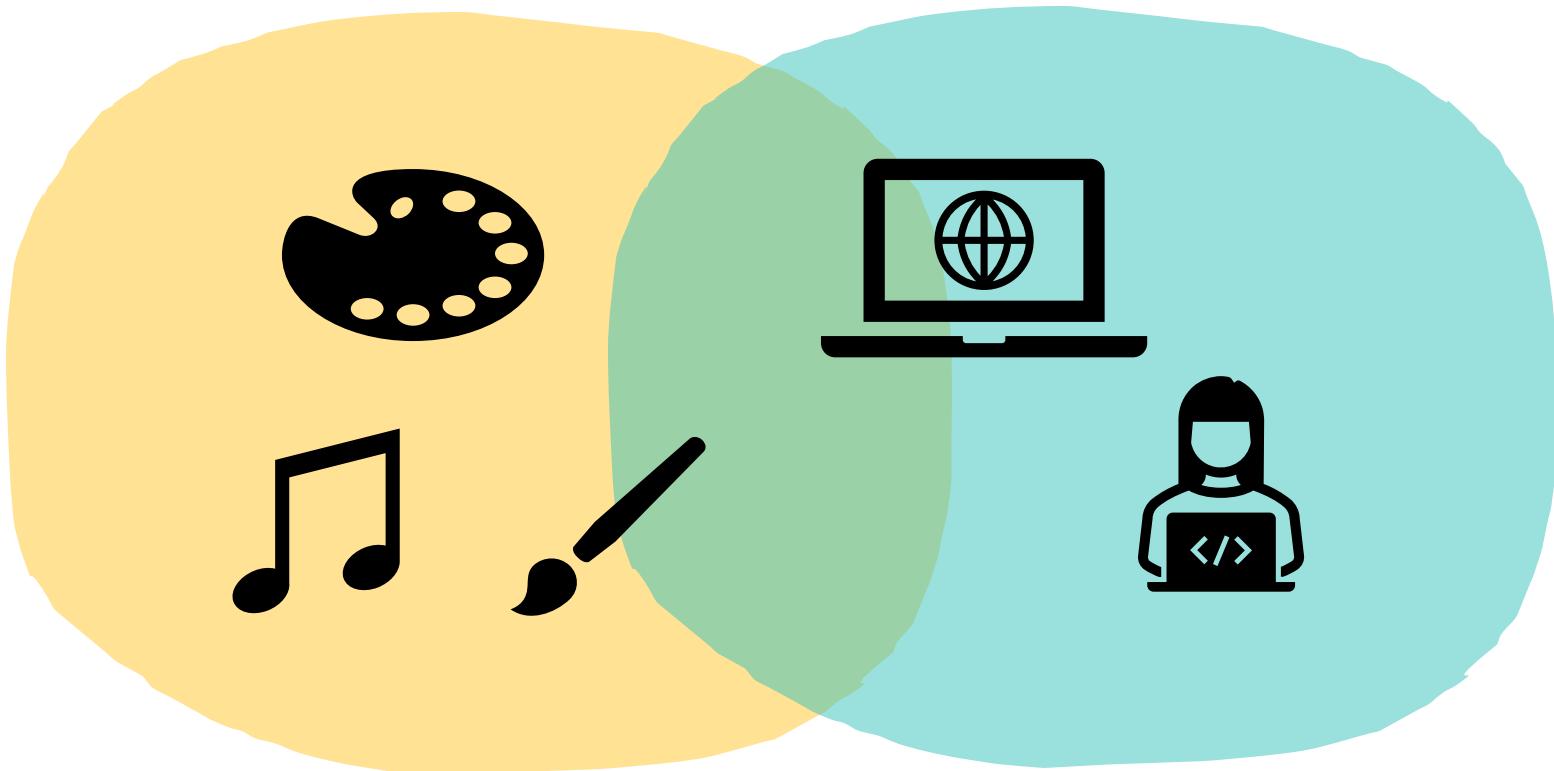
Computer languages are called **code**.

What is Coding ?

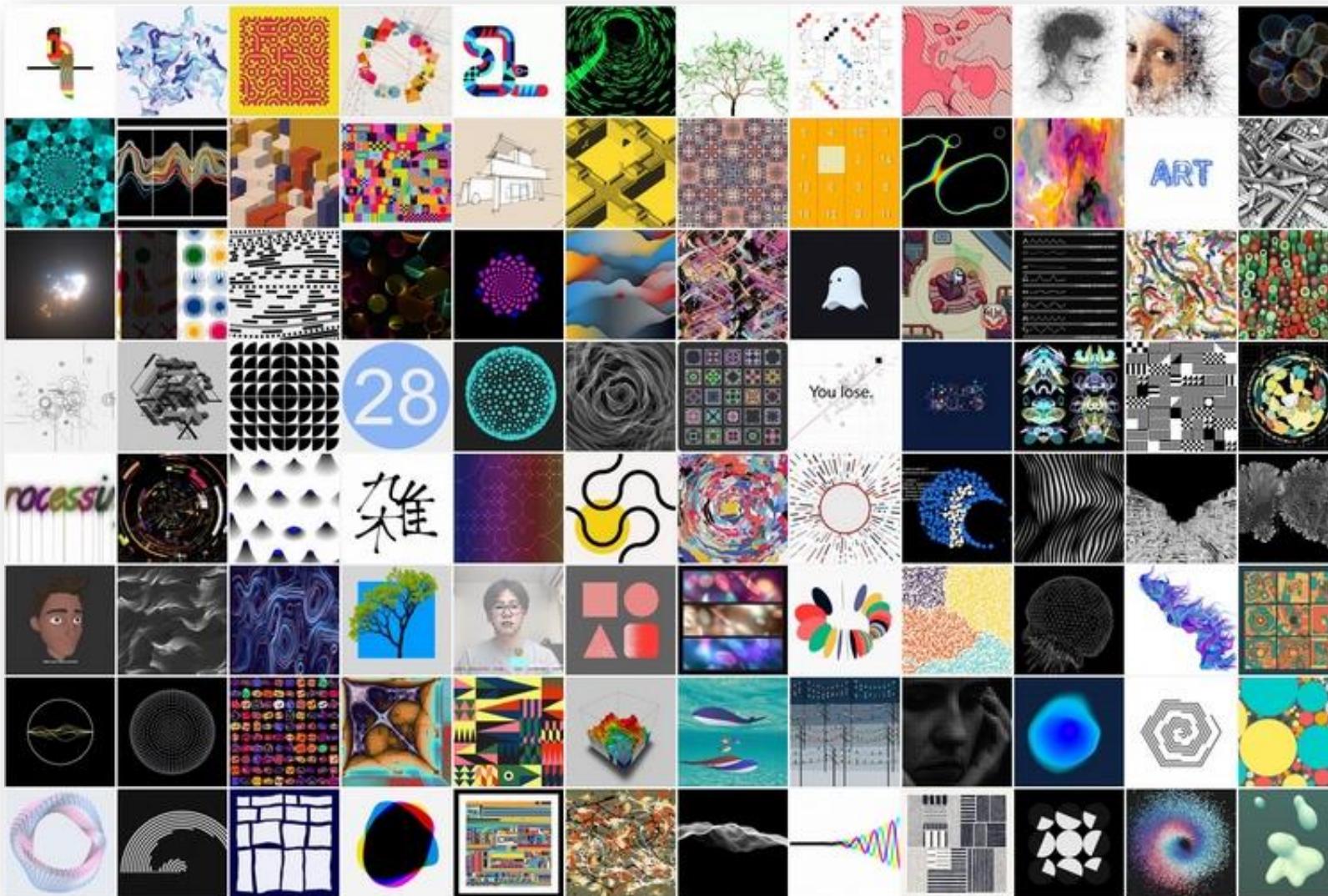
- **Computers** are devices like laptops and smart phones.
- **Programs** are a set of instructions that tell a computer what to do.
- Those instructions are written in **code**.

That means that **coding**
is the process of writing that code.

What is *Creative Coding* ?



What is *Creative Coding* ?



Creative Coding with p5.js

With **p5.js** we will learn and practice some basic concepts of coding:

- **Variables**
- **Functions**
- **Loops**
- **Conditionals**



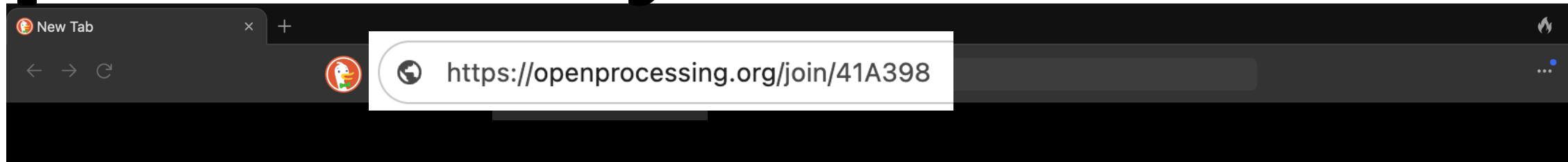
If you understand the basic concepts of coding in p5.js, you can go and apply those concepts in any other coding language like Python, Java, C++ ...

Coding needs a lot of practise



Let's get started!

Open this link in your browser



<http://tiny.cc/LTS-2025>

<https://openprocessing.org/join/41A398>

Account creation

The screenshot shows a web browser window for 'Join - OpenProcessing' at the URL <https://openprocessing.org/join/>. The page title is 'Create an account to join LVLG0 Class'. There are two input fields: 'USERNAME' (labeled '1') and 'PASSWORD' (labeled '2'). Below the fields are 'STAY SIGNED IN' and 'reCAPTCHA' checkboxes. At the bottom, it says 'By clicking Join, you agree to the [Terms of Service](#)' and there is a red 'Join' button.

1

2

USERNAME

PASSWORD

STAY SIGNED IN

I'm not a robot

reCAPTCHA

Join

USERNAME =

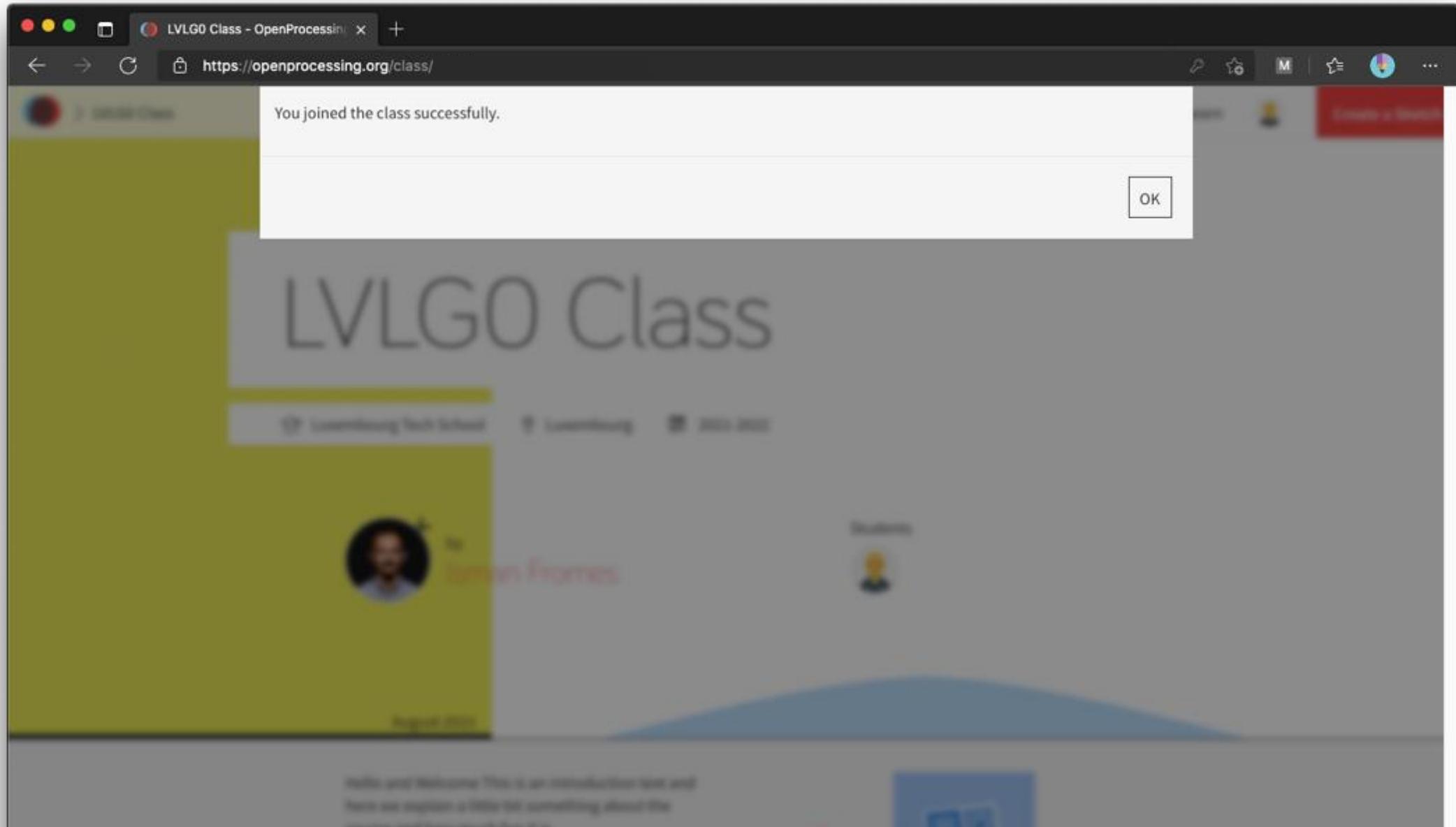
- Your first name
- +
- First letter of your last name
- +
- two random numbers (0-9)

all lowercase!

e.g. maxb78

PASSWORD =
choose a password and make
sure to save it somewhere.

You successfully joined the class



Let's get started!

Close your web browser 🤦

... and open it again ☺

Sign in on Openprocessing.org

- 1. Open a web browser**
- 2. Go to openprocessing.org**
- 3. Click on SIGN IN**
- 4. Use your username and password (created a few seconds ago!)**



Sign in

Don't have an account yet? [Join OpenProcessing](#)

EMAIL OR
USERNAME

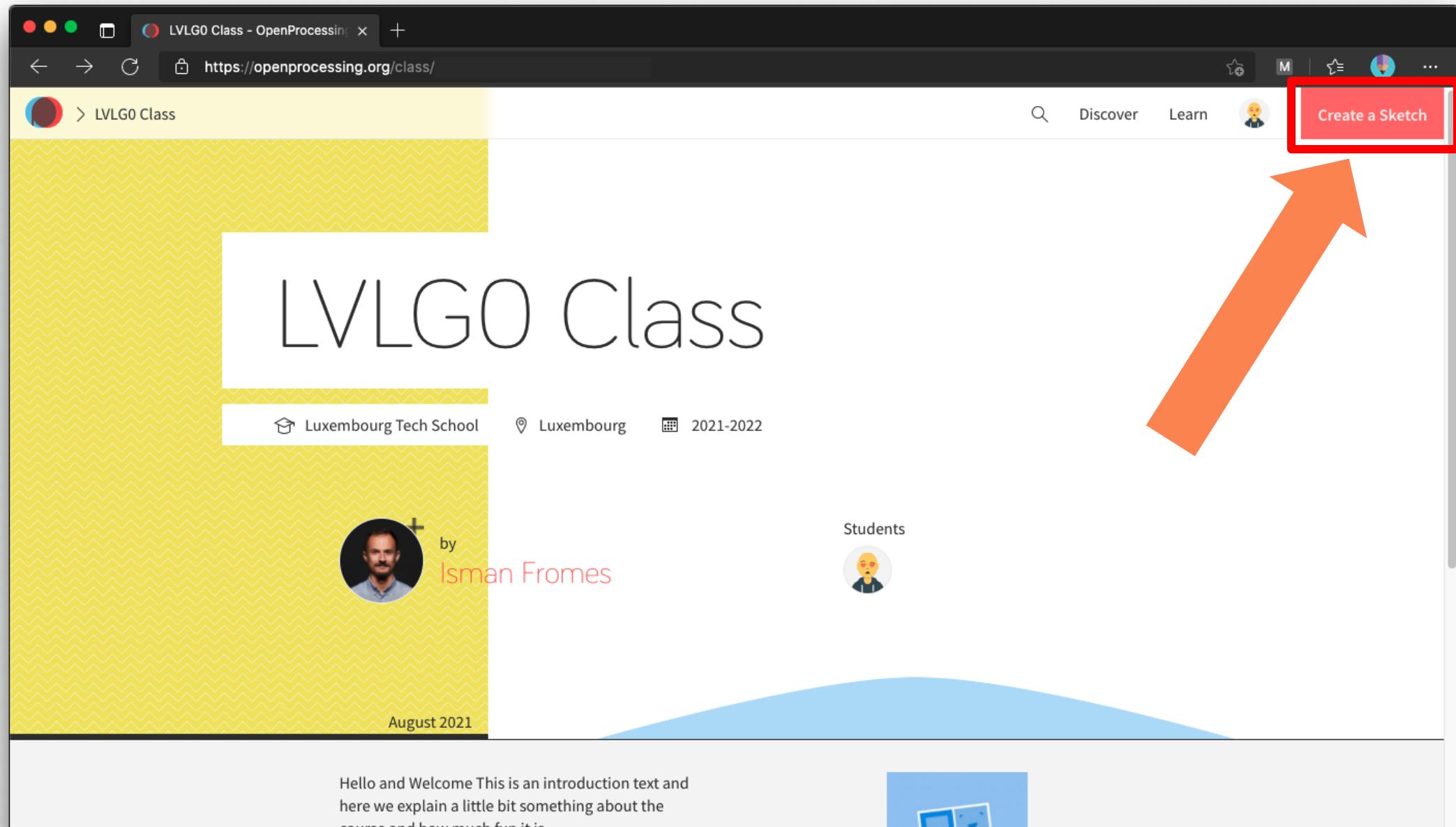
PASSWORD

STAY SIGNED IN

[forgot?](#)

SIGN IN

Click on “Create a Sketch”



The Editor View

The screenshot shows the p5.js Editor interface. The top bar includes a logo, a play button, a code preview icon, a user profile, and a red 'SAVE' button. The main area has tabs for 'SKETCH', 'FILES', and 'EDITOR'. The code editor contains the following P5.js code:

```
mySketch
function setup() {
  createCanvas(windowWidth, windowHeight);
  background(100);
}
function draw() {
  circle(mouseX, mouseY, 20);
}
```

The right sidebar contains several configuration options:

- MODE**: P5js (selected), HTML/CSS/JS, Pjs. A dropdown says "Select mode or a template".
- TUTORIAL MODE**: A toggle switch is off. Description: "Write step-by-step tutorials. [Learn more](#)".
- LIVE COLLABORATION**: A toggle switch is off. Description: "Please save your sketch to enable."
- SHOWCASE SKETCH**: A toggle switch is on. Description: "Centers sketch and matches the background color."
- LOOP PROTECTION**: A toggle switch is on. Description: "Prevents infinite loops that may freeze the sketch."
- LIBRARIES - SHOW ALL**: A section with a '+' button. Libraries listed: p5.sound (off), OP Configurator 3000 (off).

Change the Editor View

The screenshot shows the p5.js web editor interface. At the top, there's a toolbar with icons for play/pause, code editor, and save. Below the toolbar, the sketch title is "mySketch". The main area contains the following code:

```
function setup() {
  createCanvas(windowWidth, windowHeight);
  background(100);
}

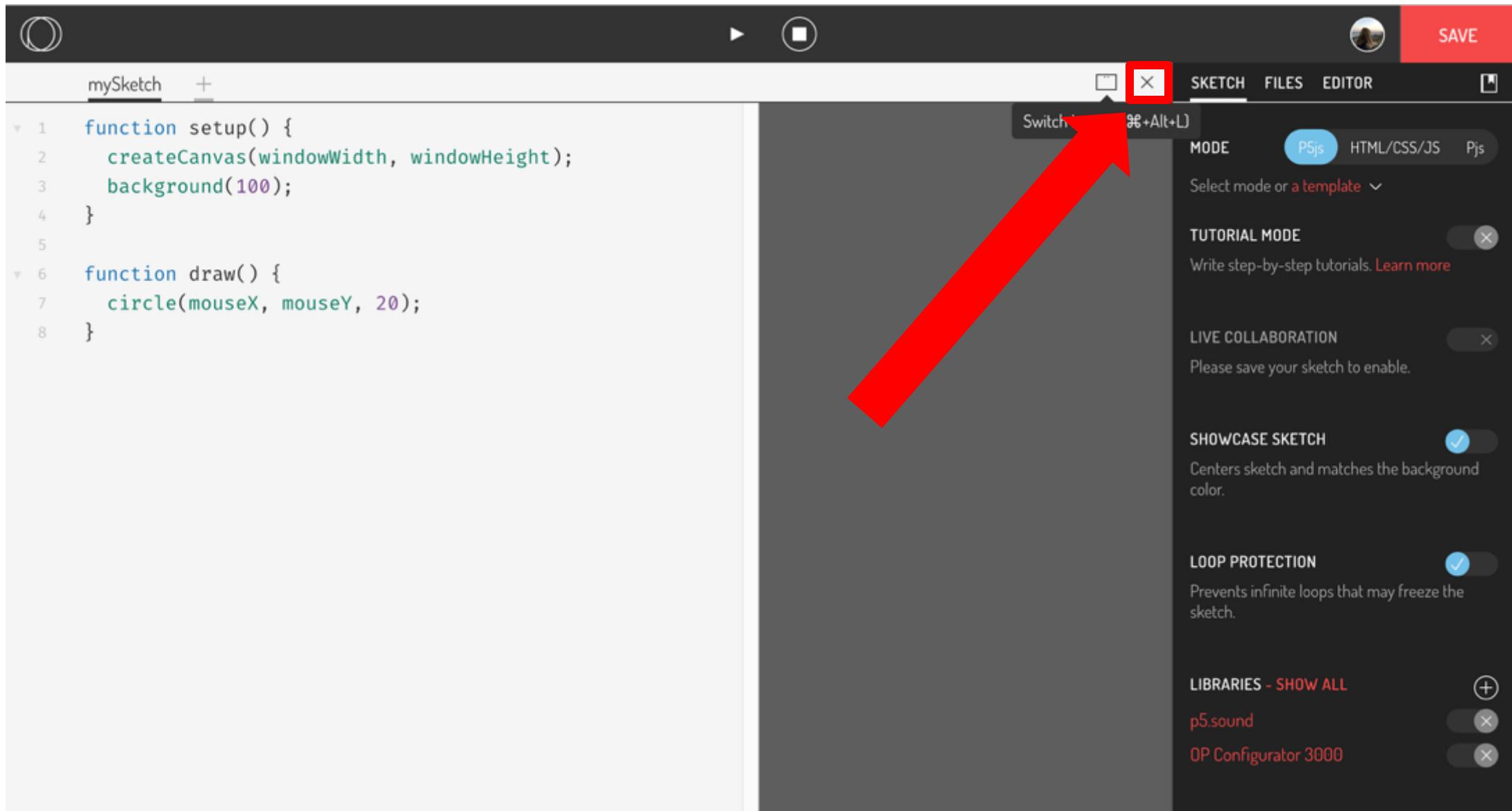
function draw() {
  circle(mouseX, mouseY, 20);
}
```

To the right of the code editor is a sidebar with various settings:

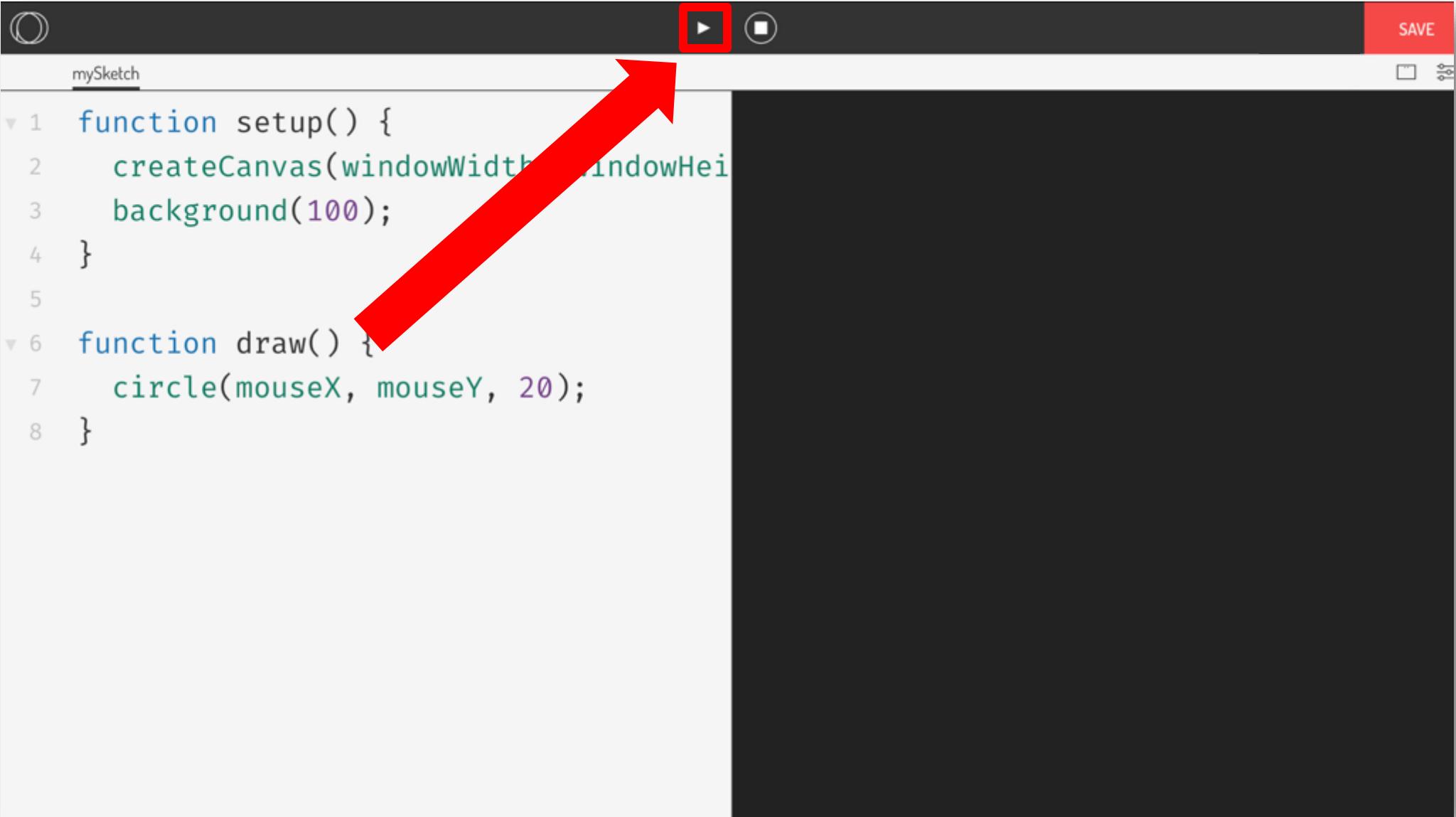
- MODE**: A radio button group with "P5js" selected, and options for "HTML/CSS/JS" and "Pjs".
Sub-options: "Select mode or a template" and "TUTORIAL MODE" (disabled).
- LIVE COLLABORATION**: A toggle switch that says "Please save your sketch to enable."
- SHOWCASE SKETCH**: A toggle switch that says "Centers sketch and matches the background color."
- LOOP PROTECTION**: A toggle switch that says "Prevents infinite loops that may freeze the sketch."
- LIBRARIES - SHOW ALL**: A section with a "+" button and two items: "p5.sound" and "OP Configurator 3000". Each item has a toggle switch to its right.

A large red arrow points from the bottom left towards the "SHOWCASE SKETCH" toggle switch, and a red box highlights the "SHOWCASE SKETCH" icon in the top bar.

Close the Menu



Press Play

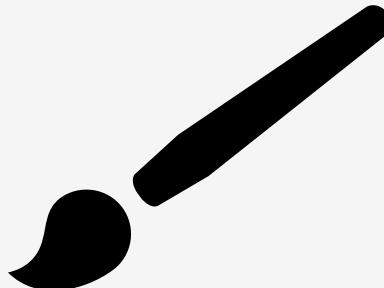


A screenshot of the Processing.js sketch editor interface. The top bar includes a file icon, a play button (highlighted with a large red arrow), a stop button, and a save button. The title bar says "mySketch". The code area contains the following P5.js code:

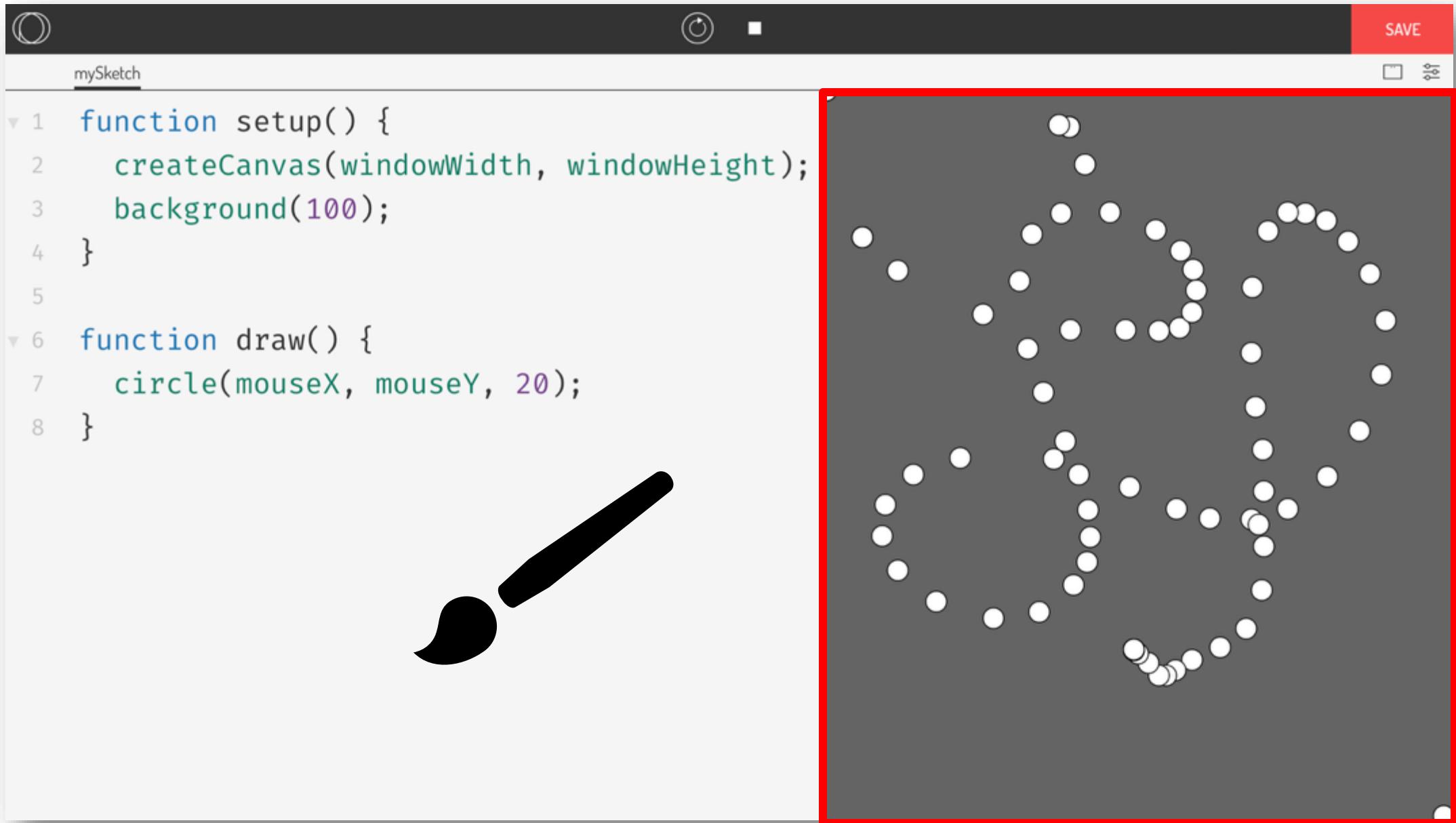
```
function setup() {
  createCanvas(windowWidth, windowHeight);
  background(100);
}

function draw() {
  circle(mouseX, mouseY, 20);
}
```

Draw with your Mouse

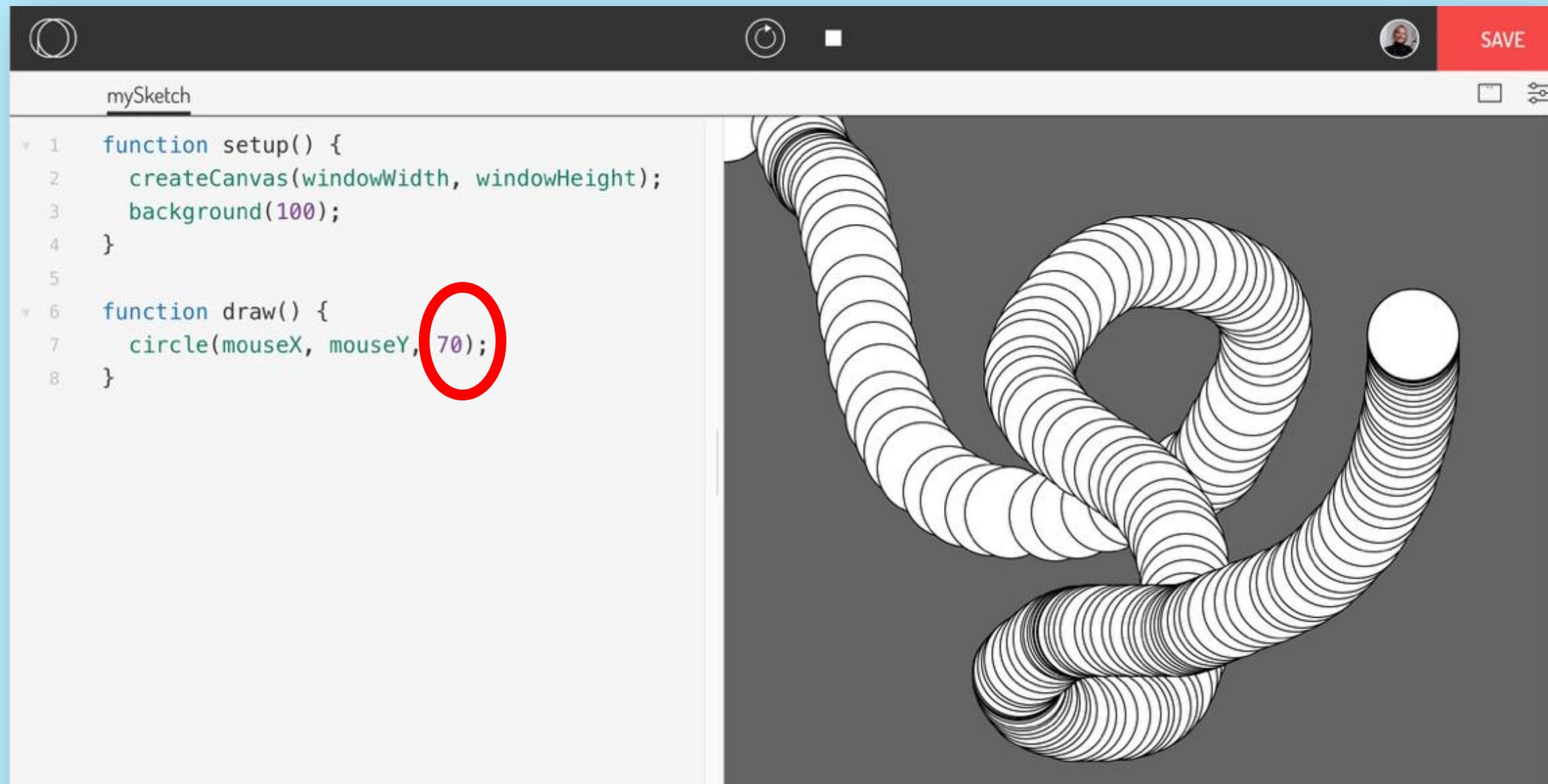


```
mySketch
1 function setup() {
2   createCanvas(windowWidth, windowHeight);
3   background(100);
4 }
5
6 function draw() {
7   circle(mouseX, mouseY, 20);
8 }
```



The image shows a Processing.js sketch titled "mySketch". The code consists of two functions: "setup" and "draw". The "setup" function creates a canvas with the same dimensions as the window and sets the background color to a light gray shade (100). The "draw" function contains a single line of code: "circle(mouseX, mouseY, 20);", which draws a white circle at the current mouse position with a diameter of 20 pixels. The sketch window has a dark gray header bar with standard window controls (minimize, maximize, close) and a "SAVE" button. The main canvas area is dark gray and contains several white circles scattered across it, forming a loose, abstract shape. A red rectangular box highlights this drawing area. The entire sketch window is framed by a thin red border.

Change the brush size

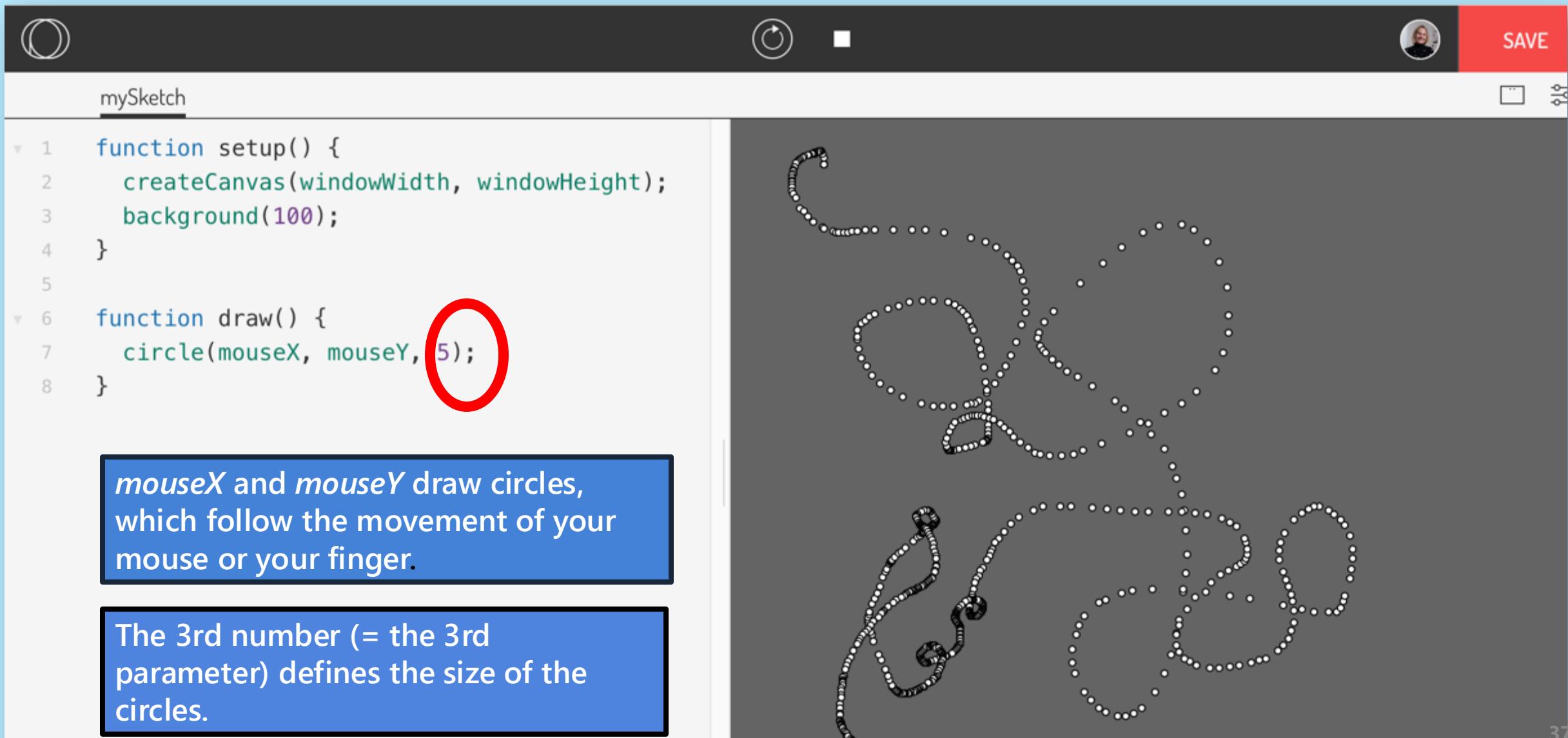


```
mySketch

function setup() {
  createCanvas(windowWidth, windowHeight);
  background(100);
}

function draw() {
  circle(mouseX, mouseY, 70);
}
```

Change the brush size



The screenshot shows a Processing.js sketch window titled "mySketch". The code in the editor is:

```
function setup() {
  createCanvas(windowWidth, windowHeight);
  background(100);
}

function draw() {
  circle(mouseX, mouseY, 5);
}
```

A red circle highlights the number **5** in the `circle(mouseX, mouseY, 5);` line, indicating it is the parameter for circle size.

mouseX and mouseY draw circles, which follow the movement of your mouse or your finger.

The 3rd number (= the 3rd parameter) defines the size of the circles.

The output canvas displays a series of small circles (size 5) connected by lines, forming a winding path that follows the movement of the mouse.

Stop button



The screenshot shows the Processing.js environment. At the top, there's a toolbar with various icons: a play button, a stop button (which is circled in red), a save button, and other controls. Below the toolbar, the sketch title is "mySketch". The code area contains the following P5.js code:

```
1 function setup() {
2     createCanvas(windowWidth, windowHeight);
3     background(100);
4 }
5
6 function draw() {
7     circle(mouseX, mouseY, 5);
8 }
```

The canvas area is currently blank.

Change background color

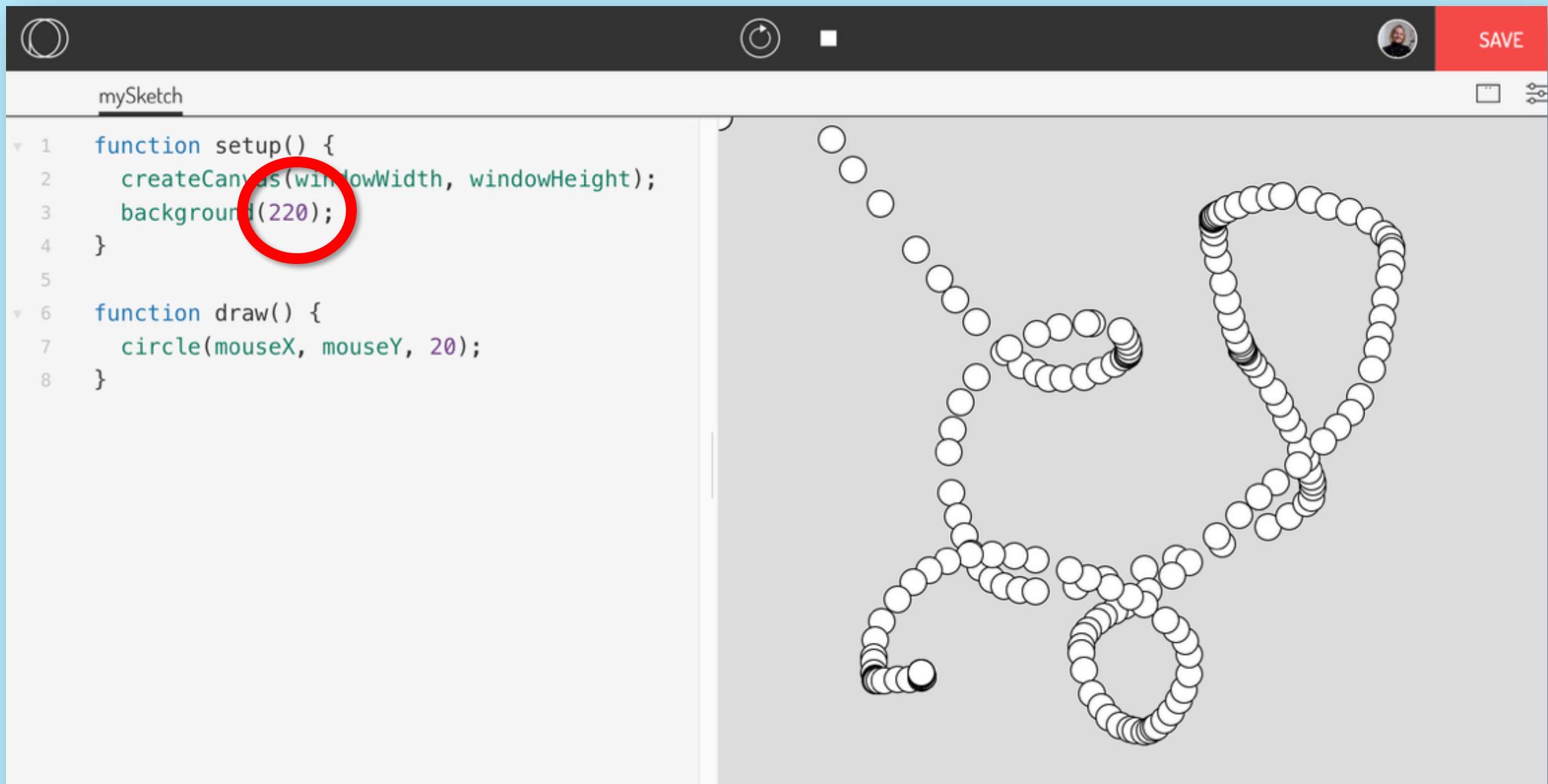


mySketch

```
function setup() {
  createCanvas(windowWidth, windowHeight);
  background(0);
}

function draw() {
  circle(mouseX, mouseY, 20);
}
```

Change background color



The image shows a screenshot of a Processing sketch titled "mySketch". The code in the editor is:

```
function setup() {
  createCanvas(windowWidth, windowHeight);
  background(220);
}

function draw() {
  circle(mouseX, mouseY, 20);
}
```

A red circle highlights the `background(220);` line in the setup function. The sketch window displays a spiral of white circles centered at the mouse position, with a light gray background.

The console

The screenshot shows the Processing.js code editor interface. The title bar says "mySketch". The code area contains the following:

```
1 function setup() {
2     createCanvas(windowWidth, windowHeight);
3     background(220); // Line 3
4 }
5
6 function draw() {
7     circle(mouseX, mouseY, 20);
8 }
```

A red circle highlights the closing parenthesis at line 3, "background(220);". Below the code editor, a red box highlights the error message in the status bar:

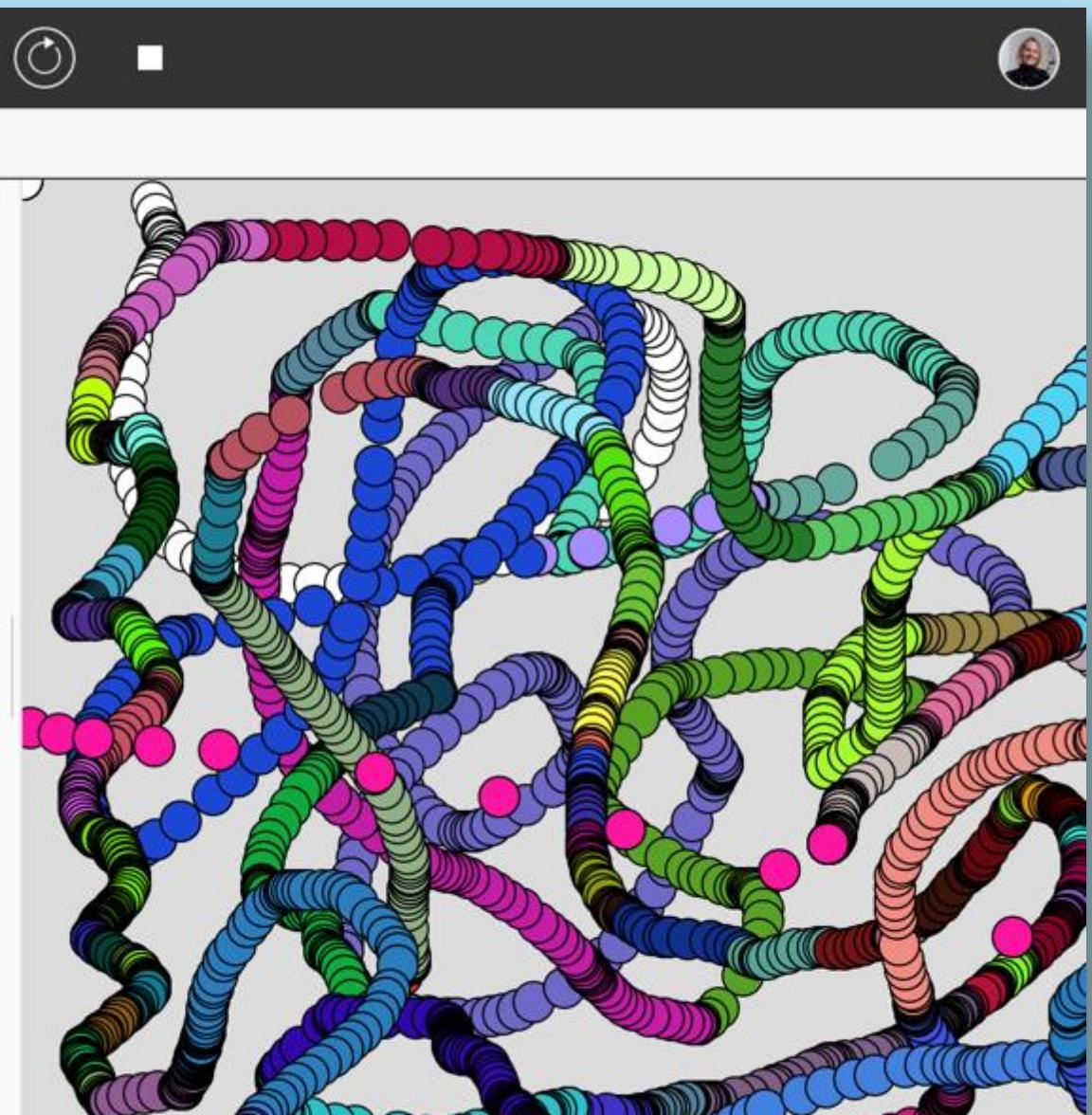
mySketch, line 3: SyntaxError: missing) after argument list

[Get help from AI?](#)

The console will only appear if there is an error in your code or if you used the print() function.

The status bar also includes a "X" icon and the number "41".

Add random colours



The screenshot shows a Processing sketch window titled "mySketch". The code is as follows:

```
function setup() {
  createCanvas(windowWidth, windowHeight);
  background(220);
}

function draw() {
  circle(mouseX, mouseY, 20);
}

function mousePressed() {
  fill(random(255), random(255), random(255));
}
```

A red box highlights the `fill()` statement in the `mousePressed()` function. Below the code, two explanatory text boxes are shown:

`fill()` = colour for the next shapes

`random(255)` = random number 0 - 255
for RGB colour

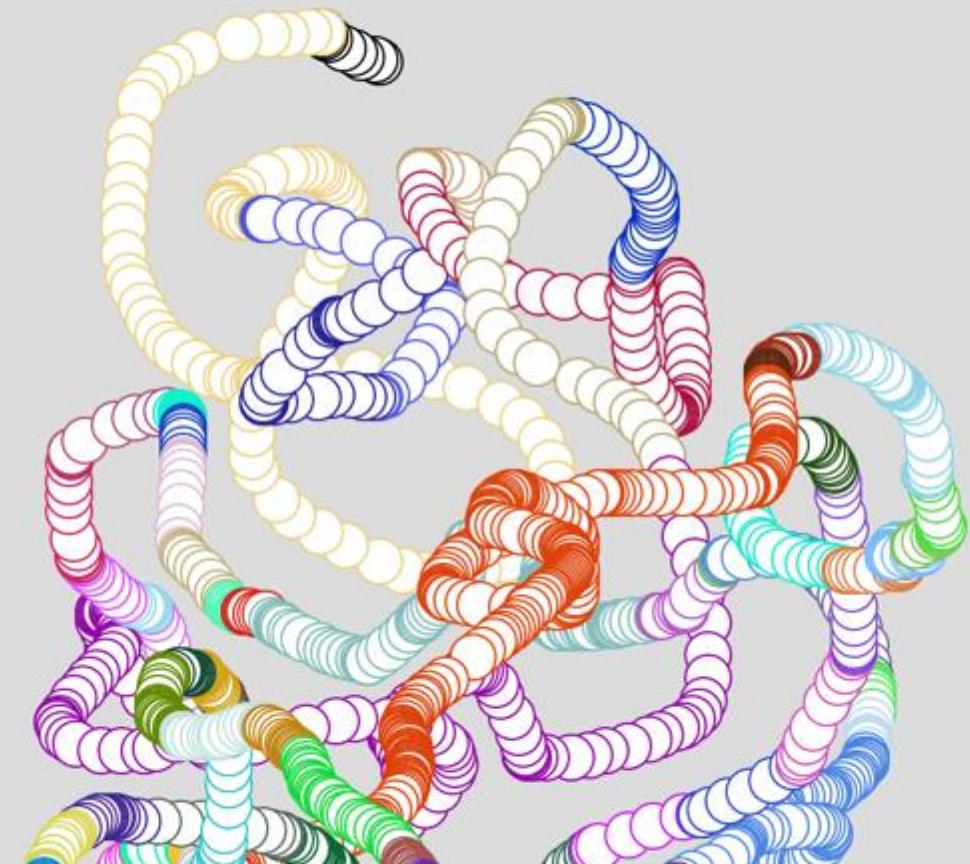
Make a rainbow mess on your screen



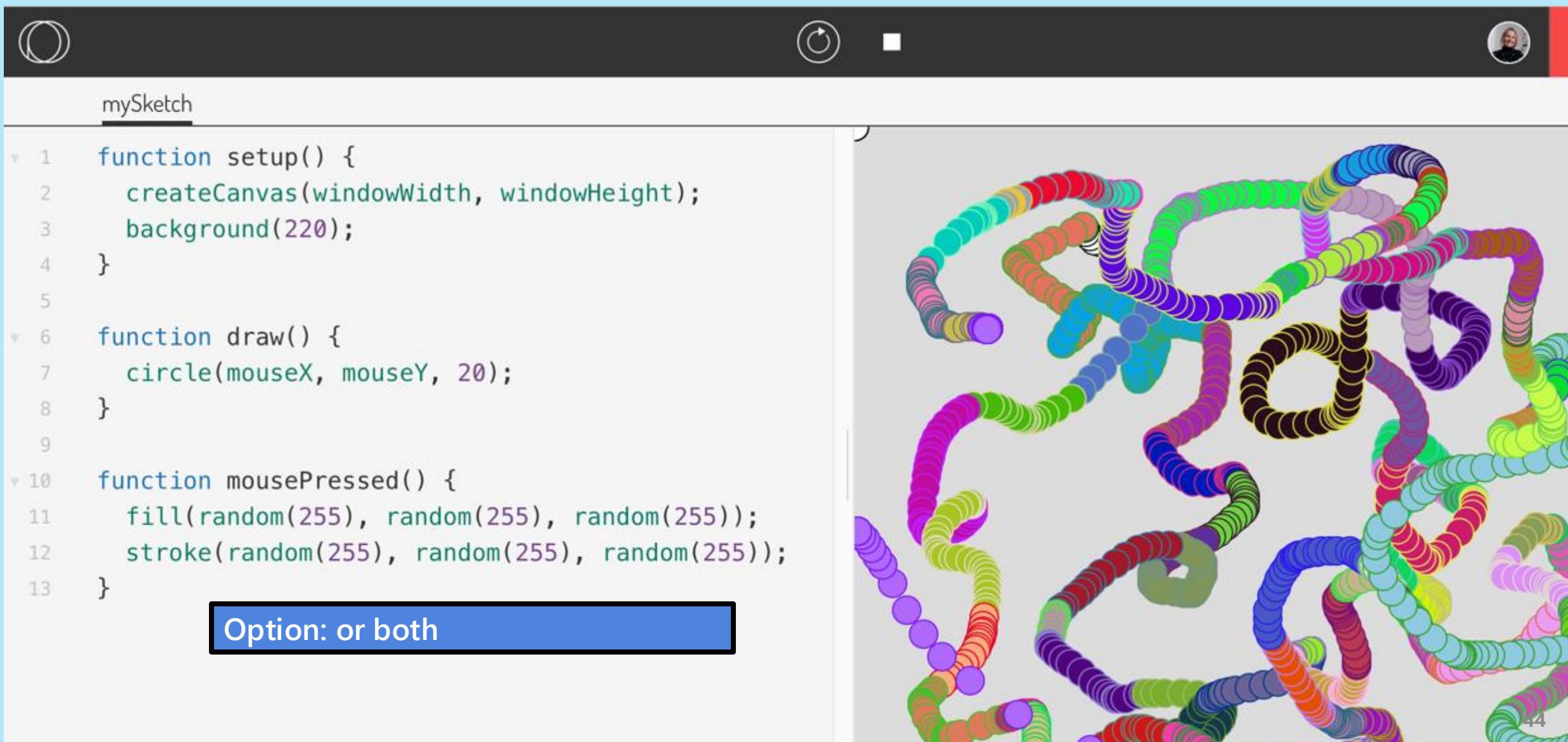
mySketch

```
1  function setup() {  
2      createCanvas(windowWidth, windowHeight);  
3      background(220);  
4  }  
5  
6  function draw() {  
7      circle(mouseX, mouseY, 20);  
8  }  
9  
10 function mousePressed() {  
11     stroke(random(255), random(255), random(255));  
12 }
```

Option: show *stroke* instead of *fill*



Make a rainbow mess on your screen



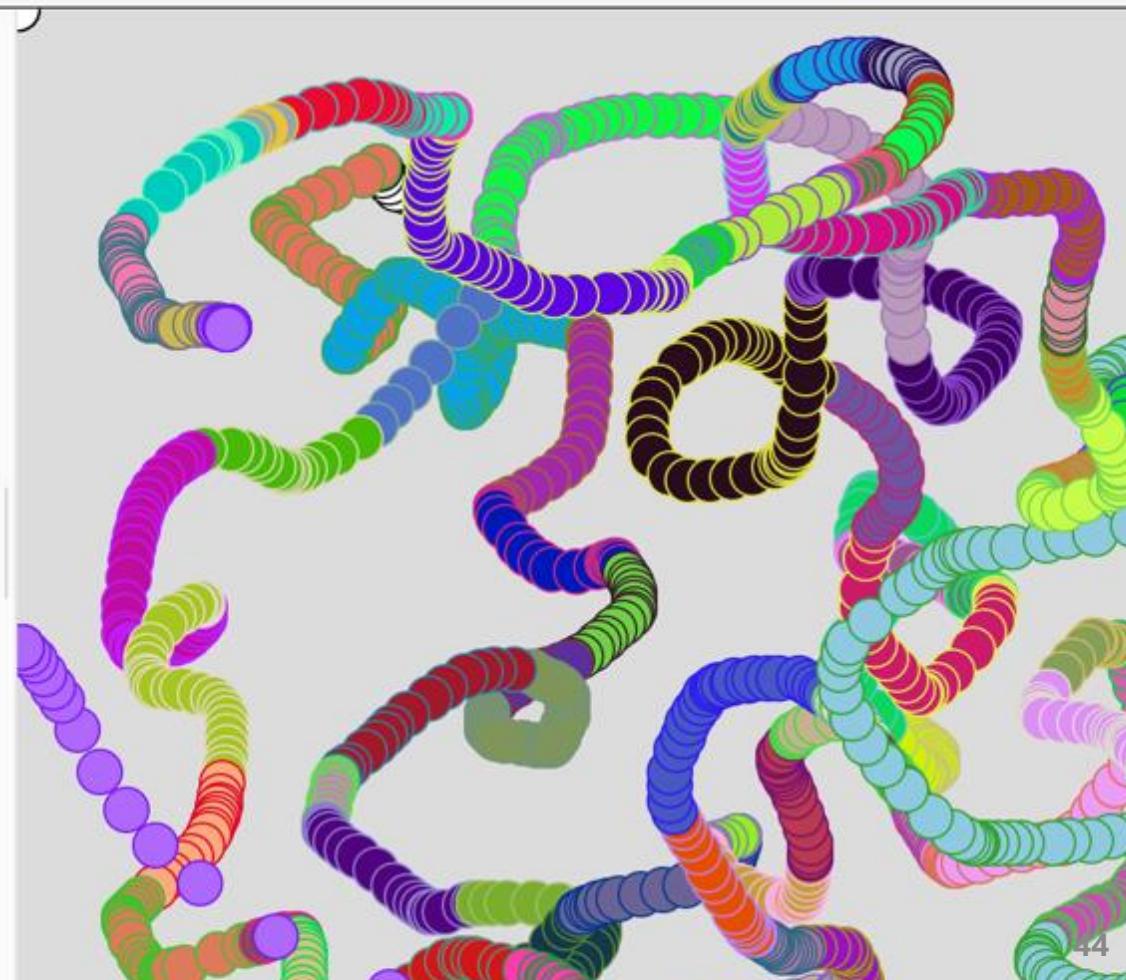
mySketch

```
function setup() {
  createCanvas(windowWidth, windowHeight);
  background(220);
}

function draw() {
  circle(mouseX, mouseY, 20);
}

function mousePressed() {
  fill(random(255), random(255), random(255));
  stroke(random(255), random(255), random(255));
}
```

Option: or both



Extra: change background color



The image shows a Processing sketch window titled "mySketch". The code is as follows:

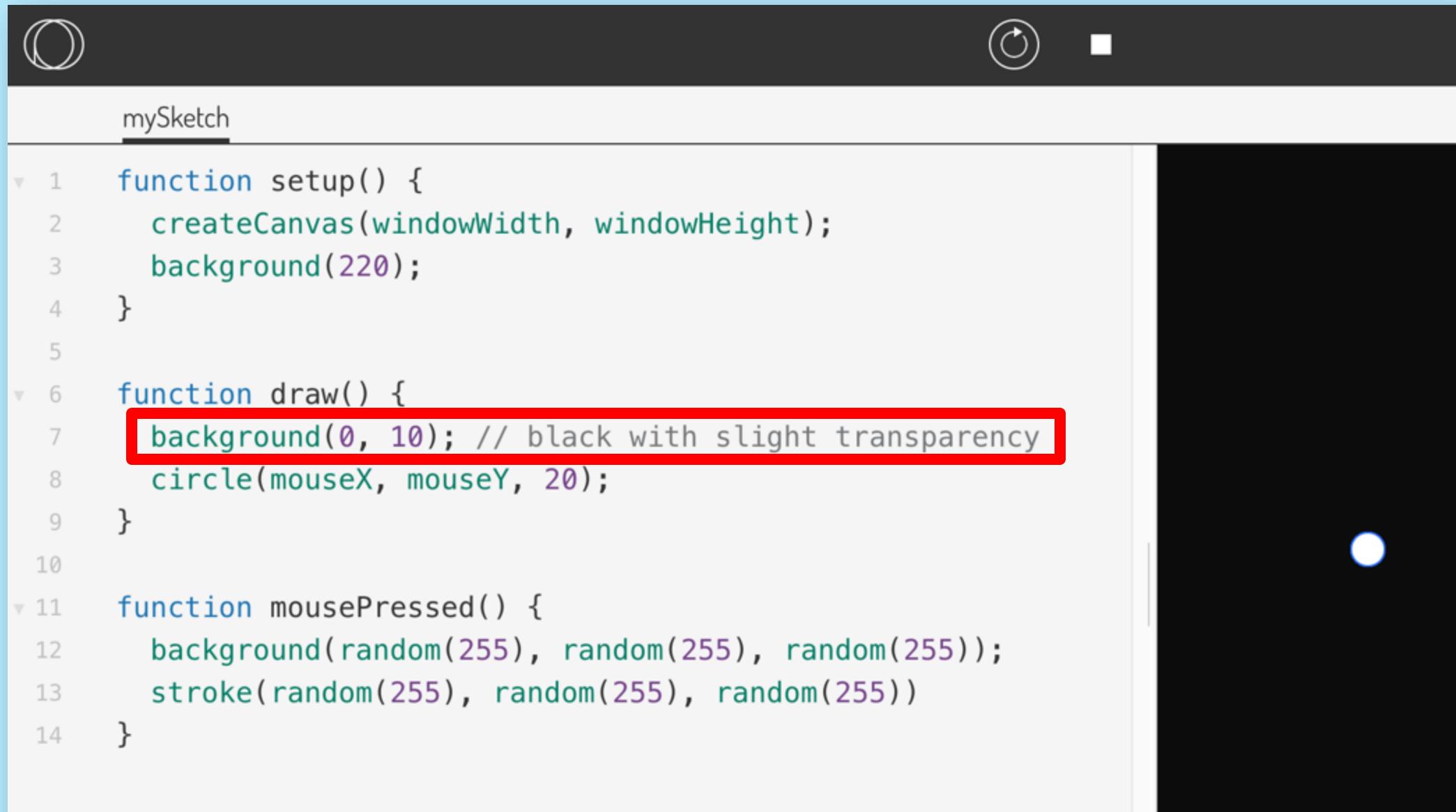
```
function setup() {
    createCanvas(windowWidth, windowHeight);
    background(220);
}

function draw() {
    circle(mouseX, mouseY, 20);
}

function mousePressed() {
    background(random(255), random(255), random(255));
    stroke(random(255), random(255), random(255));
}
```

The background of the sketch is blue. A yellow spiral line is drawn on it, starting from the bottom left and curving upwards and to the right. The spiral is composed of many small circles.

Extra: Transparent trails

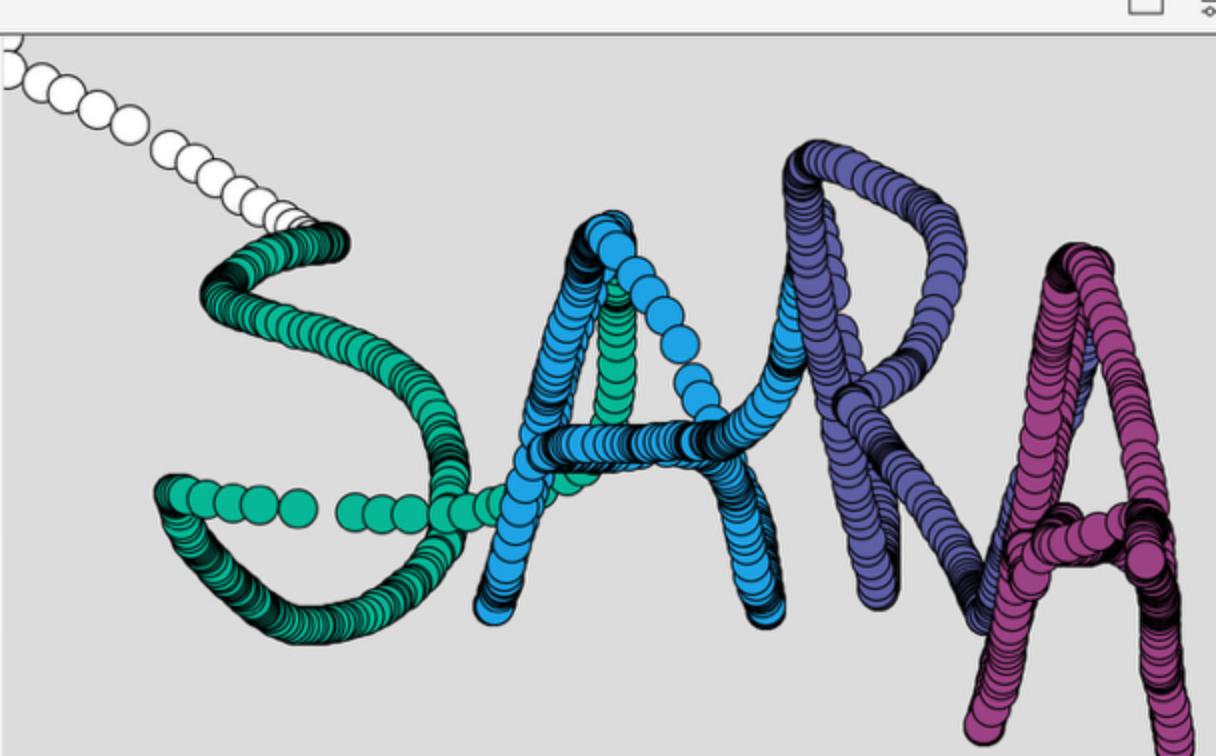


The image shows a screenshot of a Processing sketch window titled "mySketch". The code is as follows:

```
1 function setup() {
2     createCanvas(windowWidth, windowHeight);
3     background(220);
4 }
5
6 function draw() {
7     background(0, 10); // black with slight transparency
8     circle(mouseX, mouseY, 20);
9 }
10
11 function mousePressed() {
12     background(random(255), random(255), random(255));
13     stroke(random(255), random(255), random(255))
14 }
```

The line `background(0, 10);` is highlighted with a red rectangle. A single blue circle is visible on the right side of the canvas.

Extra: write your name & screenshot



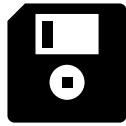
The image shows a Processing sketch titled "mySketch". The code defines three functions: setup, draw, and mousePressed. The setup function creates a canvas and sets the background to a light gray. The draw function contains a single line: "circle(mouseX, mouseY, 20);". The mousePressed function fills the circle with a random color. The resulting visualization is a large, stylized word "SARA" where each letter is composed of a series of overlapping circles in different colors: teal, blue, purple, and pink.

```
function setup() {
  createCanvas(windowWidth, windowHeight);
  background(220);
}

function draw() {
  circle(mouseX, mouseY, 20);
}

function mousePressed() {
  fill(random(255), random(255), random(255))
}
```

Save



```
mySketch
function setup() {
  createCanvas(600, 600);
  background(100);
}

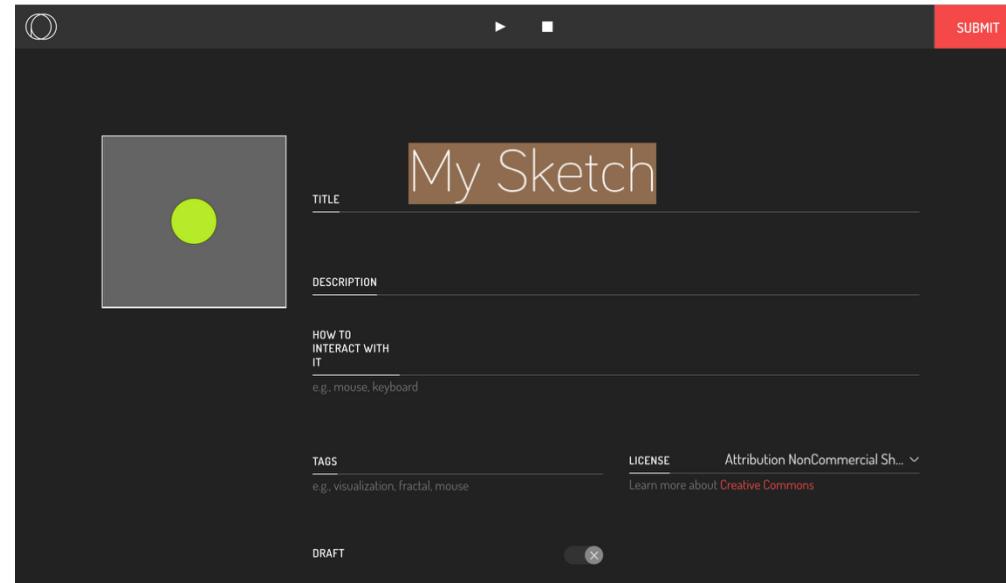
function draw() {
  fill(183, 235, 40);
```

1

Click on **SAVE**

2

Give the sketch
a new title
instead of „My
Sketch“ e.g.
„Painting“



3

Click on **SUBMIT**

How to take a screenshot

What is a screenshot?

= A photo of your computer screen, taken with the computer.

- You can take a photo of the whole screen or only a part of it.
- The screenshot is saved on your computer and you can upload or share it.

Screenshot on Windows

Press all keys at the same time:



Screenshot on Windows

... or you might have these keys on your keyboard



Screenshot on Windows

Video: <https://www.youtube.com/watch?v=MqFKuImFGYQ>

Screenshot on mac

Press all keys at the same time:



3 takes a photo of the whole screen
4 lets you select a part of the screen

Screenshot on mac

Video: https://www.youtube.com/watch?v=X5Oq2_YXV6s



ONLINE CLASSES

How to code during Live Class (1/2)

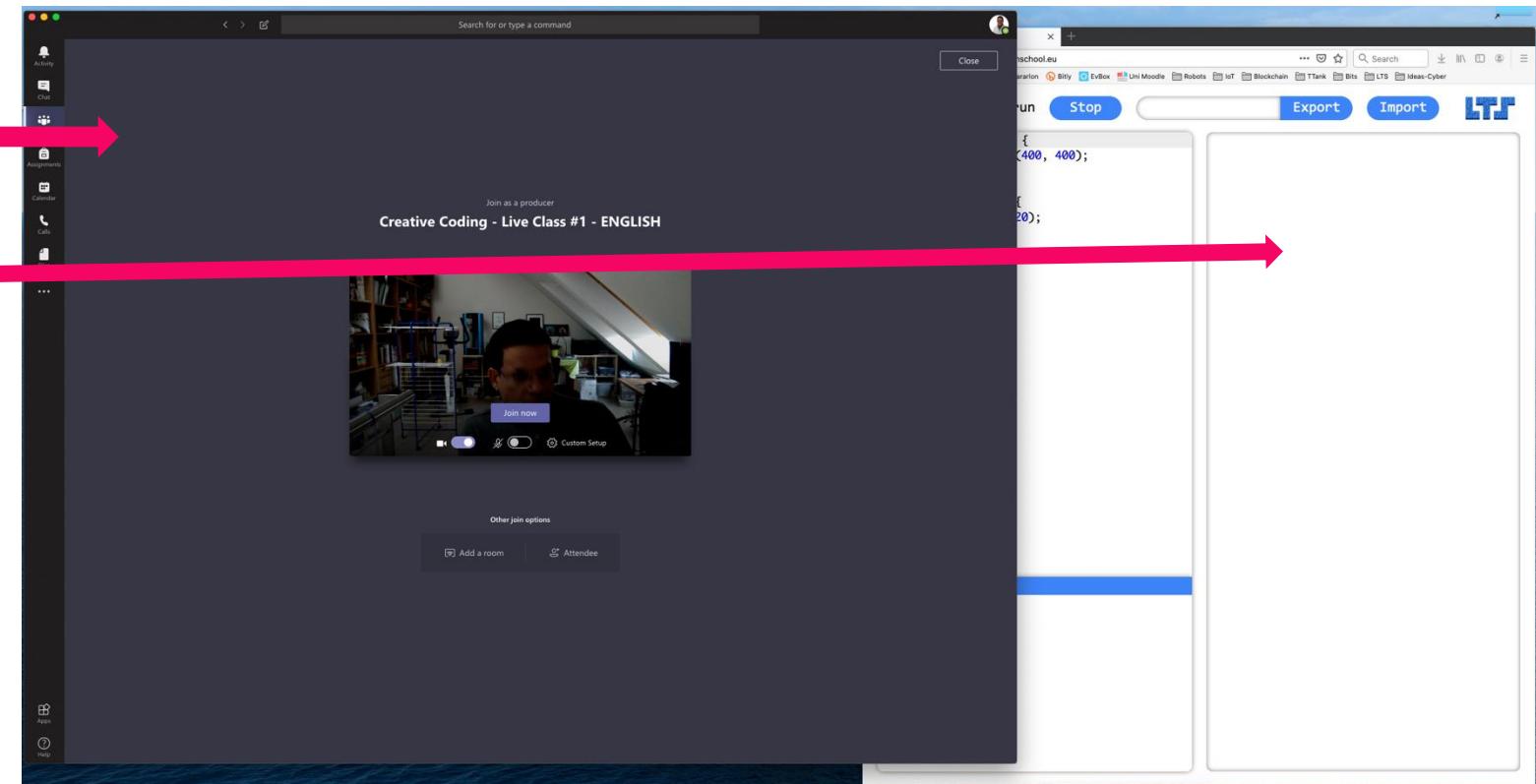
Option 1

1

Open two windows
(one with the Live Class in Teams, one with the Web Editor; close all other apps).

2

Windows are not maximized and overlap in the middle, so you can **click from one to the other**.



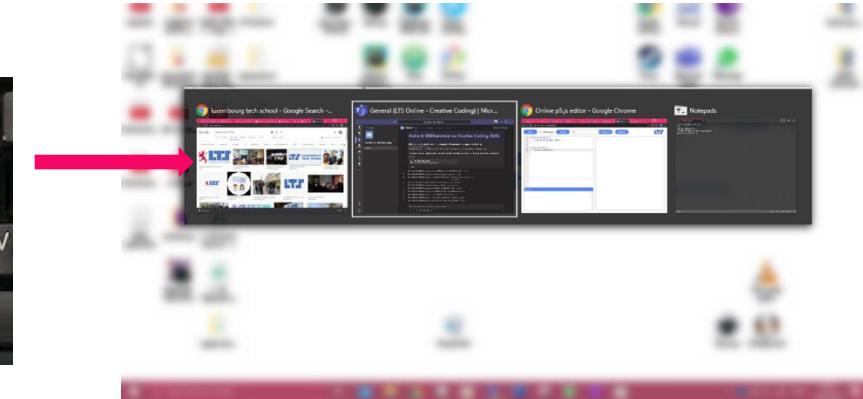
How to code during Live Class (2/2)

Option 2

1 press **Alt + Tab** (Windows)

or

press **Cmd + Tab** (Mac)



2

Now repeatedly **click on Tab**. When the window you want to go to is highlighted, you can **release the keys** and that window will open.

