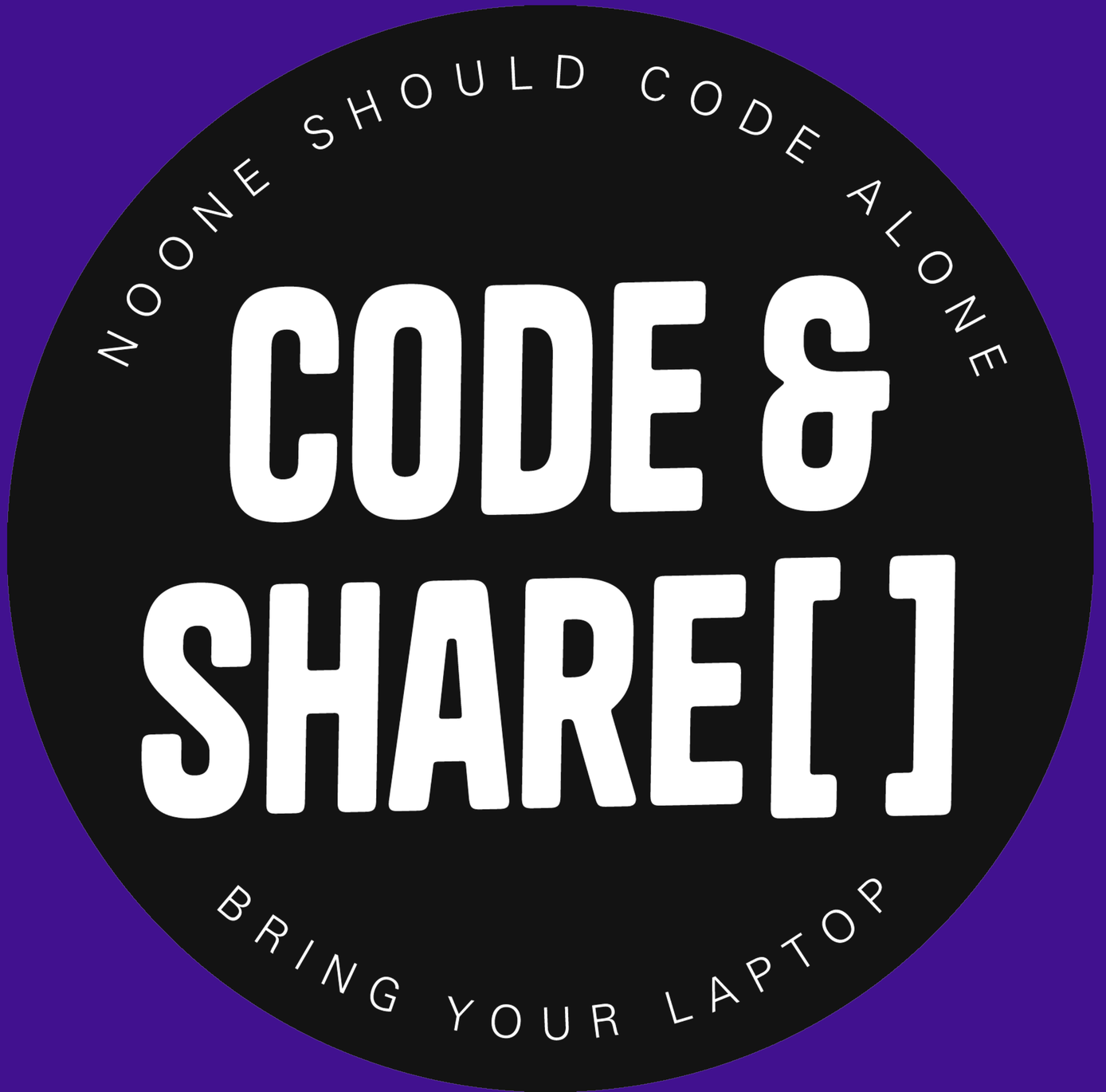
Anders Visti
artist / programmer

!=null
Code&Share[ ]
PCD @Aarhus
https://andersvisti.dk

Code&Share[ ]

A public forum for artists, researchers, developers and hackers using contemporary technology for creative expression and aesthetic inquiry

https://codeandshare.net



Presentations

Please say a few things about yourself:
_ Whats your name?
_ What school, institution, organisation or company do you represent?
_ What are you hoping to gain from these workshops?

History of the web
Web1.0/2.0/3.0 - 1989 to now

Web3.0 *(2014–...)*:
Web3 is an idea for a new iteration of the World Wide Web which incorporates concepts such as decentralization, blockchain technologies, and token-based economics.

The term "web3" was coined in 2014 by Ethereum co-founder Gavin Wood, and the idea gained interest in 2021 from cryptocurrency enthusiasts, large technology companies, and venture capital firms.
https://en.wikipedia.org/wiki/Web3

Web2.0 *(2004–2014)*:
With Web 2.0, it became common for average web users to have social-networking profiles (Myspace and Facebook) and personal blogs (Blogger, Tumblr and LiveJournal) through either a low-cost web hosting service or through a dedicated host.

In general, content was generated dynamically, allowing readers to comment directly on pages in a way that was not common previously.
https://en.wikipedia.org/wiki/Web_2.0

Web1.0 *(1989–2004)*:
Web 1.0 is a retronym referring to the first stage of the World Wide Web's evolution, from roughly 1989 to 2004 ... Personal web pages were common, consisting mainly of static pages hosted on ISP-run web servers, or on free web hosting services such as Tripod and the now-defunct GeoCities.
https://en.wikipedia.org/wiki/Web_2.0#Web_1.0

Technology stacks and technical debt

What would happen if we stripped away site builders, frameworks and content management systems?
And why would we want to do that?
…

Let’s talk about technical debt.

So, why are we building websites by hand using HTML, CSS and Javascript?

Because:

_ We know precisely what our technical debts are. Ok, sometimes stuff breaks, but we can fix things—and we have access, always.

_ The site cannot be hacked or break, because it’s static HTML.

_ It’s very stable over time, because the site is using standards compliant technologies.

_ The site loads superfast. It doesn’t have to communicate with a database and then render the HTML.

_ It’s very flexible. If you know how, you can build anything you want …

_ Also, importantly: You get a better sense of what the web is all about

And also, HTML is a vibe:

HTML energy
https://html.energy/

Olia Lialina, the vernacular web
https://art.teleportacia.org/observation/vernacular/

Let’s begin by downloading an app:

Visual Studio Code

When we’ve downloaded and installed VSCode,
we’ll install an extension called Live Server

1.

Make a folder on your laptop

Name this folder ”myWebsite” or something similar that describes the content of the folder. This folder is where your websites ”lives” locally on your computer.

Open this folder in Visual Studio Code

2.

Making the first HTML-document

Press the button ”new file” in VSCode and name this file index.html

This file is our homepage, and the server automatically finds the page called index.html...

HTML
(H)yper(T)ext (M)arkup (L)anguage

HTML is not a programming language, but a protocol to structure content on the web with.
The *html*-, *head*- og *body* elements (as seen on the right) define the basic structure of an html document.
*

HTML was invented by the physicist Tim Berners-Lee in 1990 while working at the CERN research facility in Geneva, Switzerland. His intention was to give CERN scientists a simple markup language for sharing their research results with each other on the CERN intranet.

```
1  <!DOCTYPE html>
2  <html>
3    <head></head>
4    <body></body>
5  </html>
```

The document title

Write the title of the document in the code, and also — Let's start our local live server.

When we've started the live server, we can see our code in the browser.

```
<!DOCTYPE html>
<html>
  <head>
    <title>My Own Website</title>
  </head>
  <body></body>
</html>
```

Some common HTML elements:

Paragraphs,
Headers,
Links

```
<!DOCTYPE html>
<html>
  <head>
    <title>My Own Website</title>
  </head>
  <body>
    <h1>A large header</h1>
    <p>Here’s a paragraph
      that runs over a few lines, and
      <a href=”https://codeandshare.net”>has a link</a>
      that anyone can click on</p>
  </body>
</html>
```

1.

More common (inline) HTML elements

Italic, Bold, Underline

```
<p>Here’s a <b>paragraph</b> that
<u>runs</u> over a <i>few lines</
i>, and <a href=”https://code-
andshare.net”>has a link</a> anyo-
ne can click on</p>
```

2.

... If I want to open links in a new window or tab in my browser

```
... <a href=”https://codeandshare.net”
target=”_blank”>has a link</a> ...
```

Insert an image

Make a new folder called "images" and download an image to this folder.

Then "point" to this image file from your code ...

```
...
  <p>Here's a paragraph
    that runs over a few lines, and
    <a href="https://codeandshare.net">has a link</a>
    that anyone can click on</p>
   <img src="images/rotating.gif" />
...
```

CSS
(C)ascading (S)tyle (S)heets

CSS is used to describe the style and visual appearance of the elements on the page.
HTML is structuring the content of the page and CSS describes how it looks.

```
<!DOCTYPE html>
<html>
  <head>
    <title>My Own Website</title>
    <style>
      body {
        background-color: pink;
        color: black;
      }
    </style>
  </head>
  ...
</html>
```

CSS classes

We can build our own ”classes”, and assign specific styles to certain elements in our page.

Let’s put our main content in a class called ”mainContent”, so that we can give this element some visual features.
For instance, we could limit the width of this elemen, so that the text inside is not too wide.

```
...
  <body>
    <main class=”mainContent”>
       <!-- our content here -->
    </main>
  </body>
...
```

CSS class for mainContent

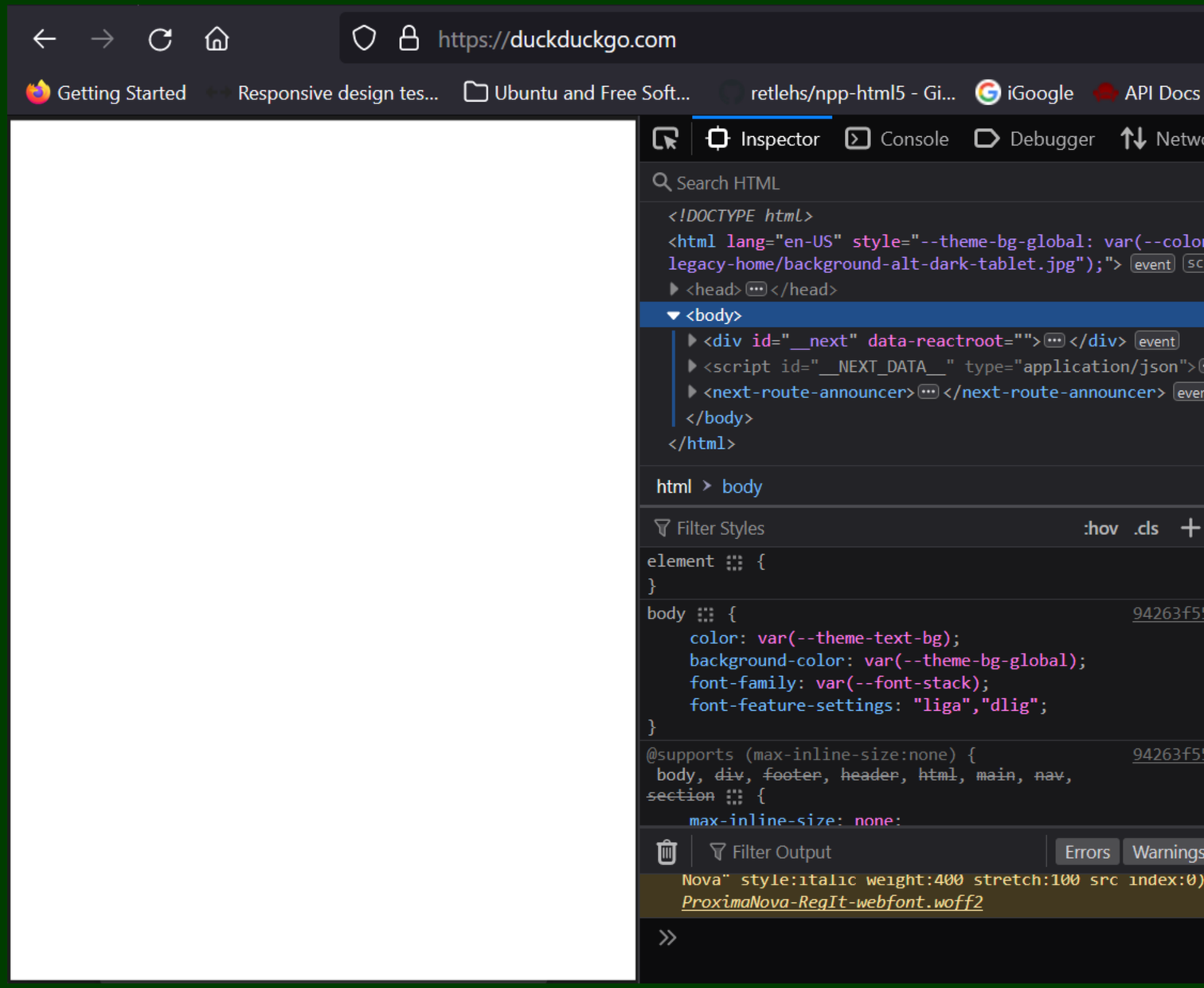
Let’s define the rules for the appearance on this class

```
...
  <style>
    body {
      background-color: pink;
      color: white;
    }

    .mainContent {
      max-width: 60ch;
      margin: 0px auto;
    }
  </style>
...
```

Developer Tools

The developer tools are a browser's built-in tools that help you structure and style your HTML content.

Right click somewhere on the page and select the menu item "inspect"



.pageWrapper

It’s very practical with an element and a class that ”wraps” the entire content of the page...

```
<body>
  <div class=”pageWrapper”>

    ...

  </div>
</body>
```

.pageWrapper

Let’s style pageWrapper

```
<style>
  ...

  .pageWrapper {
    border: 10px solid pink;
  }
</style>
```

Lets build a menu

We'll start by making the page *"about.html"*
(Because it's so common, let's just all do that)

So, how do we link *index.html* and *about.html* to each other?

*

The header element works as our page header, and it makes sense to place our menu there.

```
...
  <div class="pageWrapper">

    <header>
      <nav>
        <a href="index.html">forside</a>
        <a href="about.html">about</a>
      </nav>
    </header>

    <main class="mainContent">

      ...

    </main>

  </div>
```

CSS linked externally

Now we're having more than one page in our site, and it makes sense to have one file with our css, so that we can manage the looks of our content from one place (one file)

Make a folder called "css". And make a file inside this folder called "style.css"

Then, put the code on the right in both *index.html* and *om.html*

```
...
  <head>
    ...

    <link rel="stylesheet" href="css/style.css">
  </head>
```

Make your own content

Try to put images and text on the page yourself.

For example, you could:
- Build a portfolio site unlike any other portfolio site?
- Build a fan page for you favorite band?
...or your favorite film director?
- Make a web museum of all the mundane everyday objects
in your purse or your bag?
- Make a web photo album of all the pets
you ever owned?

It has to be something personal. Something you care about ...

Let’s repeat the HTML elements we’ve talked about:

html, head, body, title, p, a, img, b,
i, u, main, header, link, div

Other common elements:
https://web.stanford.edu/group/csp/cs21/htmlcheatsheet.pdf

Let’s add this code to the <head>

Put these four lines in <head> on all of your pages.
view-source:https://codeandshare.net/

```
<head>
  <meta charset=”utf-8”>
  <meta name=”description” content=”My website is awesome!”>
  <meta name=”viewport” content=”width=device-width, initial-scale=1.0”>
  <meta http-equiv=”X-UA-Compatible” content=”ie=edge”>
  ...
</head>
```

Add this to your style.css

Write this in the top of your stylesheet...

```
*, *:after, *:before {
  box-sizing: border-box;
}
```

Intro to Web Programming
Part II: Javascript
Aarhus May 15th, 09.00–15.30
Nobelparken—Bygning 1453–616
*

Workshop lead:
Anders Visti, artist / programmer
!=null
Code&Share[ ]
PCD @Aarhus
https://andersvisti.dk

Let’s begin by downloading an app:
Visual Studio Code

When we’ve downloaded and installed VSCode, we’ll install an extension called Live Server

Make a folder on your laptop. Name this folder *”myWebsite”* or something similarly descriptive.
Open this folder in Visual Studio Code.

Make a new HTML file. Name it *”index.html”*.
Open this file in the live server.

HTML and CSS, repetition

Let’s run through the content of the last workshop again, so that it’s present in our minds before we move on.
The content we’re looking on today builds on top of our work with HTML and CSS from last week.
*

Last workshop we talked about:
html, head, body, title, h1, h2, p, a, img, b, i, u, main, header, link, div

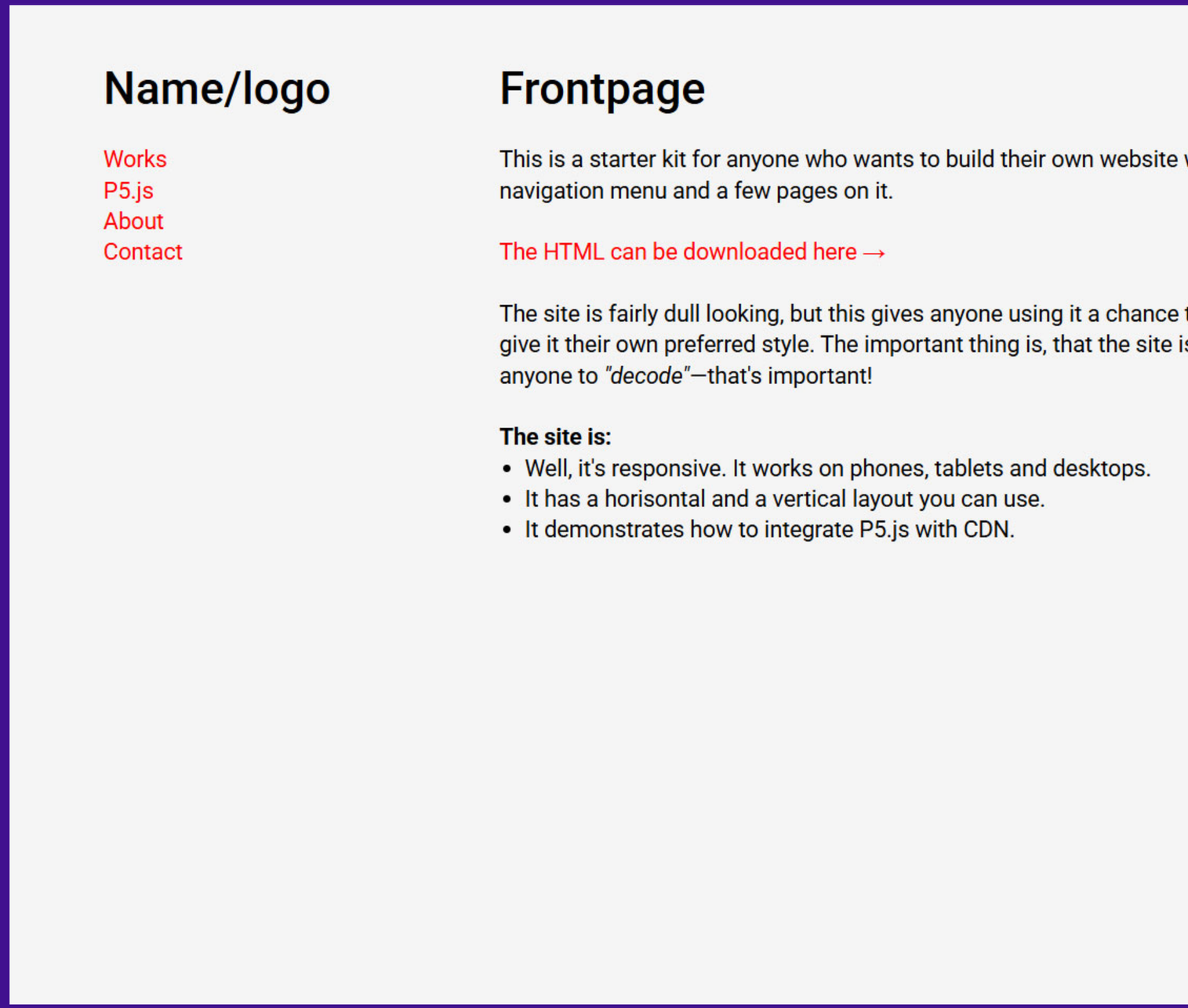
```
<!DOCTYPE html>
<html>
  <head></head>
  <body></body>
</html>
```

StarterKit.zip

I’ve made a HTML starter kit for you. It’s a kit for anyone who wants to build their own website with a simple navigation menu and a few pages on it.

The design is dull, but tweakable—and that’s the point. It’s responsive. It has a horisontal and a vertical layout and it demonstrates how to integrate P5.js with CDN.

Download here:
https://codeandshare.neocities.org/



Javascript

Let’s talk about what Javascript is ...

- HTML, CSS and Javascript are the three ”languages” (markup),
that makes our browsers work the way we’re familiar with.

Javascript handles the *”events”* that occur over time in the browser.
That could be something like form validation, mouse movement or similar types of interactivity.

Javascript is a programming language, unlike HTML and CSS.
(Why? - because JS does variables, conditional statements and iterative loops)
JS have become very powerful in recent years, and is now full stack.
That means we can build apps with it. For example *Visual Studio Code* ...

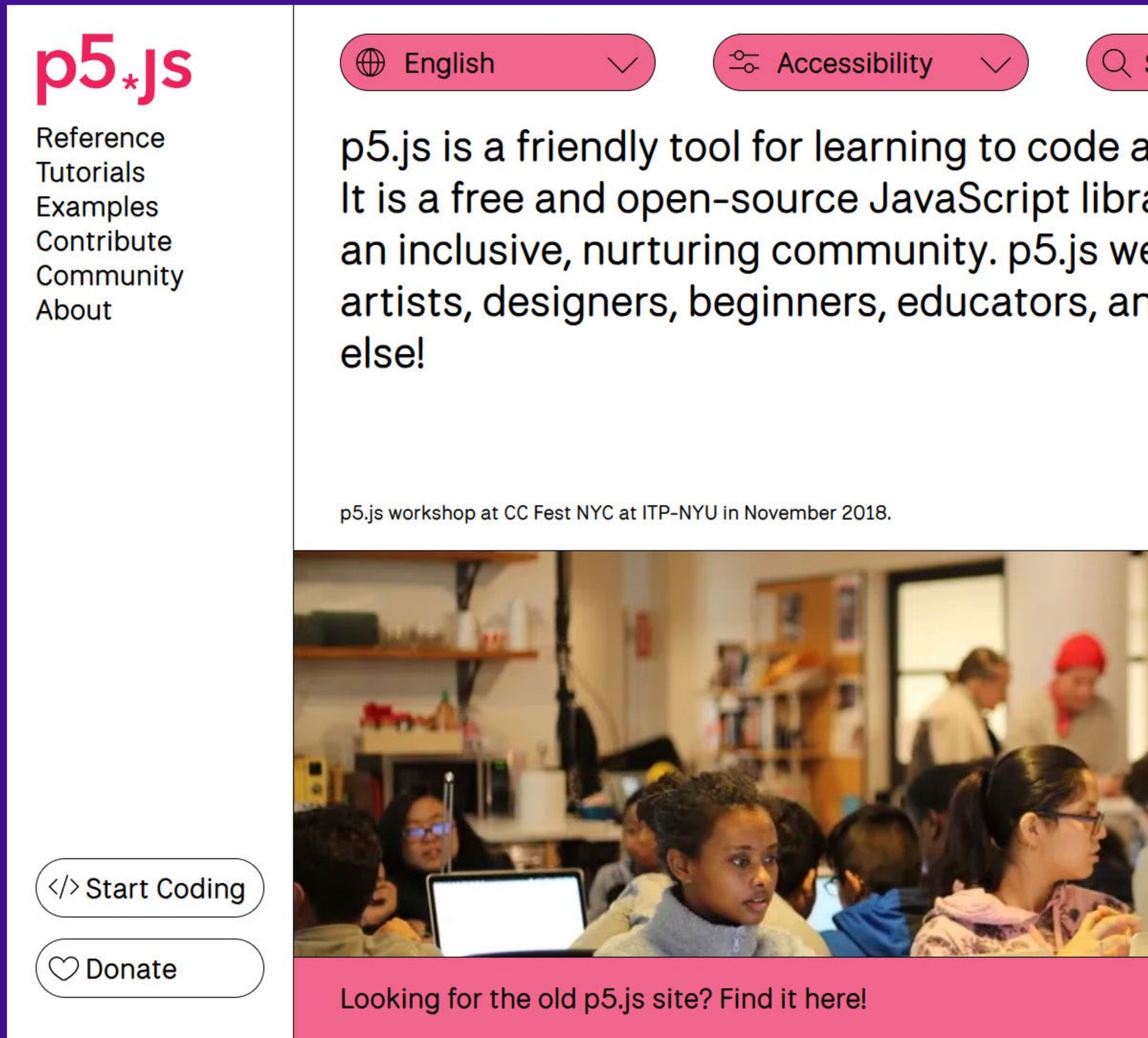
Javascript

Before we start working with p5.js,
let’s talk about DOM manipulation
https://www.freecodecamp.org/news/dom-manipulation-in-javascript/

Anders demonstrates how to *”tamper”* with the html on a webpage.

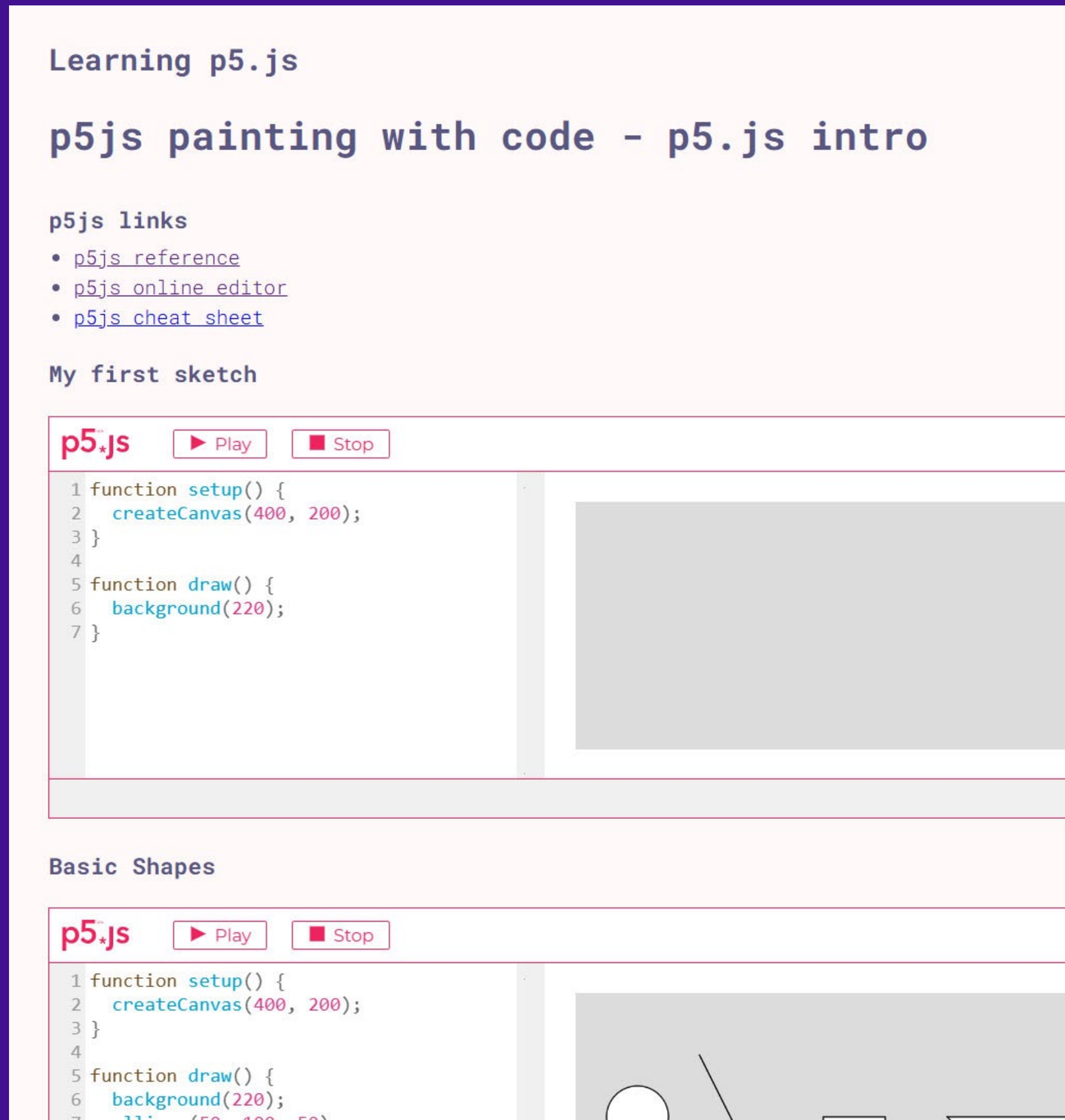
p5.js

p5.js is a friendly tool for learning to code and make art. It is a free and open-source JavaScript library built by an inclusive, nurturing community. p5.js welcomes artists, designers, beginners, educators, and anyone else!
https://p5js.org/



p5.js

Let’s look at what p5.js is and how we can work with it. Here’s a page with editors and sample code that we can tweak and test together: codeandshare.neocities.org/learningP5



p5.js

How can we use p5 on our own site? Let's load the p5.js library in the top our of page ...

We'll use the *CDN (Content Delivery Network)* version here: https://p5js.org/download/

...and put it in the <head> section of the page, so that it loads straight away when the page loads ...

```
<!DOCTYPE html>
<html>
  <head>
    <title>My Own Website</title>
    <link rel="stylesheet" href="css/style.c
    <script src="..."></script>
  </head>
  <body>
    ...
  </body>
</html>
```

p5.js

.. and then let's put the script in the bottom of the page, right before the closing </body> tag ...

You can copy the code you just made in the online editors, and paste it in here to make it work locally, on your site.

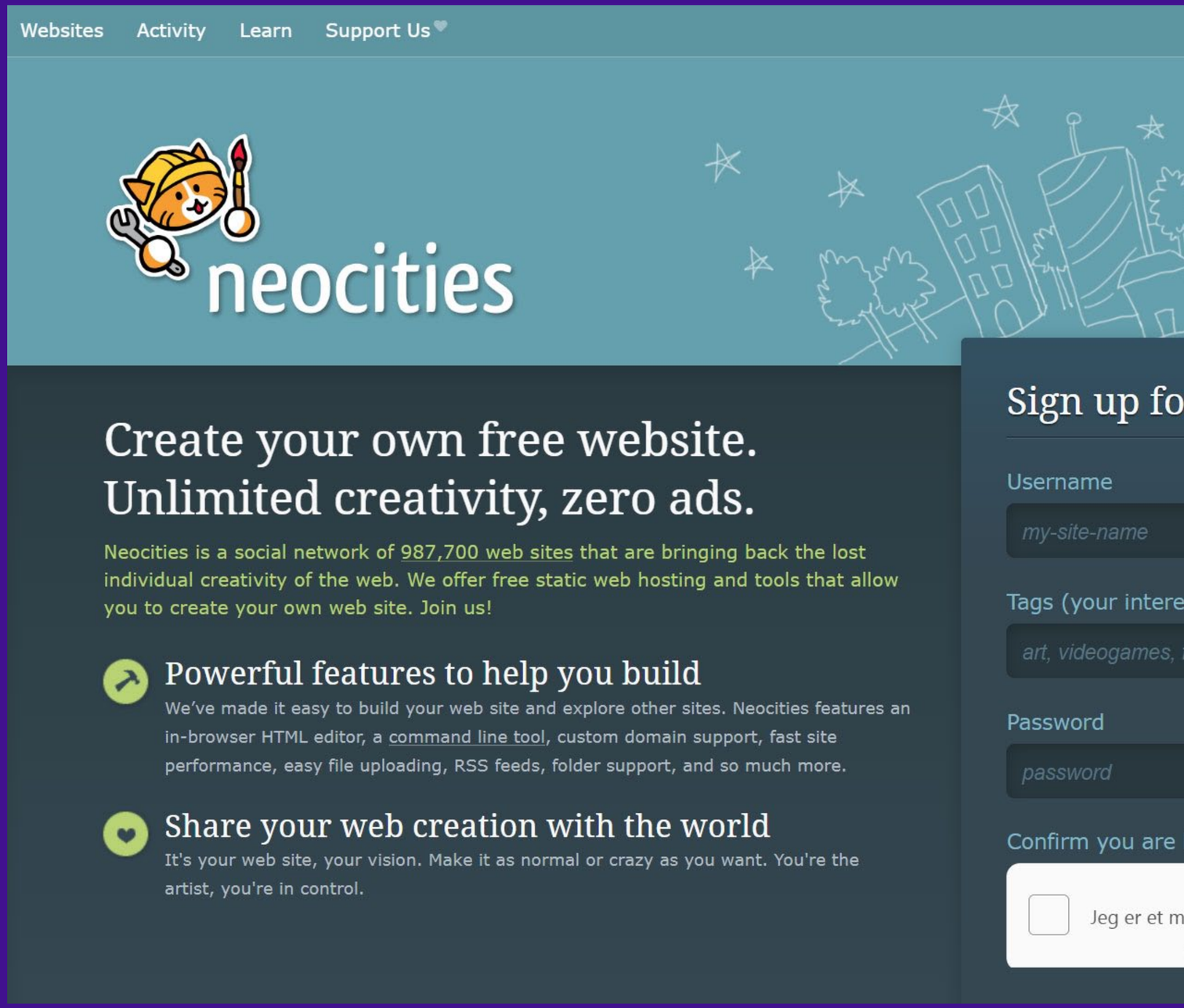
```
<body>
  ...

  <script>

    function setup() {
      createCanvas(400, 200);
    }

    function draw() {
      background(220);
      ellipse(mouseX, mouseY, 50);
    }

  </script>
</body>
</html>
```

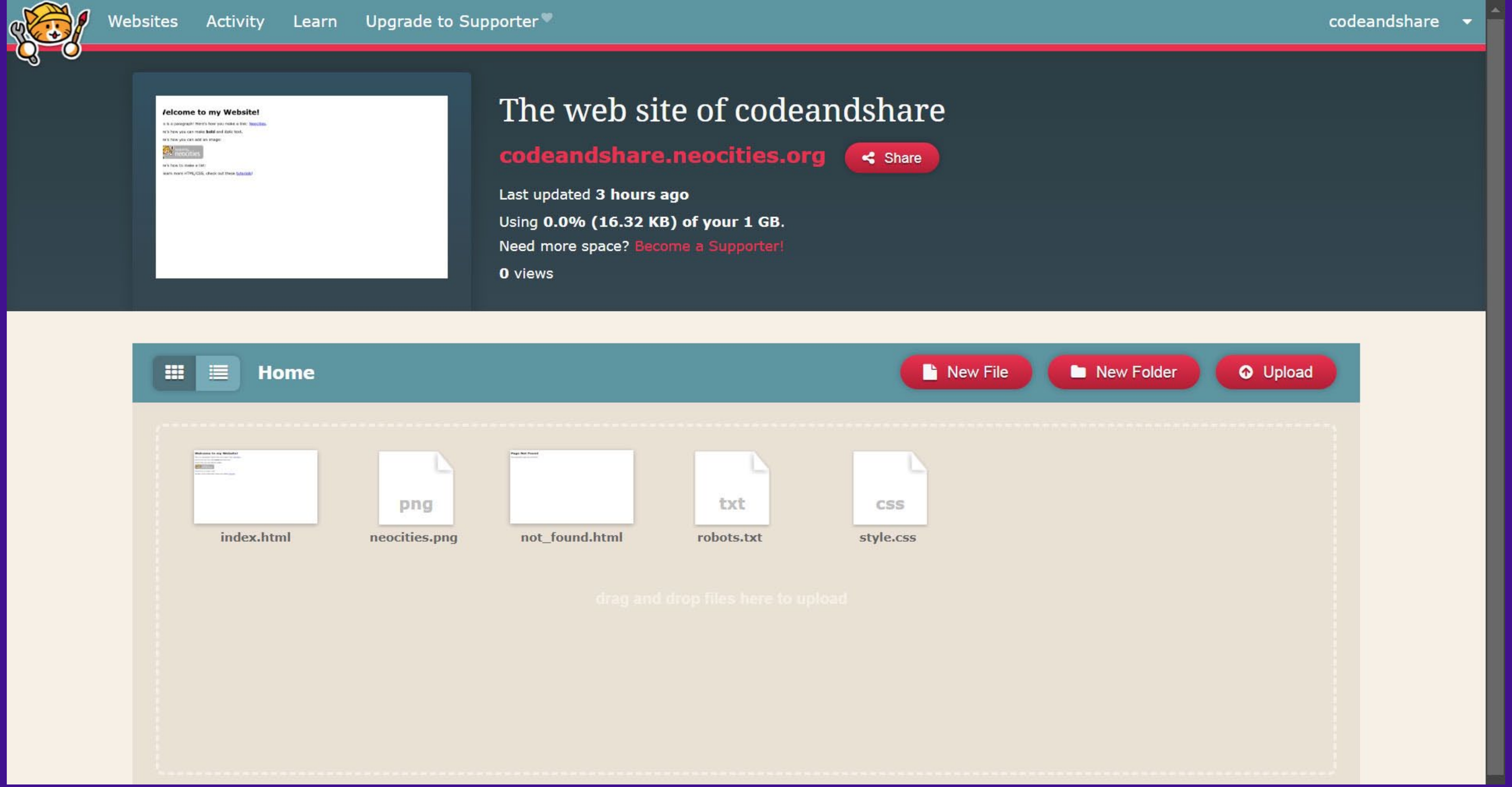
Hosting

Neocities.org let’s us host a website for free, as long as it is less than 1GB.

NeoCities.org is a remake of GeoCities (Yahoo) —a web hosting service that allowed users to create and publish websites for free and to browse user-created websites by their theme or interest, active from 1994 to 2009.
en.wikipedia.org/wiki/GeoCities

Let’s make an account …
https://neocities.org/





NeoCities.org: Logged in user dashboard.
Here you can upload your local site.

Resources, links and tools:

Find easy explanations for every piece of p5.js code. https://p5js.org/reference/

p5.js tutorials:
https://p5js.org/tutorials/

Coding Train, Youtube channel with Daniel Shiffman. Focuses on JavaScript, algorithmic art, machine learning, simulation, generative poetry, and more...
https://www.youtube.com/channel/UCvjgXvBlbQiydffZU7m1_aw
website: https://thecodingtrain.com/

Learn HTML from the ground up:
codecademy.com/learn/learn-html

Olia Lialina, presentation on vernacular web:
youtube.com/watch?v=_hFMOSmonvU

Font embedding, without Google servers:
https://gwfh.mranftl.com/fonts