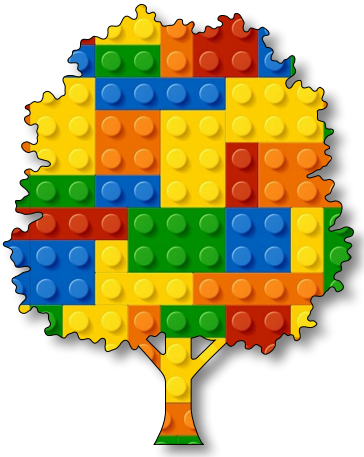
Algorithmic Art

Module F

Unit #9

using json files



Content

Module F Unit #9 using json files

Introduction to using Jason files

This is primarily about data files in the .json format. This covers how they are created and can be used. JavaScript makes good use of this data file type, as do other coding languages.

Data is formatted in many different forms; two very common ones are CSV or JSON, which are easily accessed by computers. From those file types, you can access the data in any form or order you like.

Sketch F9.1 a bubble object

This is just a simple sketch with one bubble described as an object and drawing it filled with yellow.

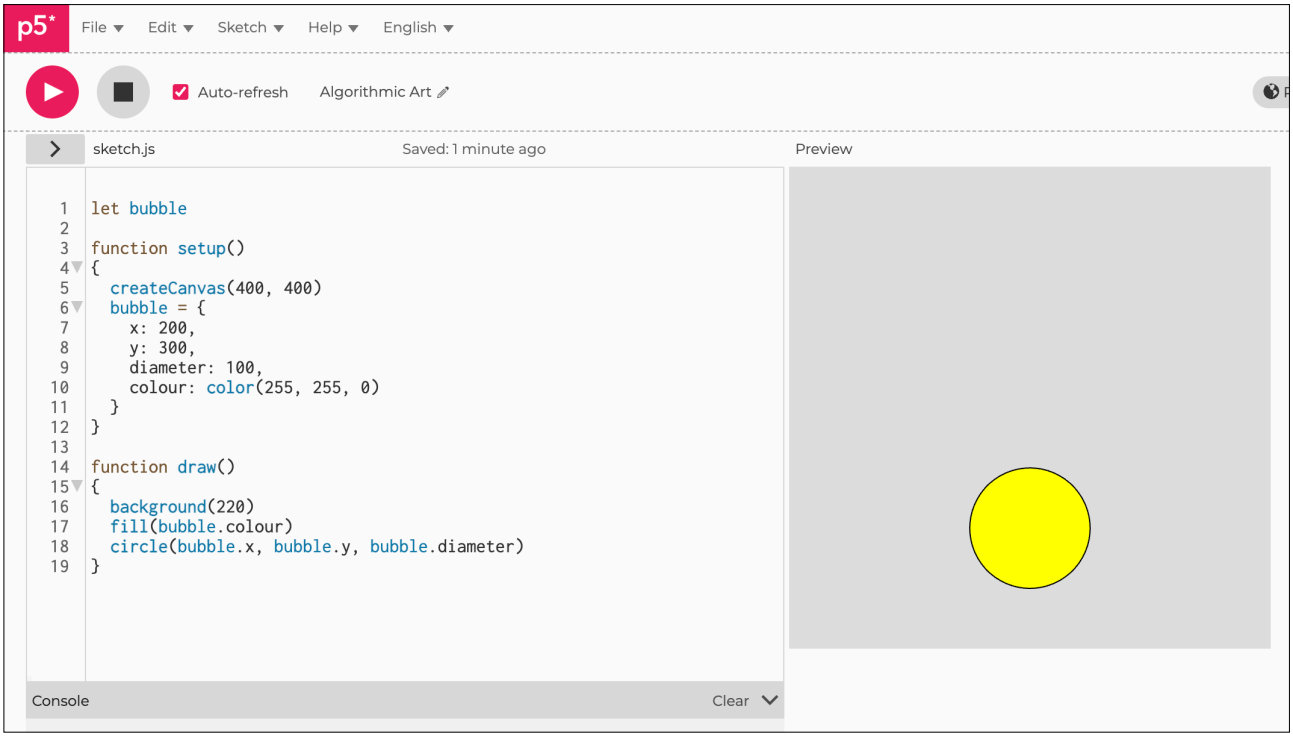
sketch.js

```
let bubble

function setup()
{
  createCanvas(400, 400)
  bubble = {
    x: 200,
    y: 300,
    diameter: 100,
    colour: color(255, 255, 0)
  }
}

function draw()
{
  background(220)
  fill(bubble.colour)
  circle(bubble.x, bubble.y, bubble.diameter)
}
```

🗒️ Notes

Creating a simple dataset of one yellow bubble with `x`, `y` co-ordinates and a `diameter`. The bubble is an object with those elements. This works fine for one bubble, but what if you want `100` different circles? It can be improved by putting the data in a separate .json file, which is a JavaScript data file.

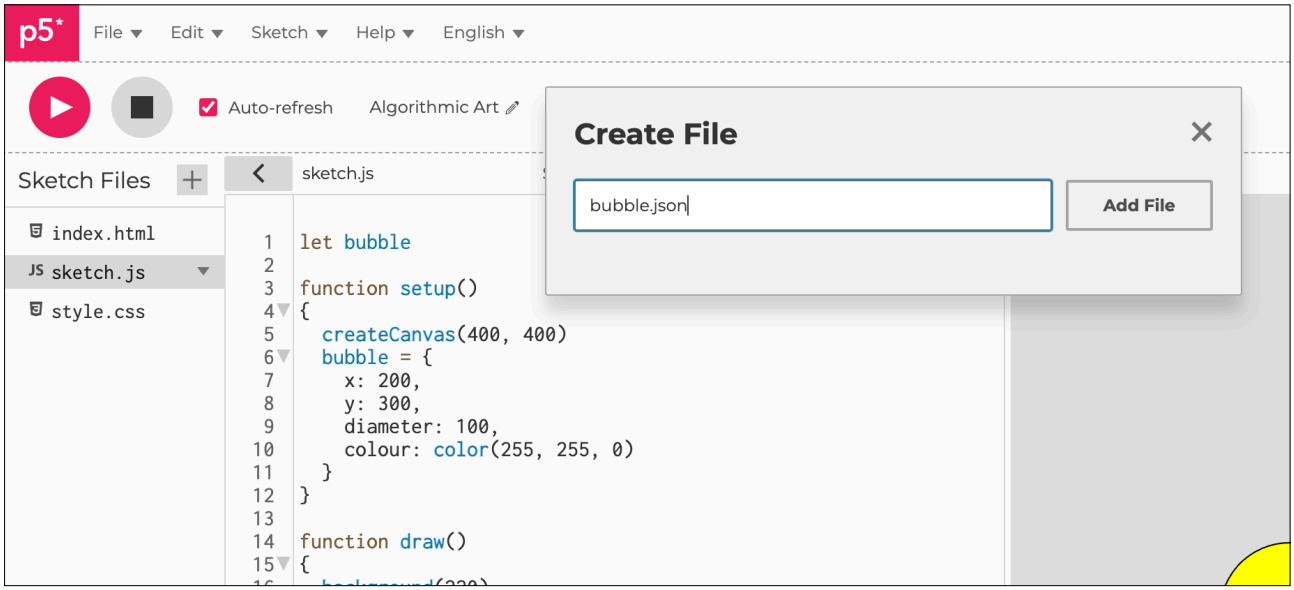
Figure F9.1


p5*
File
Edit
Sketch
Help
English
Auto-refresh
Algorithmic Art
sketch.js
Saved: 1 minute ago
Preview
let bubble
function setup()
{
createCanvas(400, 400)
bubble = {
x: 200,
y: 300,
diameter: 100,
colour: color(255, 255, 0)
}
}
function draw()
{
background(220)
fill(bubble.colour)
circle(bubble.x, bubble.y, bubble.diameter)
}
Console
Clear

Creating a json file

Create a new file and call it `bubble.json`, add it to the list of files and add it there as we have done before.

Figure 1: creating the bubble.json file



Sketch F9.2 the json data file

❗ The `.json` file.

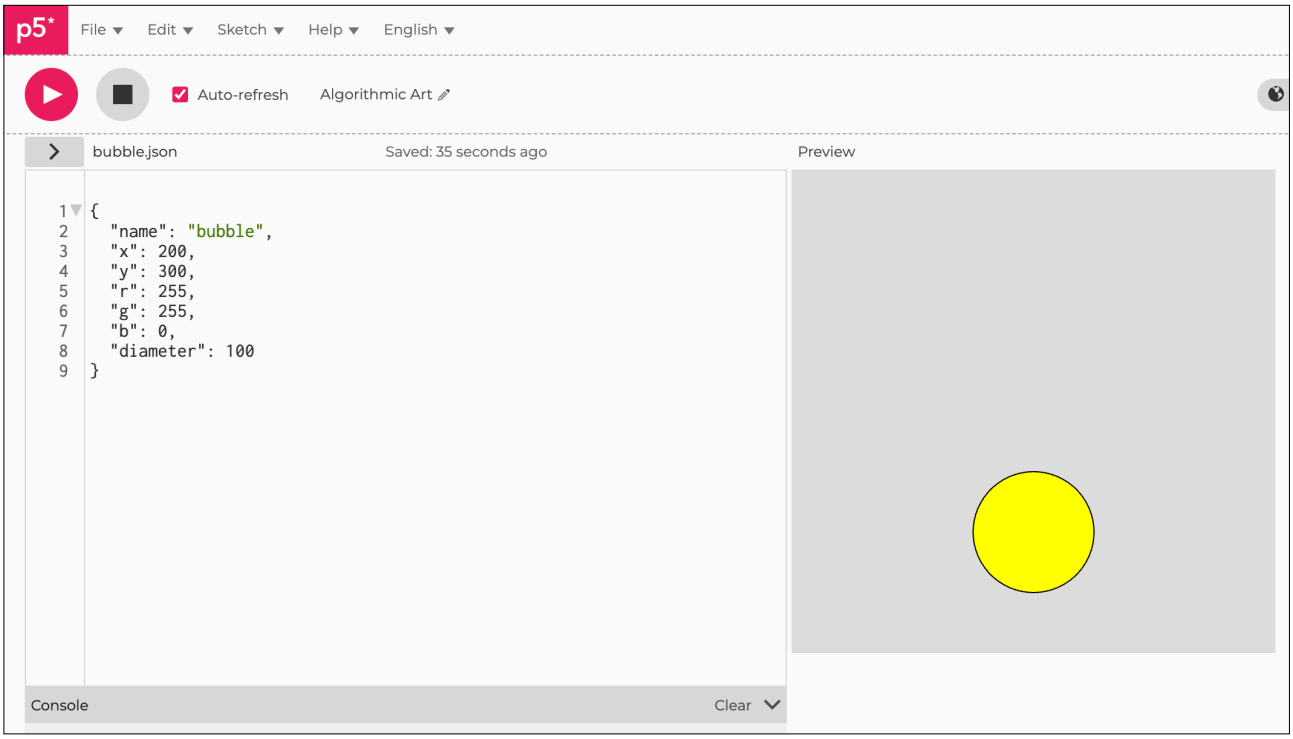
bubble.json

```
{
  "name": "bubble",
  "x": 200,
  "y": 300,
  "r": 255,
  "g": 255,
  "b": 0,
  "diameter": 100
}
```

🗒️ Notes

You can check if the format is correct because it does need to be. The following website will do that by you copying and pasting it into the website, and it will return the result:
https://jsonformatter.curiousconcept.com/#
is a way of checking if the JSON is OK.

Figure F9.2


p5*
File
Edit
Sketch
Help
English
Auto-refresh
Algorithmic Art
bubble.json
Saved: 35 seconds ago
Preview
{
"name": "bubble",
"x": 200,
"y": 300,
"r": 255,
"g": 255,
"b": 0,
"diameter": 100
}
Console
Clear

Sketch F9.3 adapting the sketch

❗ Back to sketch.js.

Now we adapt the sketch accordingly. Notice we don't need the bubble data in the sketch; we just need to upload it.

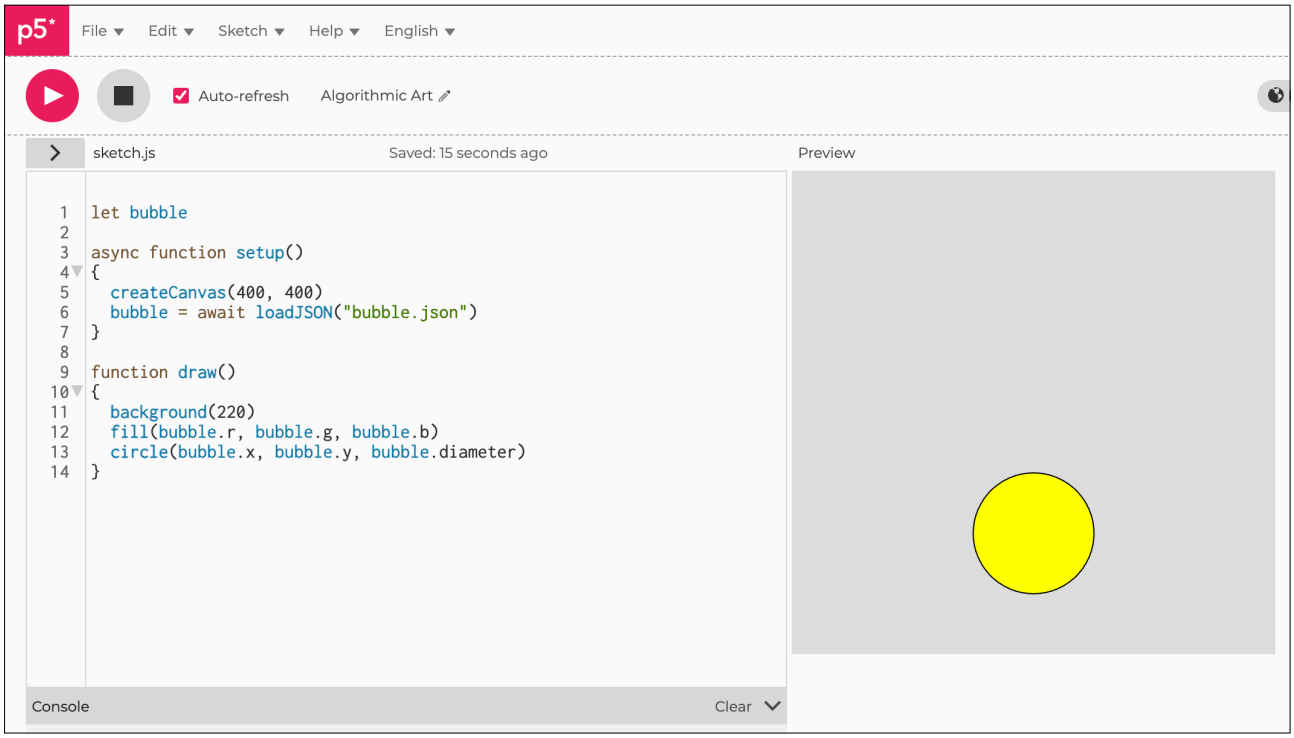
sketch.js

```
let bubble

async function setup()
{
  createCanvas(400, 400)
  bubble = await loadJSON("bubble.json")
}

function draw()
{
  background(220)
  fill(bubble.r, bubble.g, bubble.b)
  circle(bubble.x, bubble.y, bubble.diameter)
}
```

🗒️ Notes

We load the data and extract it accordingly.

Figure F9.3


p5*
File
Edit
Sketch
Help
English
Auto-refresh
Algorithmic Art
sketch.js
Saved: 15 seconds ago
Preview
let bubble
async function setup()
{
createCanvas(400, 400)
bubble = await loadJSON("bubble.json")
}
function draw()
{
background(220)
fill(bubble.r, bubble.g, bubble.b)
circle(bubble.x, bubble.y, bubble.diameter)
}
Console
Clear

Sketch F9.4 more than one bubble

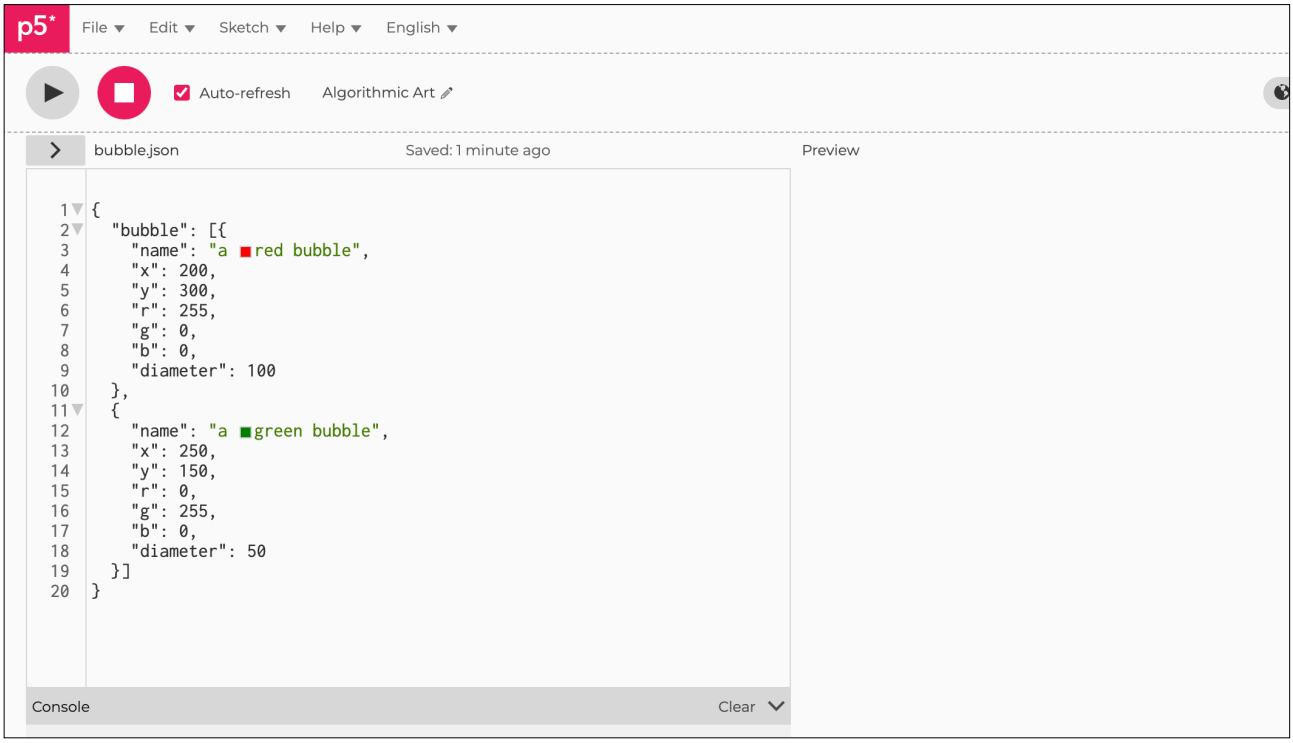
❗ Revisiting the bubble.json file.
This is the `bubble.json` file with two bubbles, a red bubble and a green bubble. The bubble is now an array called `bubble[]`. The `bubble[0]` is the `red` bubble and `bubble[1]` is the `green` bubble.

bubble.json

```
{
  "bubble": [{
    "name": "a red bubble",
    "x": 200,
    "y": 300,
    "r": 255,
    "g": 0,
    "b": 0,
    "diameter": 100
  },
  {
    "name": "a green bubble",
    "x": 250,
    "y": 150,
    "r": 0,
    "g": 255,
    "b": 0,
    "diameter": 50
  }]
}
```

Notes
This array has two bubble elements.

Figure F9.4


p5*
File
Edit
Sketch
Help
English
Auto-refresh
Algorithmic Art
bubble.json
Saved: 1 minute ago
Preview
{
"bubble": [{
"name": "a red bubble",
"x": 200,
"y": 300,
"r": 255,
"g": 0,
"b": 0,
"diameter": 100
},
{
"name": "a green bubble",
"x": 250,
"y": 150,
"r": 0,
"g": 255,
"b": 0,
"diameter": 50
}]
}
Console
Clear

Sketch F9.5 a long winded way

❗ In the sketch.js.

We can now draw these two bubbles. We create a variable called data, and the JSON file is loaded into that data variable.

sketch.js

```
let data

async function setup()
{
  createCanvas(400, 400)
  data = await loadJSON("bubble.json")
}

function draw()
{
  background(220)
  fill(data.bubble[0].r, data.bubble[0].g, data.bubble[0].b)
  circle(data.bubble[0].x, data.bubble[0].y, data.bubble[0].diameter)
  fill(data.bubble[1].r, data.bubble[1].g, data.bubble[1].b)
  circle(data.bubble[1].x, data.bubble[1].y, data.bubble[1].diameter)
}
```

🗒️ Notes

I am sure you can think of a way to do this more efficiently with a for loop.

🌻 Challenges

1. Add more bubbles
2. Add different shapes
3. There is some useful JSON data on GitHub which you can copy and create your own JSON file and try to pull data from it. The link is: https://github.com/dariusk/corpora/tree/master/data.
4. Have a go at using this dataset and see what you can do.

Sketch F9.6 a more concise way

That is so much better, using a simple `for()` loop.

sketch.js

```
let data

async function setup()
{
  createCanvas(400, 400)
  data = await loadJSON("bubble.json")
}

function draw()
{
  background(220)
  for (let i = 0; i < data.bubble.length; i++)
  {
    fill(data.bubble[i].r, data.bubble[i].g, data.bubble[i].b)
    circle(data.bubble[i].x, data.bubble[i].y, data.bubble[i].diameter)
  }
}
```

Notes

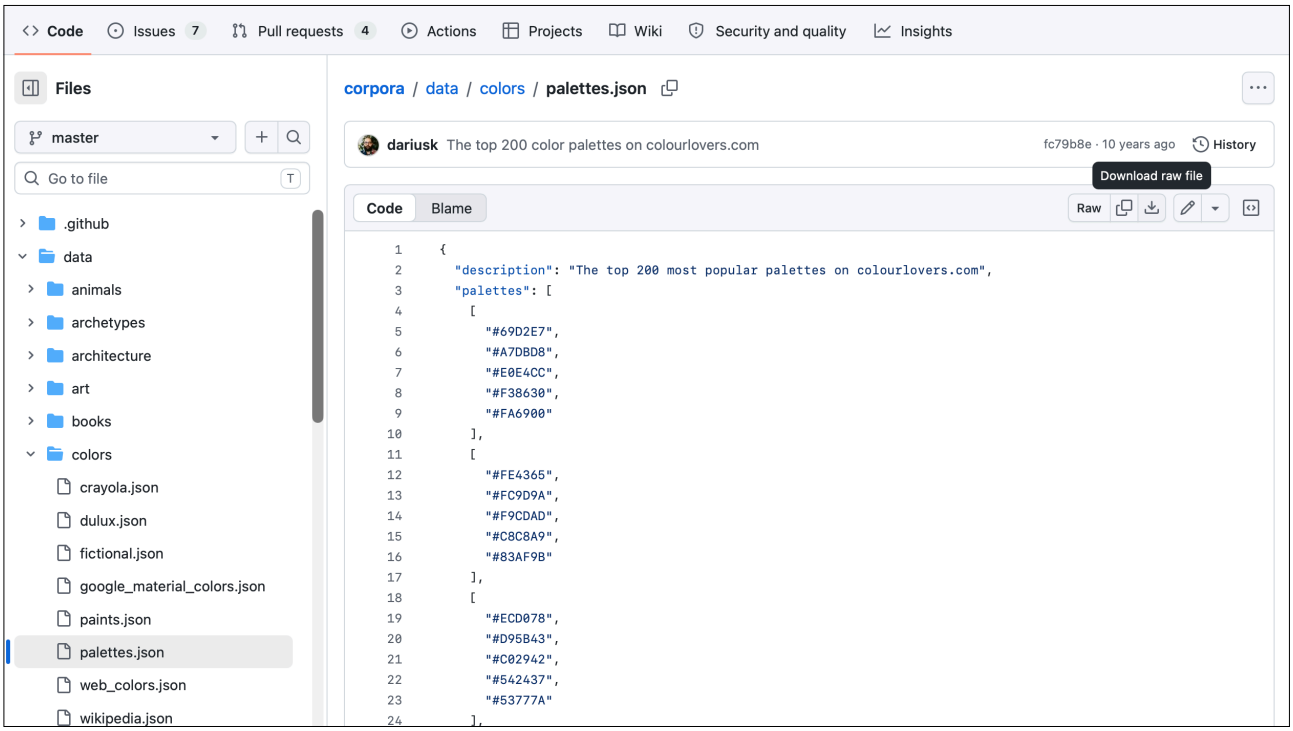
Always try to find a way of using the fewest lines of code.

Using a pre-made .json file

We are going to use a pre-made JSON file that has a collection of some of the favourite colour palettes in hex values. We can access this file and use the five colours in each set of palettes; there are 200 palettes altogether.

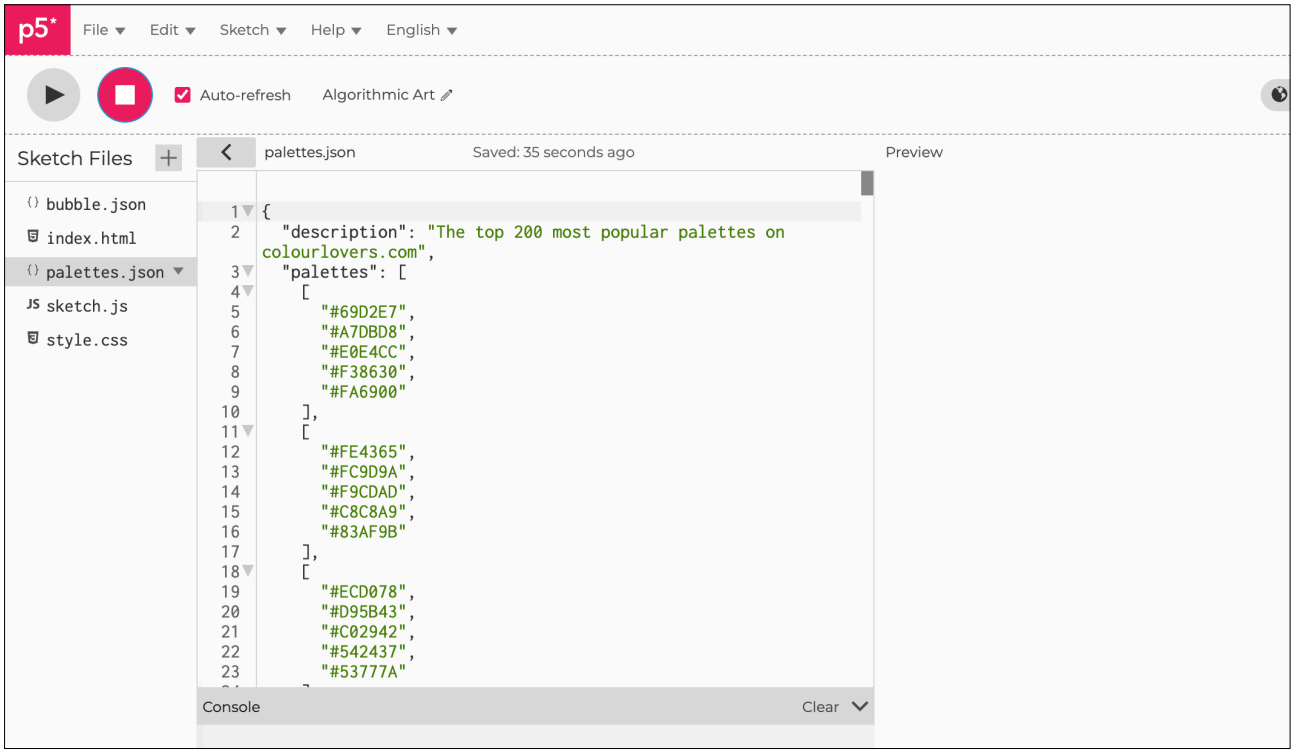
To get the JSON file, follow the link here or use the link on the website to download the file:
https://github.com/dariusk/corpora/blob/master/data/colors/palettes.json
You may notice there are other collections that might be of interest.

Figure 2: download the file



Do the usual upload:

Figure 3: the uploaded file



Sketch F9.7 console log

❗ Start a new sketch.

The great thing about the console log is that it can help you understand the structure of a file such as this one. We are going to upload the file and check what is inside.

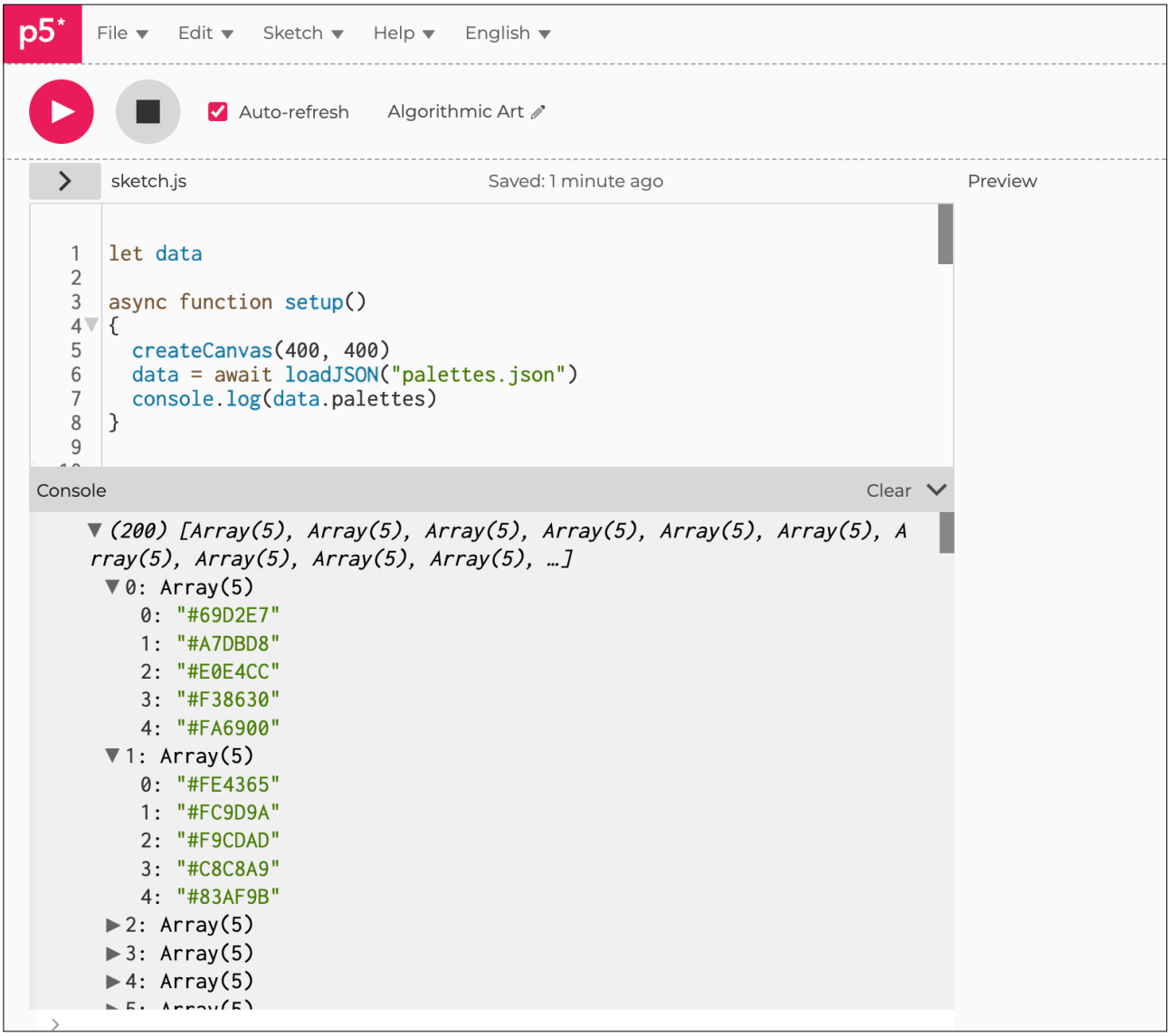
sketch.js

```
let data


async function setup()
{
  createCanvas(400, 400)
  data = await loadJSON("palettes.json")
  console.log(data.palettes)
}
```

Notes

This gives us a comprehensive list of all the palettes and their colours, as you could see in the GitHub repository.

Figure F9.7


p5*
File
Edit
Sketch
Help
English
Auto-refresh
Algorithmic Art
sketch.js
Saved: 1 minute ago
Preview
let data
async function setup()
{
createCanvas(400, 400)
data = await loadJSON("palettes.json")
console.log(data.palettes)
}
Console
Clear
▼ (200) [Array(5), Array(5), Array(5), Array(5), Array(5), Array(5), Array(5), Array(5), Array(5), Array(5), …]
▼ 0: Array(5)
0: "#69D2E7"
1: "#A7DBD8"
2: "#E0E4CC"
3: "#F38630"
4: "#FA6900"
▼ 1: Array(5)
0: "#FE4365"
1: "#FC9D9A"
2: "#F9CDAD"
3: "#C8C8A9"
4: "#83AF9B"
▶ 2: Array(5)
▶ 3: Array(5)
▶ 4: Array(5)

Sketch F9.8 palette number 10

If we want to look inside one palette, we need to specify which one. This is palette number 10 (counting starts at 0).

sketch.js

```
let data


async function setup()
{
  createCanvas(400, 400)
  data = await loadJSON("palettes.json")
  console.log(data.palettes[10])
}
```

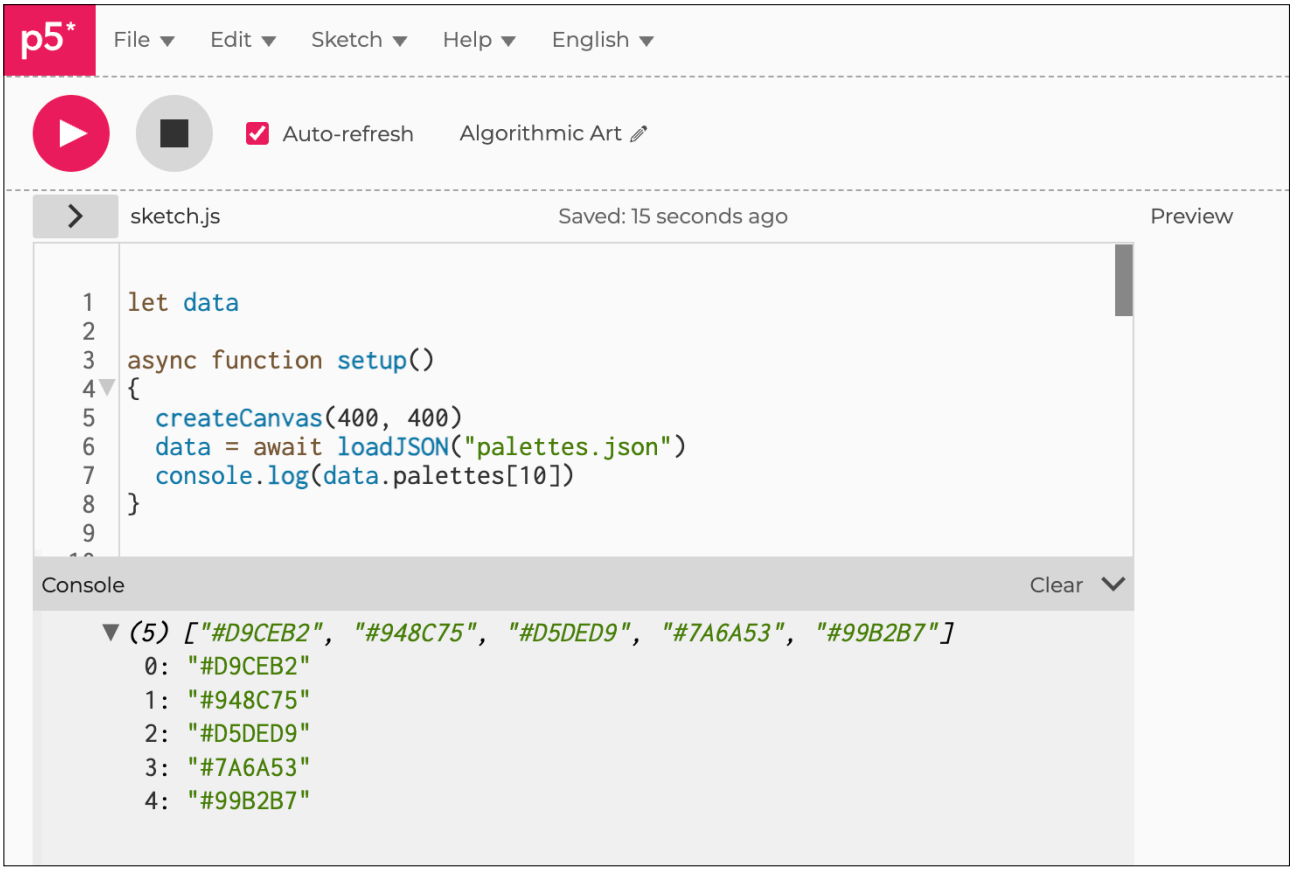
🗒️ Notes

Randomly picked a palette.

🌻 Challenge

Select your own palette.

Figure F9.8


p5*
File
Edit
Sketch
Help
English
Auto-refresh
Algorithmic Art
sketch.js
Saved: 15 seconds ago
Preview
let data
async function setup()
{
createCanvas(400, 400)
data = await loadJSON("palettes.json")
console.log(data.palettes[10])
}
Console
Clear
(5) ["#D9CEB2", "#948C75", "#D5DED9", "#7A6A53", "#99B2B7"]
0: "#D9CEB2"
1: "#948C75"
2: "#D5DED9"
3: "#7A6A53"
4: "#99B2B7"

Sketch F9.9 specific colour

To select just one colour from that palette of five colours, we specify it thus. In this case, it is colour number 2 (the 3rd colour) in palette number 10 (the 11th palette).

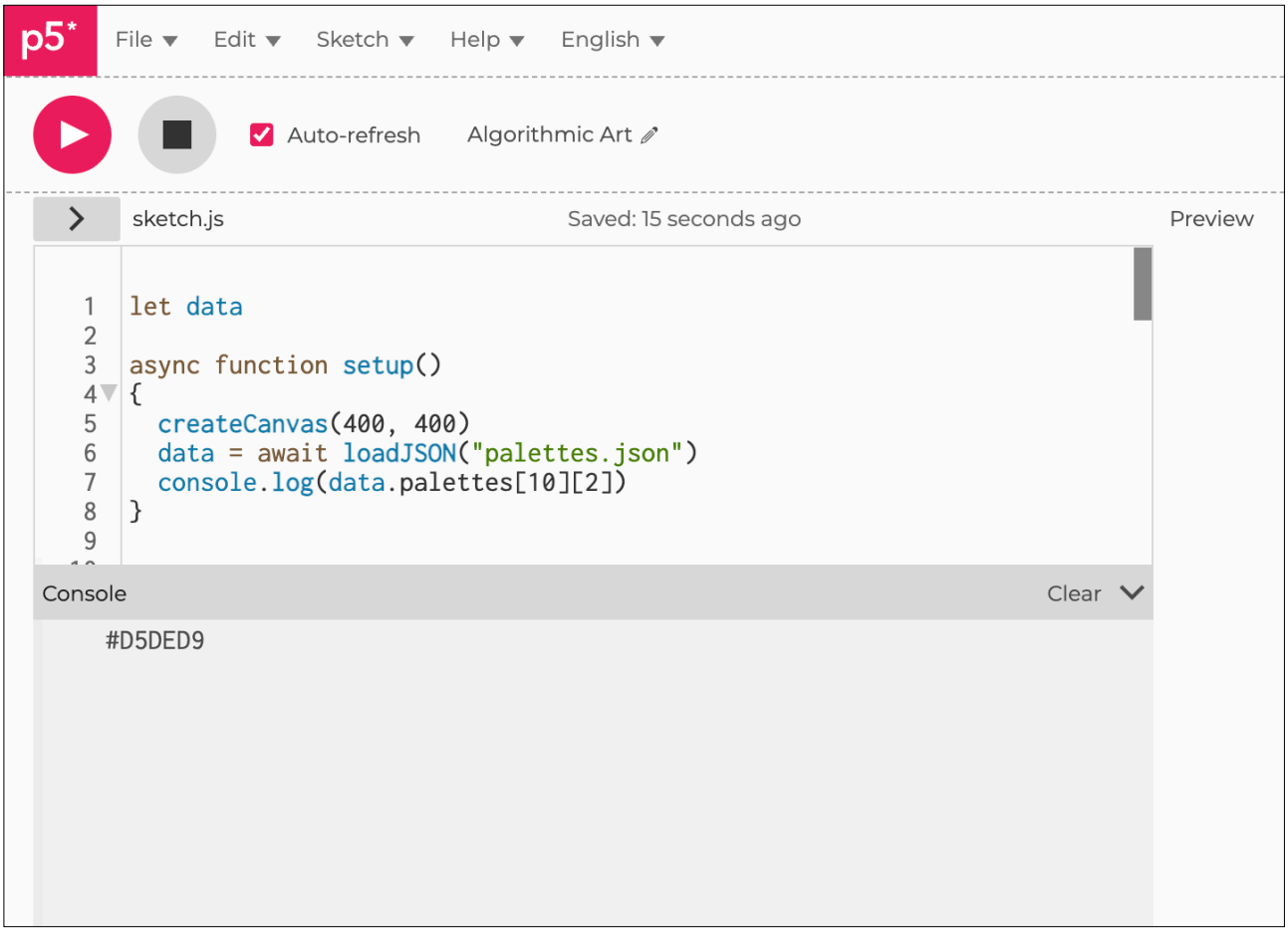
sketch.js

```
let data


async function setup()
{
  createCanvas(400, 400)
  data = await loadJSON("palettes.json")
  console.log(data.palettes[10][2])
}
```

Notes

Unless you speak hex, it will be a bit meaningless.

Figure F9.9


p5*
File
Edit
Sketch
Help
English
Auto-refresh
Algorithmic Art
sketch.js
Saved: 15 seconds ago
Preview
let data
async function setup()
{
createCanvas(400, 400)
data = await loadJSON("palettes.json")
console.log(data.palettes[10][2])
}
Console
Clear
#D5DED9

Sketch F9.10 fill the circle

Let's have a look at it.

❗ Remove console log.

sketch.js

```
let data


async function setup()
{
  createCanvas(400, 400)
  data = await loadJSON("palettes.json")
}


function draw()
{
  background('white')
  fill(data.palettes[10][2])
  circle(100, 100, 100)
}
```

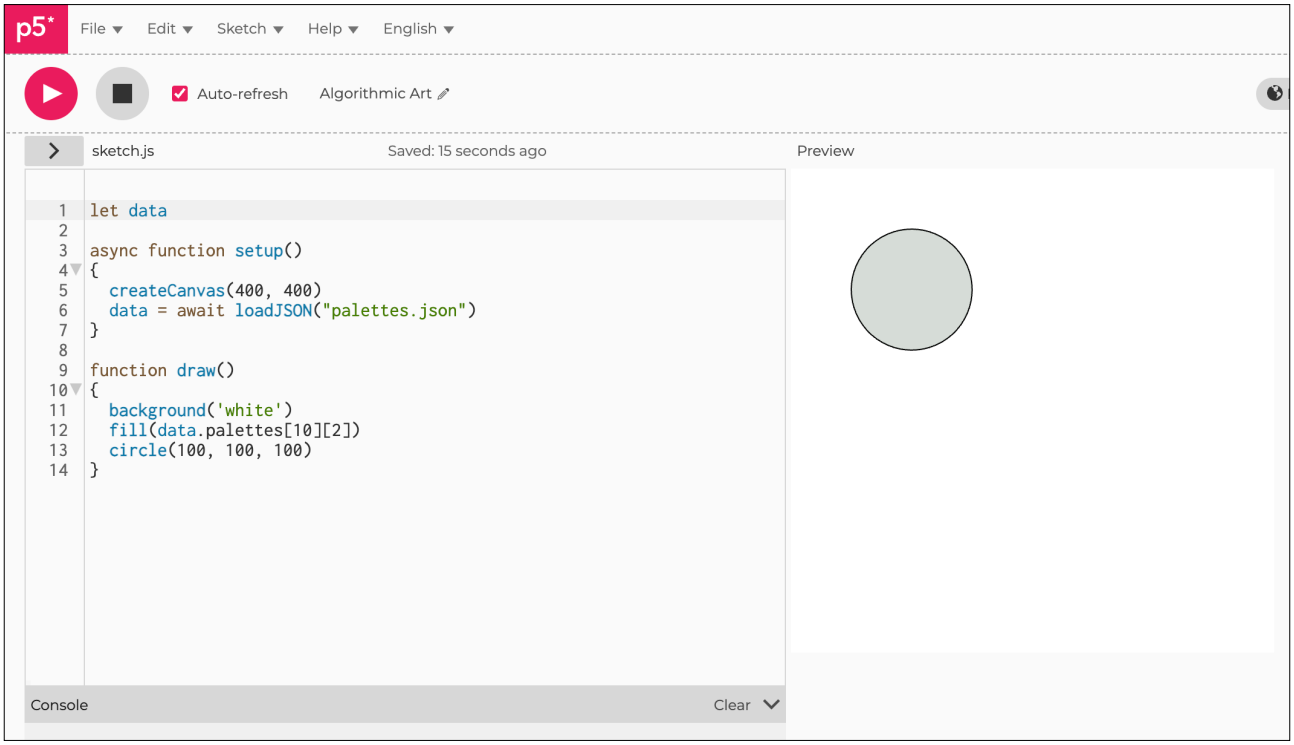
🗒️ Notes

We get a rather dull colour, but it is a palette colour, and I did choose it randomly.

🌻 Challenges

1. Select another one and see what you get.
2. What would be nice is to scroll through all the palettes one by one. Can you think of a way of doing that?

Figure F9.10


p5*
File
Edit
Sketch
Help
English
Auto-refresh
Algorithmic Art
sketch.js
Saved: 15 seconds ago
Preview
let data
async function setup()
{
createCanvas(400, 400)
data = await loadJSON("palettes.json")
}
function draw()
{
background('white')
fill(data.palettes[10][2])
circle(100, 100, 100)
}
Console
Clear

Sketch F9.11 cycle through the palette

I won't highlight all the changes, as there are many. The main function is to iterate through each palette one at a time.

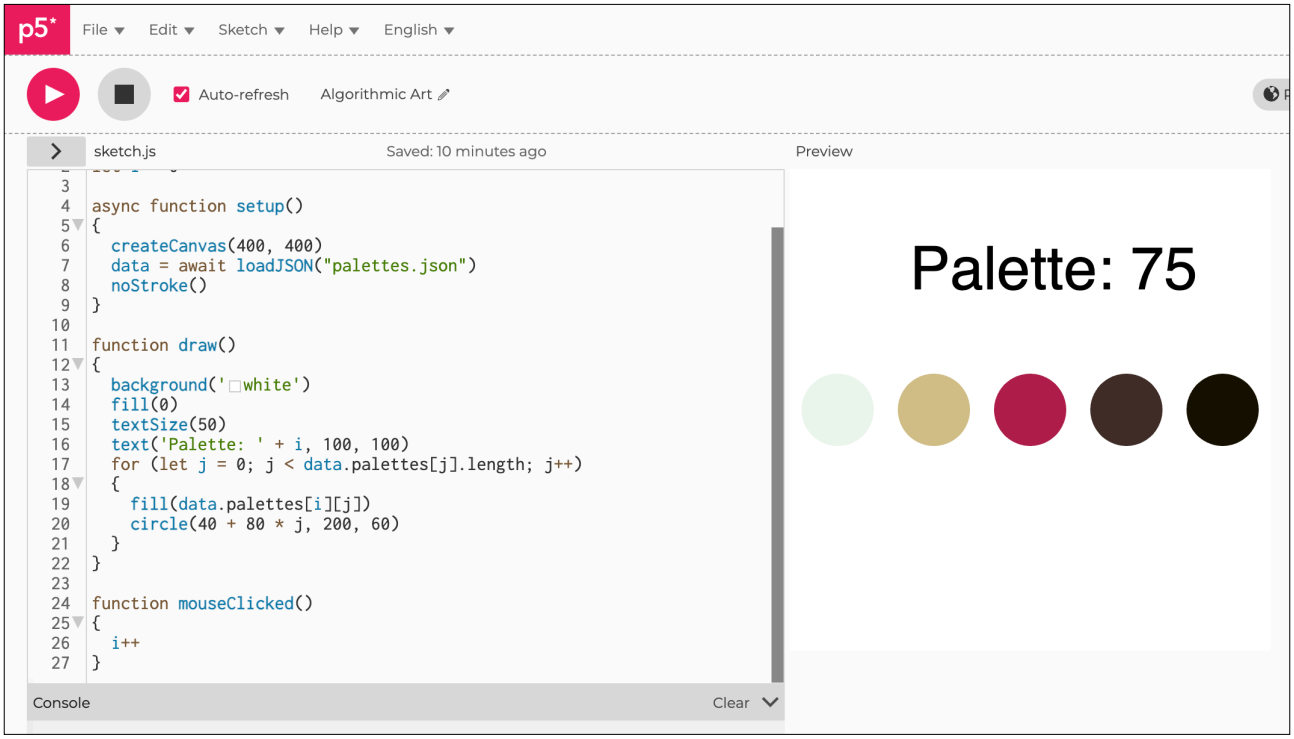
sketch.js

```
let data
let i = 0

async function setup()
{
  createCanvas(400, 400)
  data = await loadJSON("palettes.json")
  noStroke()
}

function draw()
{
  background('white')
  fill(0)
  textSize(50)
  text('Palette: ' + i, 100, 100)
  for (let j = 0; j < data.palettes[j].length; j++)
  {
    fill(data.palettes[i][j])
    circle(40 + 80 * j, 200, 60)
  }
}

function mouseClicked()
{
  i++
}
```

🗒️ Notes

Every time you click the mouse, you get the next palette.

Figure F9.11


p5*
File
Edit
Sketch
Help
English
Auto-refresh
Algorithmic Art
sketch.js
Saved: 10 minutes ago
Preview
3
4 async function setup()
5 {
6 createCanvas(400, 400)
7 data = await loadJSON("palettes.json")
8 noStroke()
9 }
10
11 function draw()
12 {
13 background('white')
14 fill(0)
15 textSize(50)
16 text('Palette: ' + i, 100, 100)
17 for (let j = 0; j < data.palettes[j].length; j++)
18 {
19 fill(data.palettes[i][j])
20 circle(40 + 80 * j, 200, 60)
21 }
22 }
23
24 function mouseClicked()
25 {
26 i++
27 }
Palette: 75
Console
Clear

Sketch F9.12 saveJSON()

❗ Starting a new sketch.

Inserting the two bubble JSON data into the main sketch this time. We can now save this as our own JSON file. Click on the canvas and then find the saved file and open it with whatever software it offers to open it if you want to look inside.

sketch.js

```
let data

function setup()
{
  createCanvas(400, 400)
  background(220)
  textSize(20)
  text("click here", 50, 50)
}

function mousePressed()
{
  data = {
    bubble: [
      {
        name: "a red bubble",
        x: 200,
        y: 300,
        r: 255,
        g: 0,
        b: 0,
        diameter: 100,
      },
      {
        name: "a green bubble",
        x: 250,
        y: 150,
        r: 0,
        g: 255,
        b: 0,
```

```
        diameter: 50,
      },
    ],
  }
  saveJSON(data, "bubbles")
}
```

🗒️ Notes

If you can look inside it, this is what you see.

🌻 Challenge

1. Try loading this or some other JSON file you have created.
2. Create your own palette, perhaps using RGB values instead.

Figure F9.12

bubbles.json

```
{
  "bubble": [
    {
      "name": "a red bubble",
      "x": 200,
      "y": 300,
      "r": 255,
      "g": 0,
      "b": 0,
      "diameter": 100
    },
    {
      "name": "a green bubble",
      "x": 250,
      "y": 150,
      "r": 0,
      "g": 255,
      "b": 0,
      "diameter": 50
    }
  ]
}
```