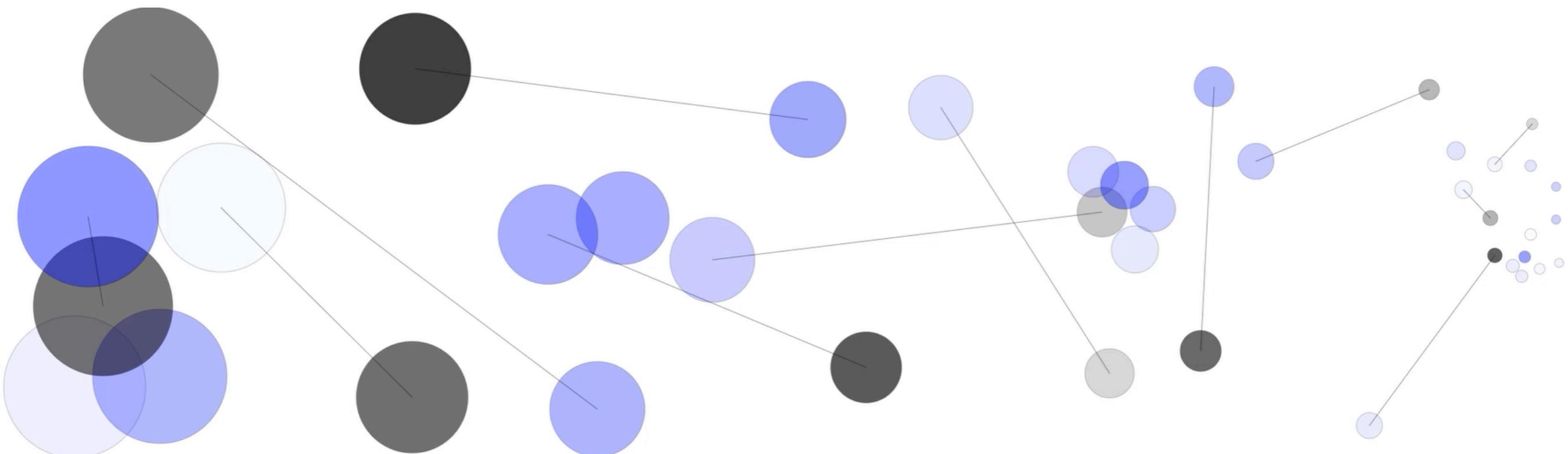


an introduction to WEB audio API
and p5.js sound library

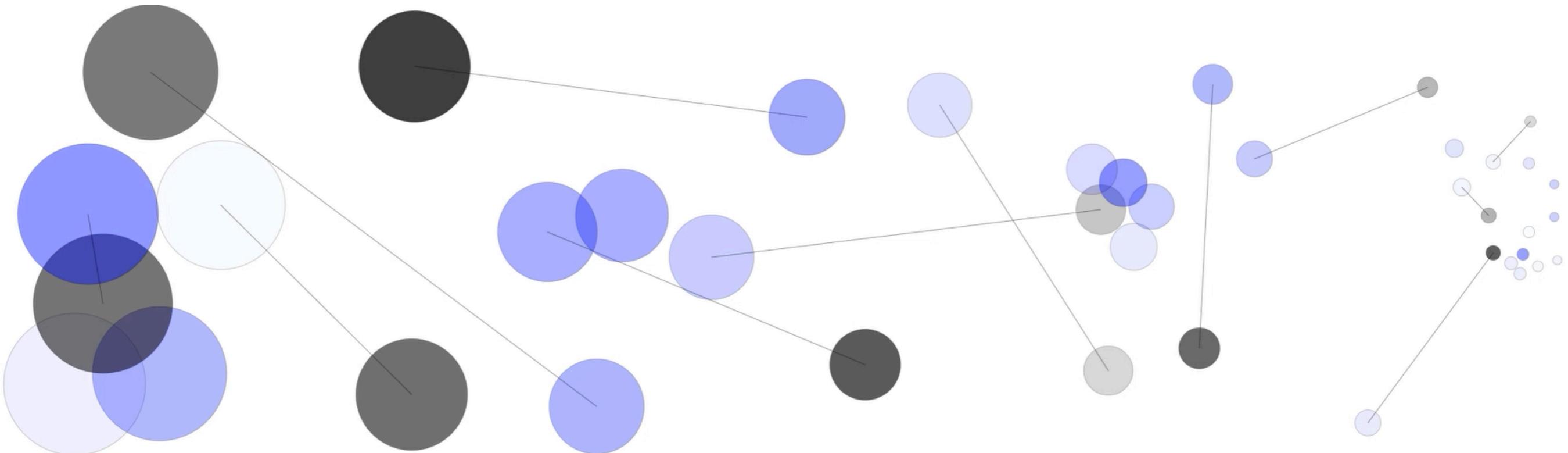
highSCORE



Alberto Barberis
2019

an introduction to WEB audio API and p5.js sound library

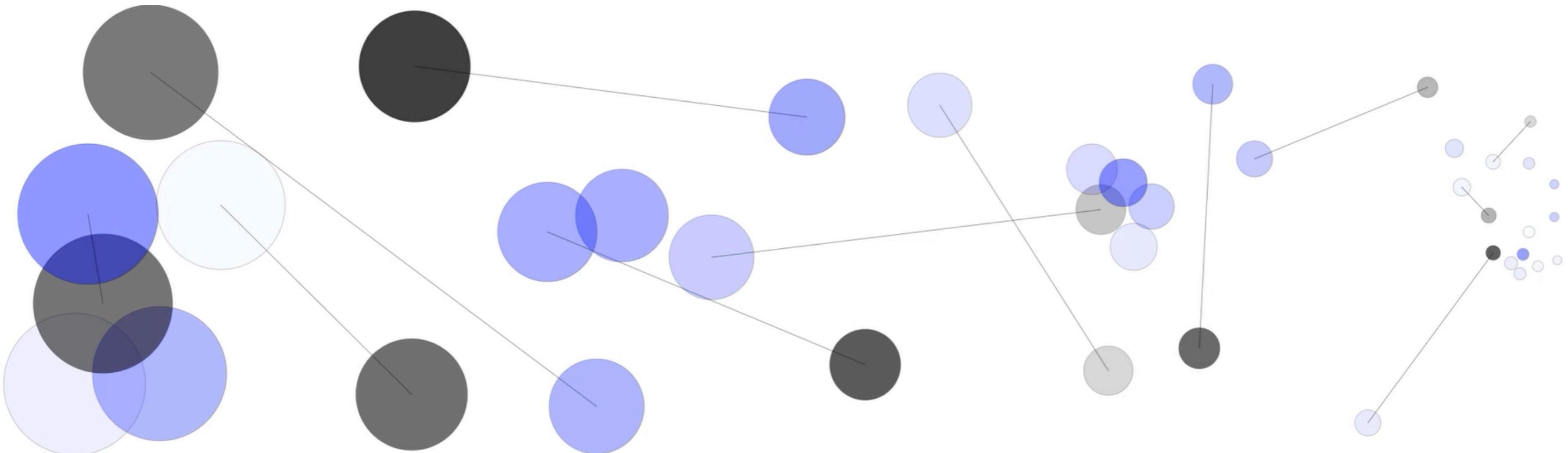
<https://www.albertobarberis.it>



Alberto Barberis
2019

an introduction to WEB audio API and p5.js sound library

https://github.com/tamburo11/hs2019_Drone_Generator

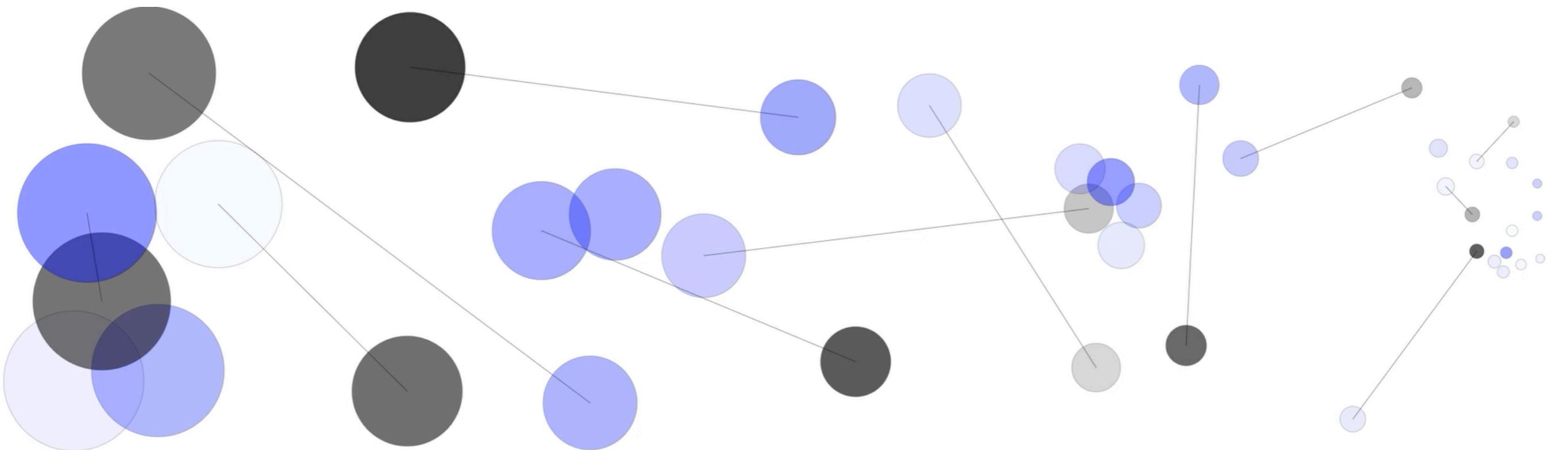


Alberto Barberis
2019

GOALS

1. understand what is the **web audio** and what are the basic concepts of the **web audio API**;
2. analyze together **some examples** of web audio applications;
3. summarize what are the **possibilities** offered by the web audio;
4. understand **what you need to know** to deal with the web audio (web programming: HTML, css, javascript);
5. **develop** your first simple web audio app with the javascript **library p5.js**.

1. WHAT IS THE WEB AUDIO



1.1 WHAT IS THE WEB AUDIO



With the expression **web audio** we refer to the technologies that allow to **process and/or synthesize audio signals in web applications** (running on a browser).

Therefore, the web audio allows you to create music in all on-line devices (smartphone, computer, tablet, smart TV, etc.) which use a browser that supports **web audio**.

1.1 WHAT IS THE WEB AUDIO

The **web audio**, as it is considered today, is a rather new technology, made available thanks to the release of the **web audio API** by the **W3C** (in 2010). Previously, audio could be embedded in browsers, but more complex audio effects were only available through the use of plugins.

The **World Wide Web Consortium (W3C)** is the main international standards organization for the World Wide Web.

Dealing with web application design means primarily dealing with **web programming languages** like HTML, CSS, javascript.

[What is a programming language?](#)

1.2 WHAT IS A PROGRAMMING LANGUAGE

A **programming language** is a formal language used to develop software programs, scripts, or other sets of instructions for computers (programmable machines) to execute.

Thousands of different programming languages have been created, and more are being created every year.

Although many languages share similarities, each has its own **syntax** (rules, structure, vocabulary, etc.) and **semantics** (meaning, logic, characteristics, etc.).

1.2 WHAT IS A PROGRAMMING LANGUAGE

```
<hs2019.html>
1  <!-- WEB AUDIO highSCORE 2019-->
2
3  <!DOCTYPE html>
4
5  <html lang="en">
6
7  <head>
8      <title>web audio HS2019</title>
9      <meta charset="utf-8">
10     </head>
11
12 <body>
13     <div>
14         this is n
15     </div>
16 </body>
17
18 </html>
19
```

HTML

```
1  body {
2      margin: 0px;
3      font-family: 'Courier New', Courier, monospace;
4  }
5  h1 {
6      margin: 0px;
7      padding: 10px;
8      background-color: #rgl
9      color: #white;
10     -webkit-user-select: no
11 }
12 }
```

CSS

```
197     function mouseReleased() {
198         initializeIndex();
199     }
200     function keyReleased(){
201         initializeIndex();
202     }
203
204     function frequencyModulation(i,j){
205
206         var scalar1=(random(300)+50)*random([-1,1]);
207         i.osc.freq(j.freqModulator.mult(scalar1));
208
209         var scalar2=(random(300)+50)*random([-1,1]);
210         j.osc.freq(i.freqModulator.mult(scalar2));
211     }

```

JAVASCRIPT

```
import com.cycling74.max.*;
public class SimpleBiquad extends MSPPerformer
{
    private float freq = 1000.0f;
    private float c, a0, a1, a2, b1, b2, tin1, tin2,tout1, tout2;
    private float pi = 3.1415926f;
    private float r = 1.4142135f;
    private double sr; // sample rate
    private static final String[] INLET_ASSIST = new String[]{
        "input (sig)"
    };
    private static final String[] OUTLET_ASSIST = new String[]{
        "output (sig)"
    };
    public SimpleBiquad()
    {
        declareInlets(new int[]{SIGNAL, DataTypes.ALL});
        declareOutlets(new int[]{SIGNAL});
        setInletAssist(0, "signal input");
        setInletAssist(1, "Cut-Off Frequency");
        setOutletAssist(OUTLET_ASSIST);
        createInfoOutlet(false); // suppress info outlet
    }
    public void inlet(float f)
    {
        freq = f; // this is the cutOff frequency
    }
    public void dspsetup(MSPSignal[] ins, MSPSignal[] outs)
    {
        sr = ins[0].sr; //this is the sample rate
    }
}
```

JAVA

1.2 WHAT IS A PROGRAMMING LANGUAGE

```

18 #include "pitches.h"
19
20 // notes in the melody:
21 int melody[] = {
22     NOTE_C4, NOTE_G3, NOTE_G3, NOTE_A3, NOTE_G3, 0, NOTE_B3, NOTE_C4
23 };
24
25 // note durations: 4 = quarter note, 8 = eighth note, etc.:
26 int noteDurations[] = {
27     4, 8, 8, 4, 4, 4, 4, 4
28 };
29
30 void setup() {
31     // iterate over the notes of the melody:
32     for (int thisNote = 0; thisNote < 8; thisNote++) {
33
34         // to calculate the note duration:
35         // e.g. quarter note = 1000
36         int noteDuration = 1000 / noteDurations[thisNote];
37         tone(8, melody[thisNote], noteDuration);
38
39         // to distinguish the notes
34         // the note's duration + 30
35         int pauseBetweenNotes = noteDuration + 30;
36         delay(pauseBetweenNotes);
37
38         // stop the tone playing:
39         noTone(8);
40     }
41 }
42
43 void loop() {
44     // no need to repeat the melody
45 }
46
47
48
49
50

```

MAX/MSP

ARDUINO IDE (C++)

PROCESSING (JAVA)

```

NOC_1_8_motion101_acceleration Mover ▾
1 class Mover {
2
3     PVector location;
4     PVector velocity;
5     PVector acceleration;
6     float topspeed;
7
8
9     Mover() {
10        location = new PVector(width/2, height/2);
11        velocity = new PVector(0, 0);
12        acceleration = new PVector(-0.001, 0.01);
13        topspeed = 10;
14    }
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50

```

1.2 WHAT IS A PROGRAMMING LANGUAGE

Usually a **developer** writes the source code in a **text editor** or IDE (Integrated Development Environment) and creates a source file (.html, .js, .css, .java, etc.).

Even if a simple text editor (such asTextEdit for Mac) can be used, there are editors dedicated, which contain many support features: Atom, Visual Studio Code, Notepad ++, Sublime Text, etc.

Then the **source code** is compiled or interpreted to be transformed into a code understandable and executable by the machine itself.

[What is web programming?](#)

1.3 WHAT IS THE WEB PROGRAMMING

Web programming refers to the coding involved in Web development, which usually includes different languages:

- **markup languages** (HTML, CSS): they define the structure, the organization, and the look and feel of a site.
- **client-side scripting languages** (javascript): they transform the website from a static page to an interactive application; client-side code runs in the browser.
- **server-side scripting languages** (PHP): server-side code lives on a server, and serves as go-between architecture, transferring data to the browser, minimizing the browser's workload, etc..

1.4 WHAT IS AN API

In the context of computer programming, an **Application Programming Interface** (API) represents a **set of procedures** for carrying out a given task.

An API lists a **bunch of operations** that developers can use, along with a description of what they do (the **API documentation**). The developer doesn't necessarily need to know how they work internally, they just need to know how to use them.

The term can also indicate a **software library** of a programming language.

1.4 WHAT IS AN API

APIs allow developers to **save time** by taking advantage of a platform's implementation to do the most important part of the work.

An API **reduces the amount of code developers need to create**, and also helps create more consistency apps for the same platform.

In last years, the specification and implementation of new APIs - such as **web audio** and webGL- in web browsers has allowed for envisioning the web platform as **a fertile play-ground for artists and musicians**.

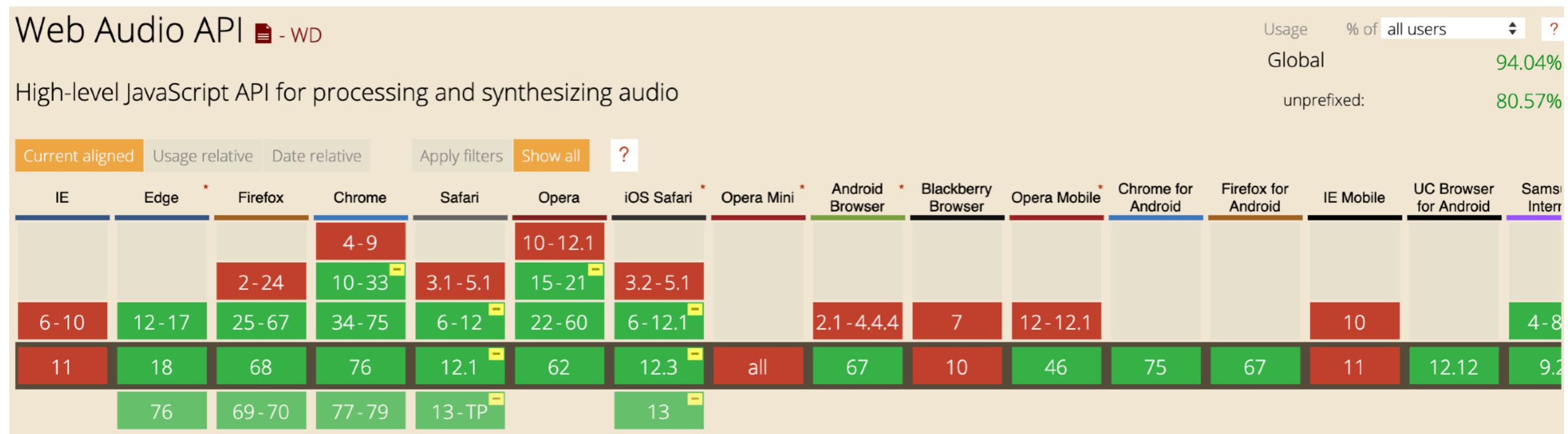
1.5 WHAT IS THE WEB AUDIO API

The **web audio API** is a **high level javascript API** developed by W3C **used for processing and synthesizing audio in web applications.**

The **web audio API** provides a powerful and versatile system for controlling audio on the Web, allowing developers to choose audio sources, add effects to audio, create audio visualization, apply spatial effects, etc.

[WEB AUDIO DOCUMENTATION: https://
developer.mozilla.org/en-US/docs/Web/API/
Web Audio API](https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API)

1.6 WEB AUDIO API BROWSER SUPPORT



1.7 BASIC CONCEPTS

The Web Audio API involves handling audio operations inside an **audio context**, and has been designed to allow **modular routing**.

Audio operations are performed with **audio nodes**, which are linked together into **chains** to form an **audio routing graph**. They typically start with one or more **sources** (digital oscillators or samples). Outputs of these nodes could be linked to inputs of others, which modify them.

Once the sound has been processed, it can be linked to the input of an **audio destination**, which sends the sound to the speakers or headphones.

1.7 BASIC CONCEPTS

THE WORKFLOW

1. **Create an audio context:** an AudioContext is a kind of container for AudioNode objects, which allow different kind of audio sources and processing.



1.7 BASIC CONCEPTS

THE WORKFLOW

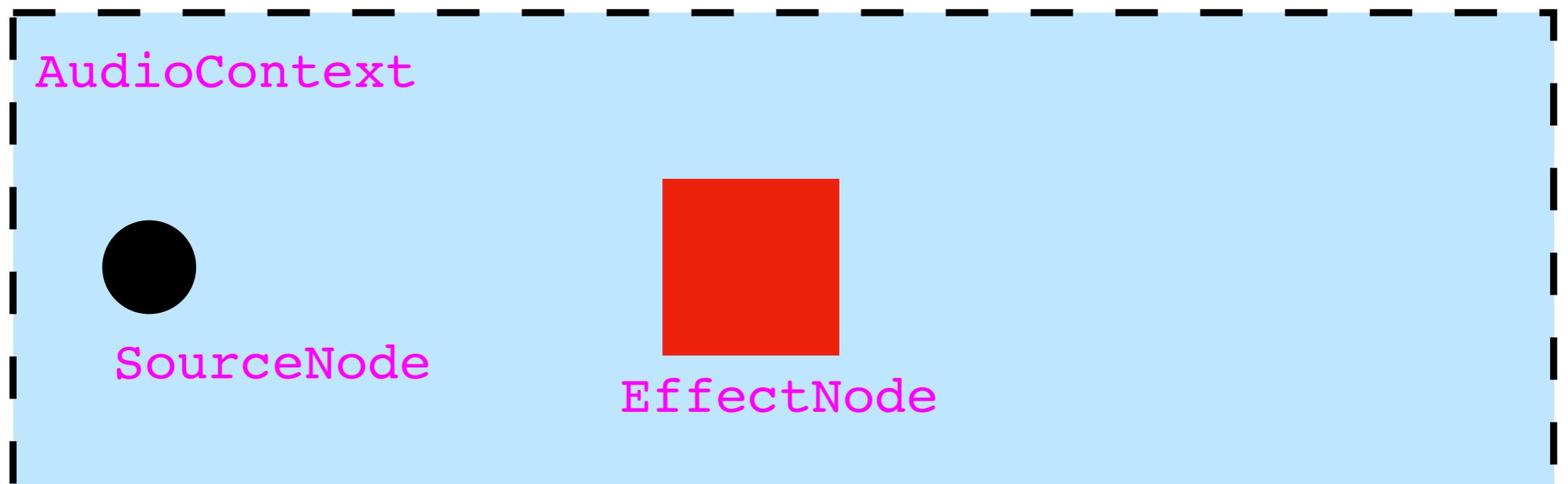
2. **Create audio sources inside the context:** it is possible to create different sound sources, called SourceNode (audio samples, or oscillators).



1.7 BASIC CONCEPTS

THE WORKFLOW

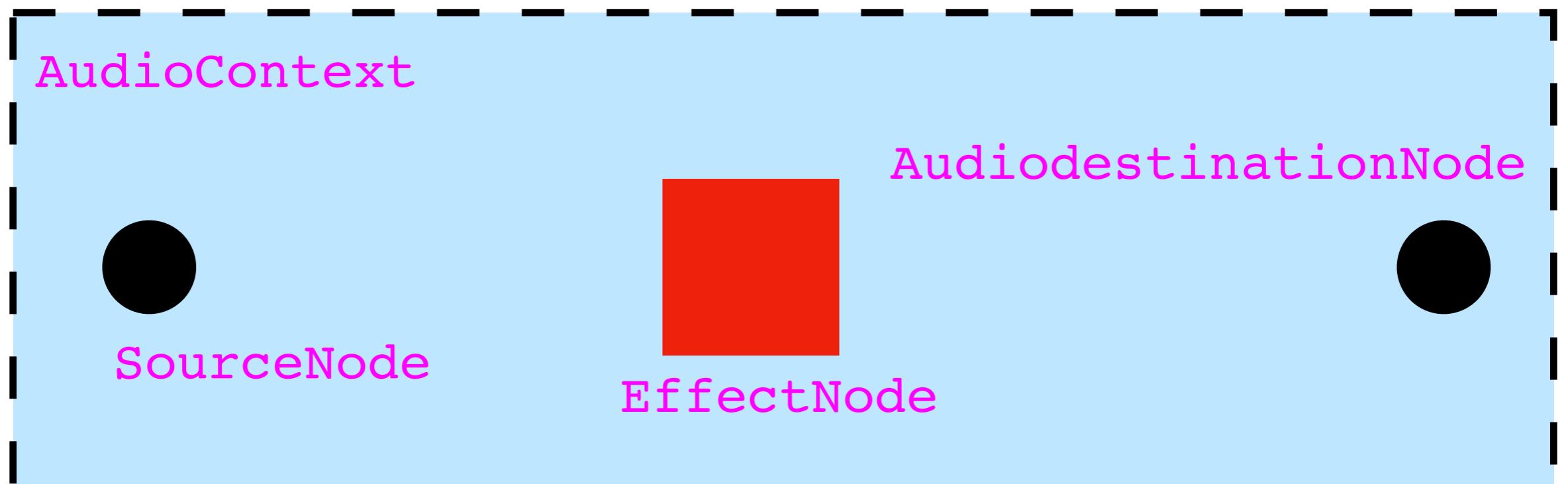
3. **Create effects nodes:** it is possible to apply different audio effects to the audio signal source (filter, delay, compressor, stereo panning, convolver, etc.).



1.7 BASIC CONCEPTS

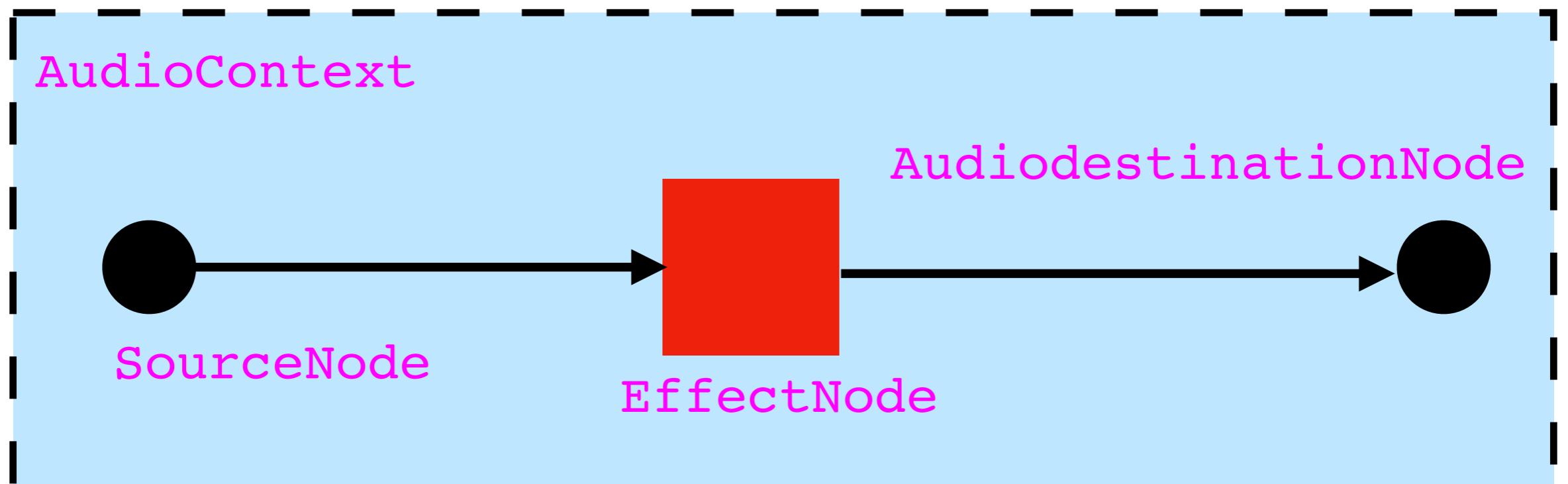
THE WORKFLOW

4. **Choose a final destination for the audio chain:** the `AudioDestinationNode` routes the sound inputs to a final audio destination, usually some kind of speaker system.

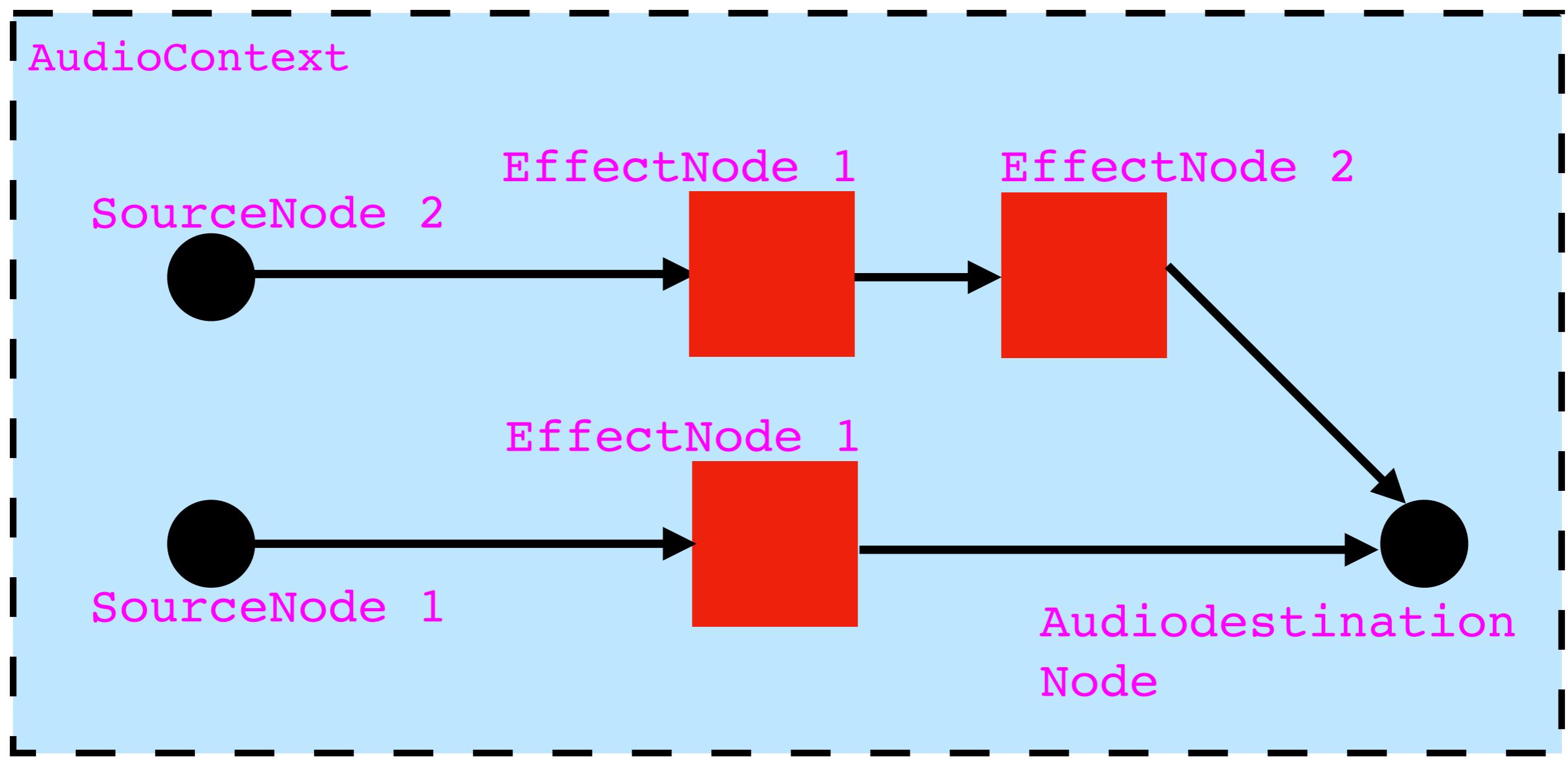


1.7 BASIC CONCEPTS THE WORKFLOW

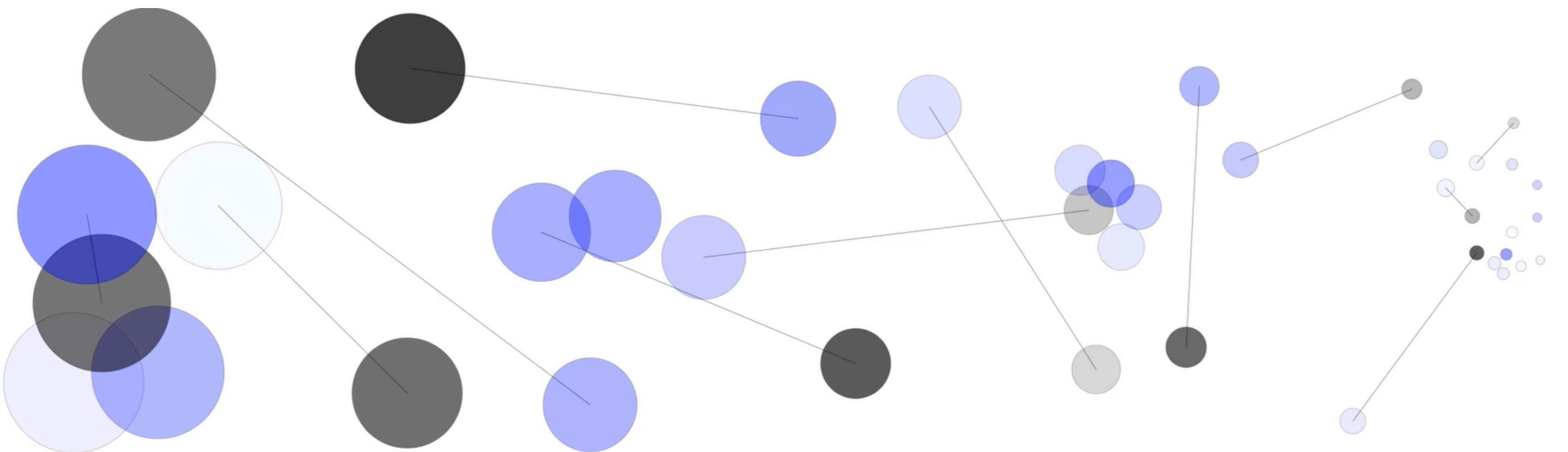
5. Connect the sources to effects and the effects to the destination



1.7 BASIC CONCEPTS THE WORKFLOW

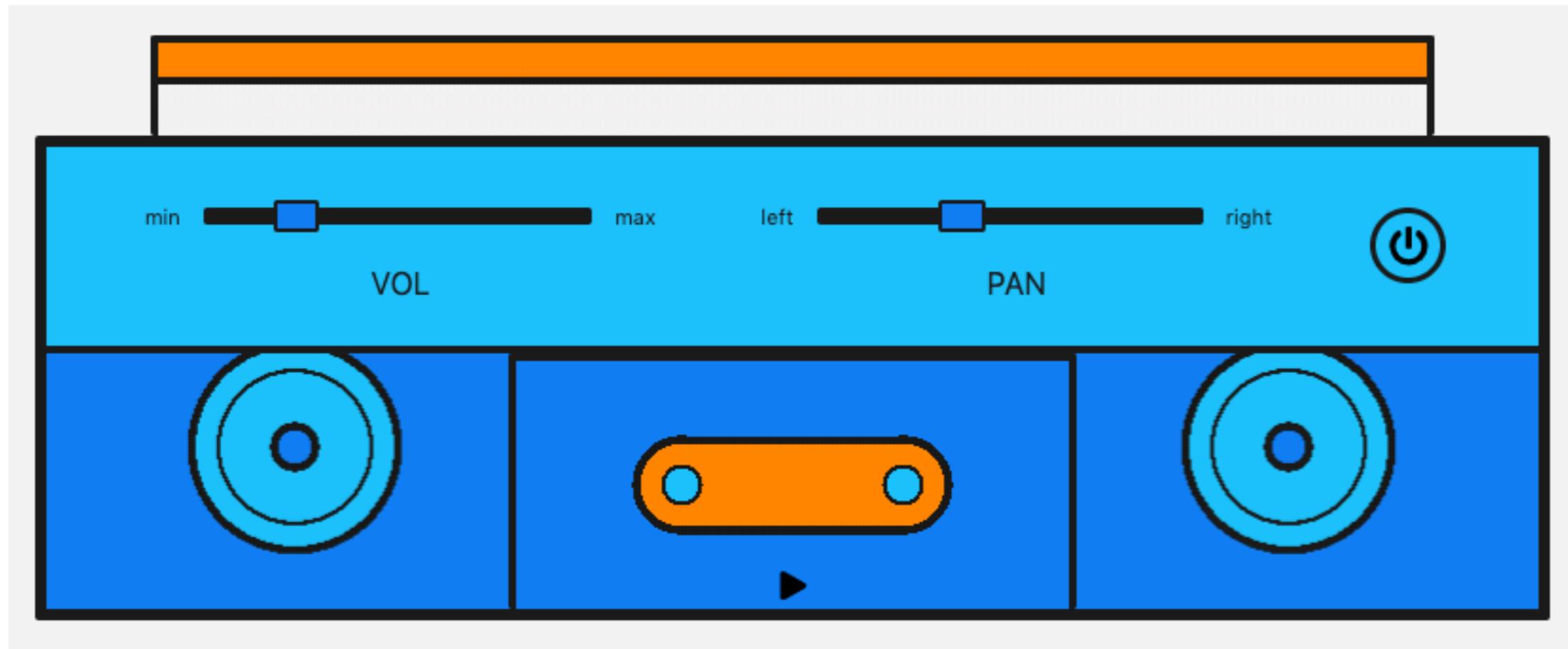


2. WEB AUDIO EXAMPLES



2.1 EXAMPLES

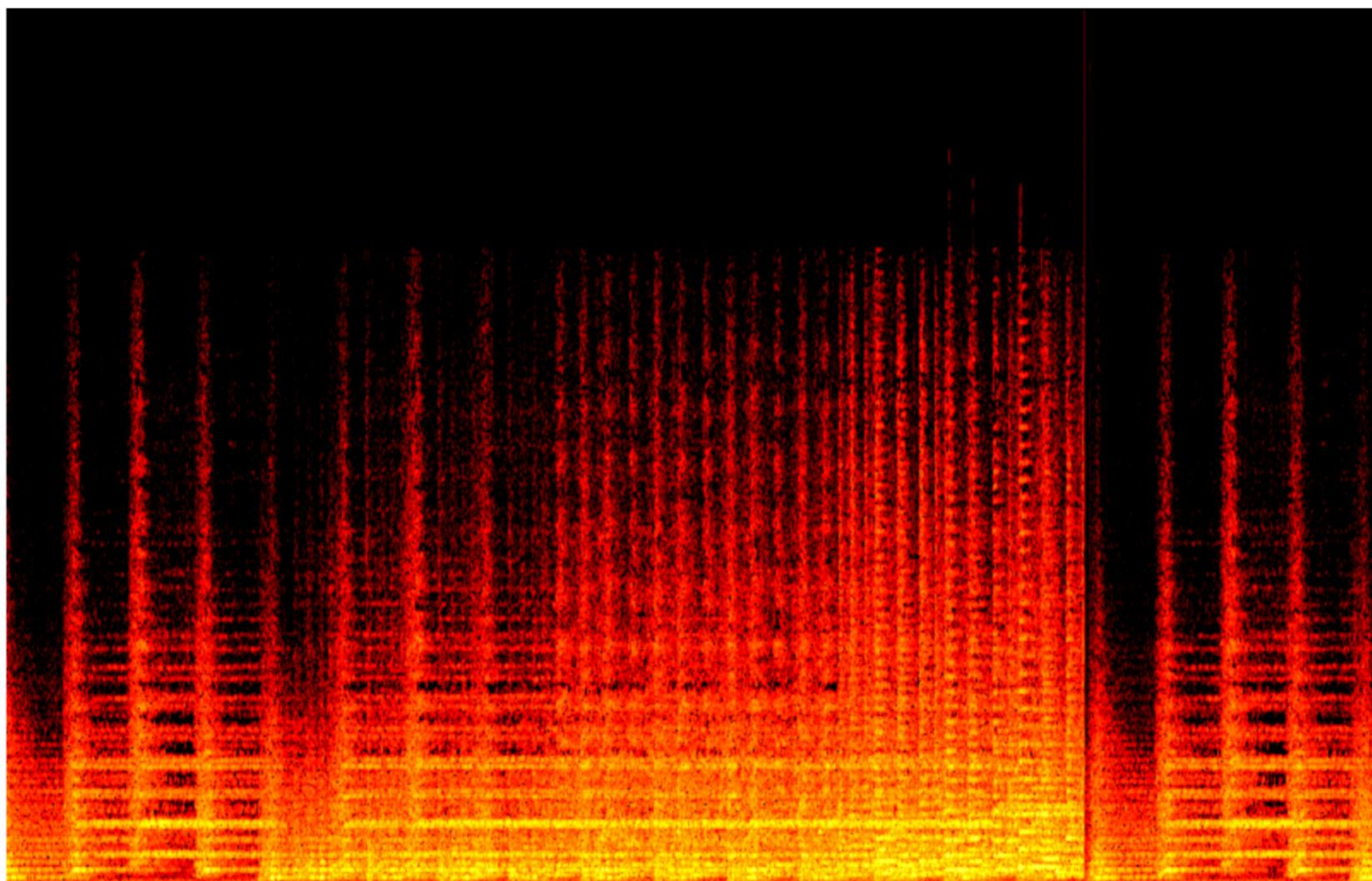
SIMPLE PLAYER



<https://codepen.io/Rumyra/pen/qyMzqN/>

2.2 EXAMPLES

AUDIO ANALYZER



<http://www.smartjava.org/examples/webaudio/example4.html>

2.3 EXAMPLES

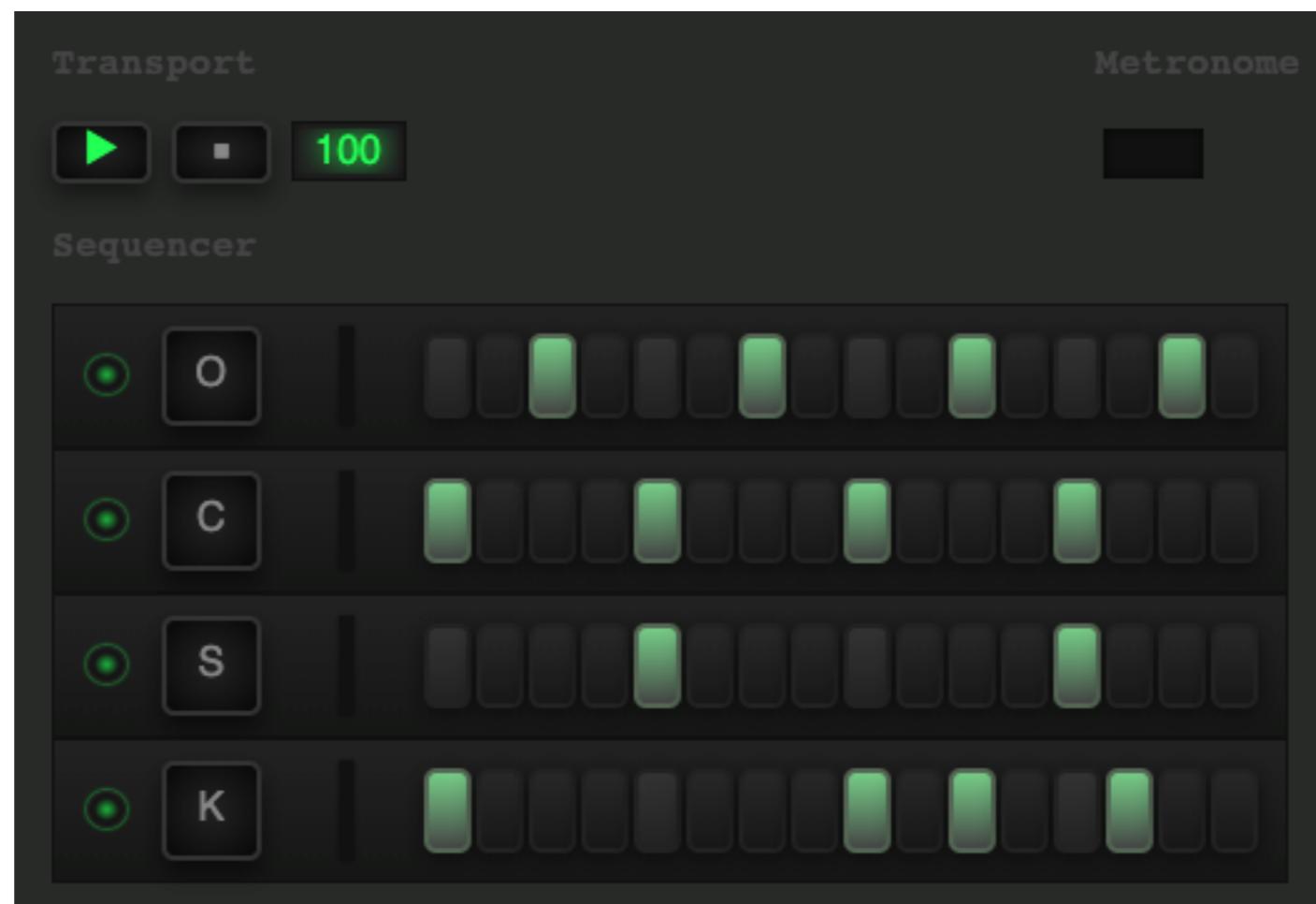
SUPERSAW SYNTH



<https://noisehack.com/scissor/>

2.4 EXAMPLES

STEP SEQUENCER



<https://codepen.io/njmcode/pen/PwaXwB>

2.5 EXAMPLES

GENERATIVE WEB COMPOSITION by Halldór Eldjárn

Poco Apollo is a generative music piece by Icelandic musician/programmer Halldór Eldjárn. It is built upon NASA's photo archive from the Apollo Space mission, which consists of ~15.000 photos taken during the missions. The piece is a web app that generates musical soundscapes for the images in the library. It uses common computer vision algorithms to determine the mood of the picture and that is used as the input to the music composition engine.

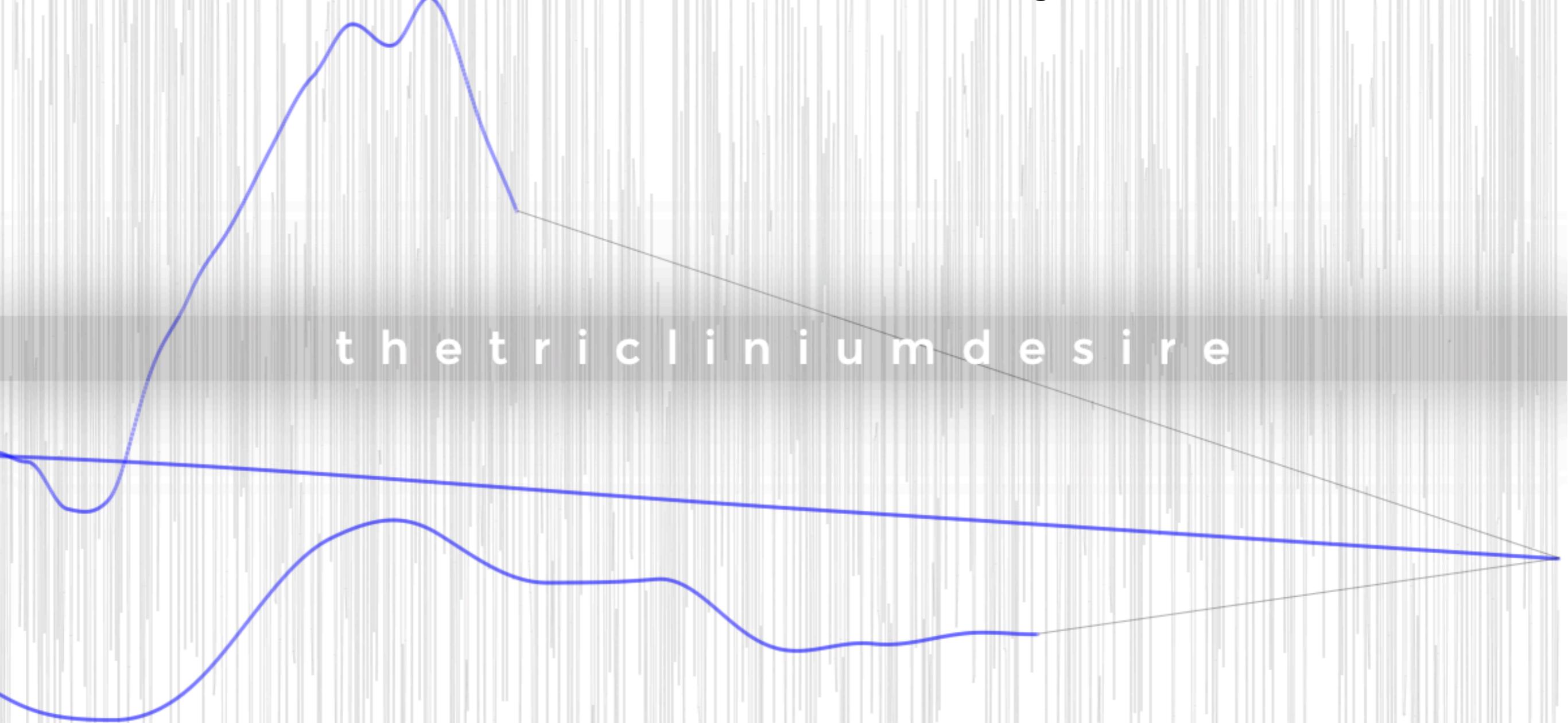
<https://hdor.is/project/generative/2018/03/29/poco-apollo.html>



https://pocoapollo.hdor.is/?song=21700411935_d9e863f759_k.jpg

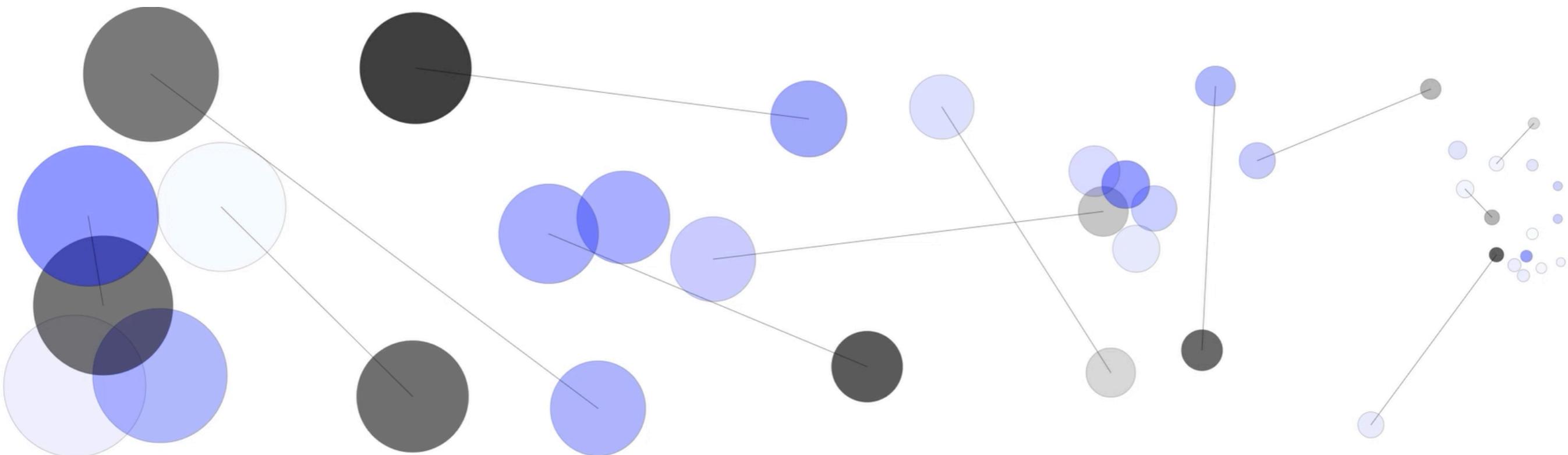
2.6 EXAMPLES

GENERATIVE WEB COMPOSITION by Alberto Barberis



<https://tamburo11.github.io/thetricliniumdesireWeb/>

3. POSSIBILITIES OFFERED BY WEB AUDIO



3.1 (NON-EXHAUSTIVE) LIST OF WEB AUDIO POSSIBILITIES

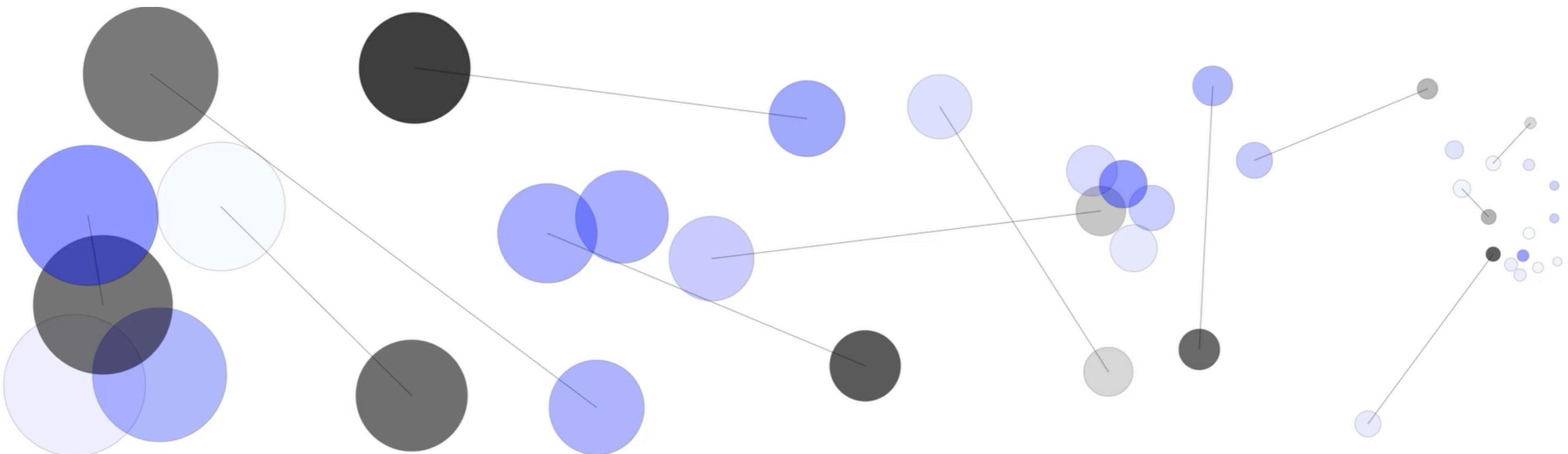
1. **play audio tracks on a web site** with a better control on panning, volumes, etc.
2. perform **audio analysis** and **audio signal visualization**.
3. create **educational web applications** (music theory, ear-training, etc.).
4. create **web digital instruments** (accessible from any device, shareable with anyone).

3.1 (NON-EXHAUSTIVE) LIST OF WEB AUDIO POSSIBILITIES

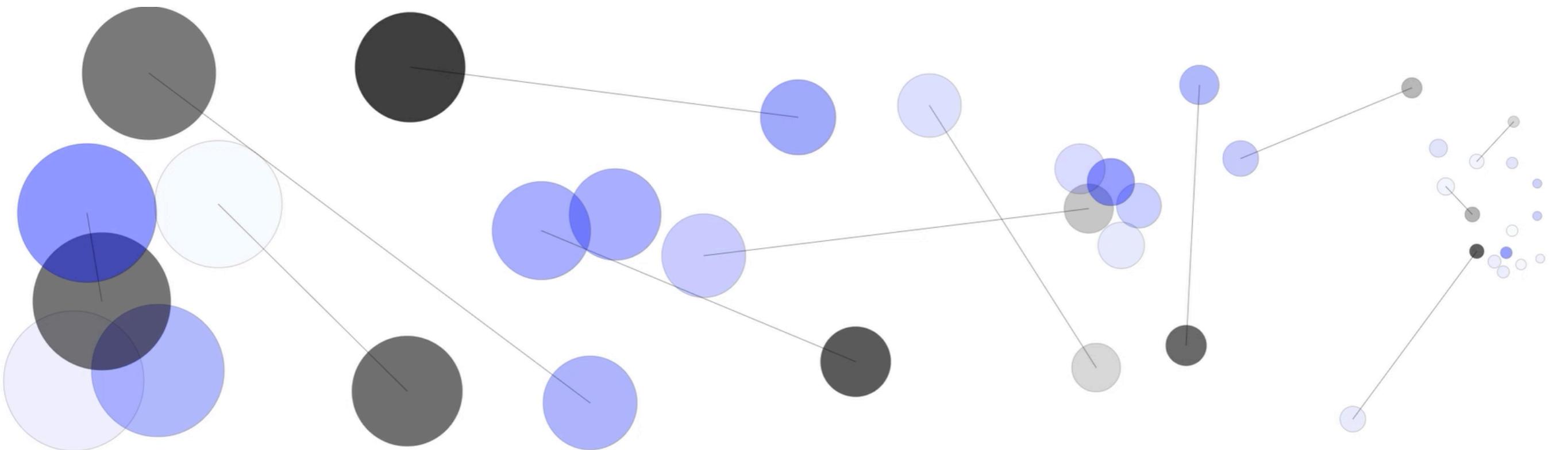
5. **create generative compositions** running on the browser (exploiting the algorithmic logic offered by the javascript programming language).
6. create **interactive compositions** (where users can play an active role in the artistic/creative process).
7. real-time **data analysis and data sonification** (using the open Data API available today on the web). [an example : <https://www.creativeapplications.net/processing/sorting-visualisation-sortification-and-sonification-of-an-algorithm/>]

3.1 (NON-EXHAUSTIVE) LIST OF WEB AUDIO POSSIBILITIES

8. create **not fixed media electronic music.**
9. create **web audio effects.**
10. etcetera.....



4. WHAT YOU NEED TO KNOW

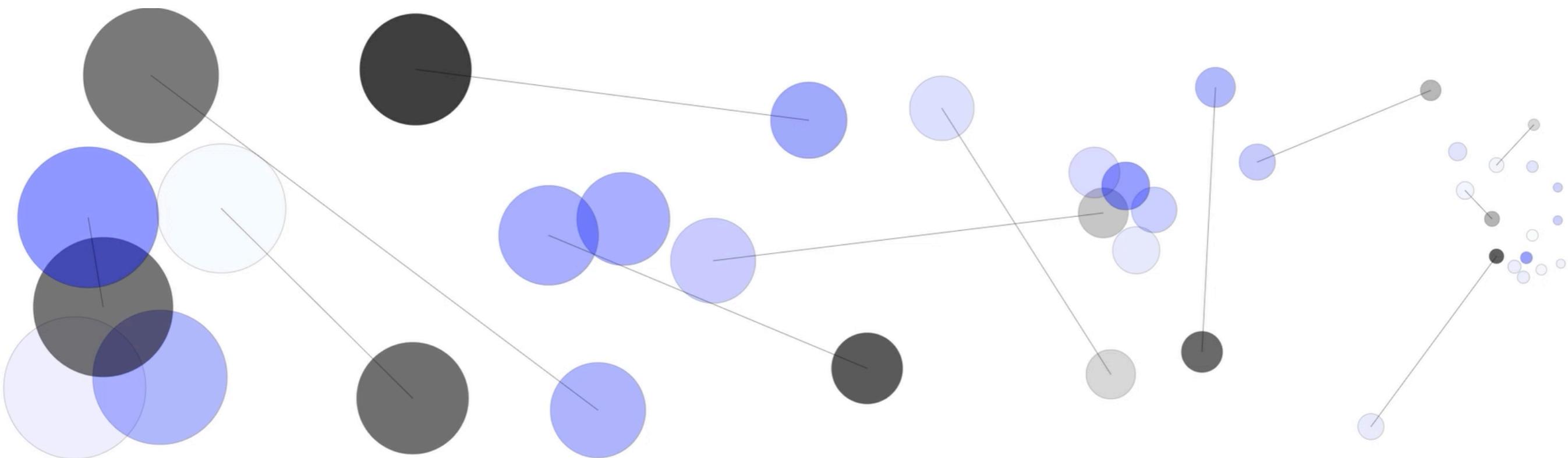


4.1 THE WEB AUDIO SKILLS

Dealing with **web application design** means fundamentally dealing with the following domains:

1. **web programming languages** (HTML, CSS, javascript, web audio API, Tone.js, p5.js, etc.);
2. **sound synthesis** methods (subtractive, additive, FM, AM, RM, wavetable, spectral, granular, physical modelling, analysis/resynthesis, etc.);
3. digital **signal processing** algorithms (compression, reverberation, delay, distortion, filtering, equalization, phasing, flanger, etc.).

5. DEVELOP A SIMPLE WEB AUDIO APP



an introduction to WEB audio API and p5.js sound library

The screenshot shows a web browser window titled "web audio HS2019". The address bar indicates the file is located at "/Users/albertobarberis/Desktop/WEB_AUDIO/hs2019_app/hs2019.html". The main content area has a blue header with the title "The highSCORE Dual Modulation Drone Generator". Below the header, a message says "highSCORE Festival, Pavia, August 2019". The main content area contains several paragraphs of text with callout lines pointing to specific features:

- "Dual Modulation Drone Generator is a polifonic drone generator which allows you to create synthetic textures directly on your browser."
- "It exploits both an AM (Amplitude Modulation) of sinusoidal waves by means of random triangular LFOs (Low Frequency Oscillators), and a double and reciprocal FM (Frequency Modulation) between pairs of oscillators."
- "Click on the window to create a sinusoidal wave (symbolized by a circle). The y-axis represents the maximum amplitude reached by the LFO. Each sinusoid is amplitude modulated by a random triangular LFO, with a frequency from a minimum of 0.001Hz to a maximum of 0.2Hz. The x-axis represents the frequencies, from 5Hz to 2000Hz."
- "Press 'L' and then Click+Drag from an existing circle to another one, to link two sinusoids and create a mutual double frequency modulation (symbolized by a line), where each sinusoid is treated as a carrier frequency modulated by a modulation Pulse (with random Pulse Width) at the same frequency of the other sinusoid of the pair."
- "Click on an existing circle to delete it or to delete the modulation pair of which it is part. And Enjoy."

5.1 HTML: BASIC STRUCTURE

HTML is the standard markup language for creating Web pages. With the HTML language you can create the **structure of the contents** of a web page.

- HTML stands for Hyper Text Markup Language;
- HTML describes the **structure** of a Web page;
- HTML consists of a series of **elements**;
- HTML elements tell the browser **how to display the content**;
- HTML elements are represented by **tags**;
- HTML tags label pieces of content such as "heading", "paragraph", and so on;
- Browsers use HTML tags to render the content of the page;

https://www.w3schools.com/html/html_intro.asp

5.1 HTML: BASIC STRUCTURE

```
<html>
```

```
  <head>
```

```
    <title>Page title</title>
```

```
  </head>
```

```
<body>
```

```
  <h1>This is a heading</h1>
```

```
  <p>This is a paragraph.</p>
```

```
  <p>This is another paragraph.</p>
```

```
</body>
```

```
</html>
```

basic structure of an .html page



let's create our first .html file!

5.1 HTML: BASIC STRUCTURE

```
<!DOCTYPE html> <!-- declaration defines this document to be HTML5 -->

<html> <!-- html tag is the root element of an HTML page -->

<head> <!-- head tag contains meta information about the document -->

</head>

<body> <!-- body tag contains the visible page content -->
    opening tag < name of the tag >
    <h1> <!-- h1 tag defines a large heading -->
        The highSCORE Dual Modulation Drone Generator
    </h1> closing tag </ name of the tag >

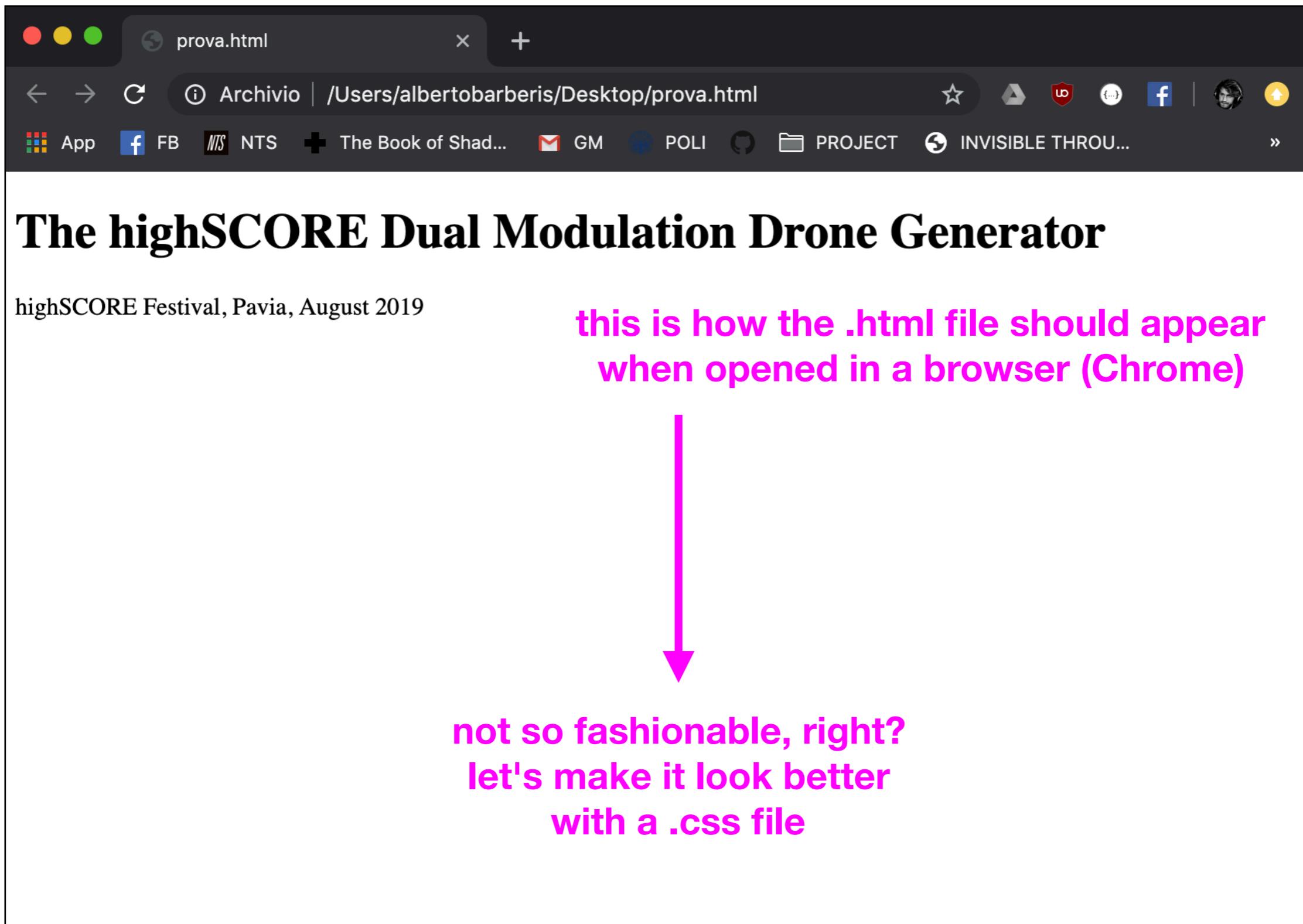
    <p> <!-- p tag defines a paragraph -->
        highSCORE Festival, Pavia, August 2019
    </p>

    <div id="infoText"> <!-- div tag defines a generic HTML elements -->
    </div>

</body>
</html>
```

this is an empty element that we are going to fill later dynamically with our javascript code (the language for the user-interaction!)

5.1 HTML: BASIC STRUCTURE



5.2 CSS: BASIC STYLING

CSS (Cascading Style Sheets) is a language that describes the **style of an HTML document** (describes how HTML elements should be displayed).

With the CSS language you can modify the **presentation of the elements** of an HTML code (including layout, colors, and fonts). CSS is designed to enable the separation of presentation (CSS) and content/structure (HTML).

CSS works applying some presentation rules to an HTML element. It is possible to apply different rules to the same element. The name **cascading** comes from the specified priority scheme to determine which style rule applies.

<https://www.w3schools.com/css/default.asp>

5.2 CSS: BASIC STYLING

THE BOX MODEL

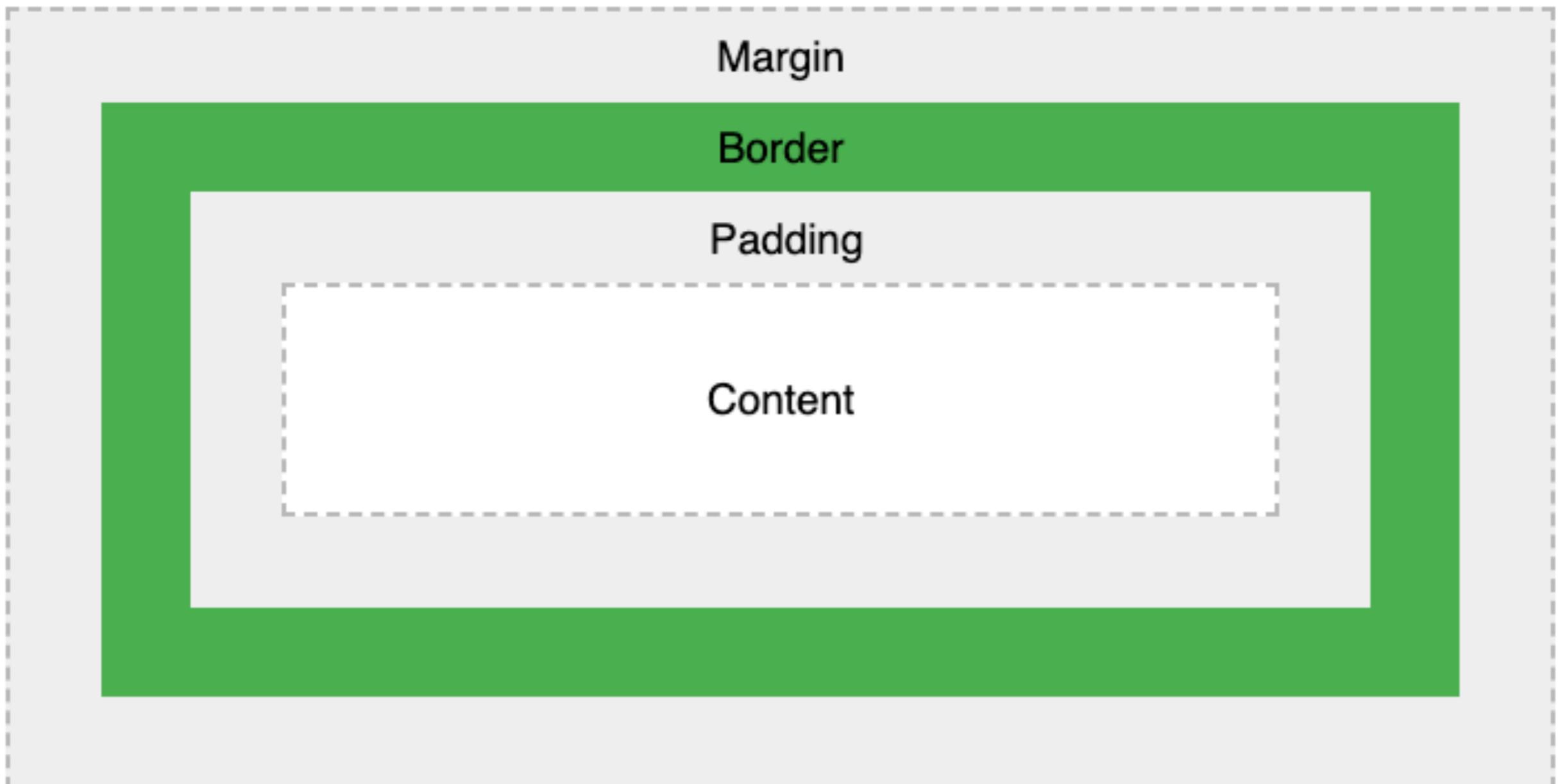
All HTML elements can be considered as boxes. The CSS box model is essentially a box that wraps around every HTML element. It consists of:

- **Content** - The content of the box, where text and images appear
- **Padding** - Clears an area around the content. The padding is transparent
- **Border** - A border that goes around the padding and content
- **Margin** - Clears an area outside the border. The margin is transparent

https://www.w3schools.com/css/css_boxmodel.asp

5.2 CSS: BASIC STYLING

THE BOX MODEL



https://www.w3schools.com/css/css_boxmodel.asp

an introduction to WEB audio API and p5.js sound library

```
<!DOCTYPE html> <!-- declaration defines this document to be HTML5 -->

<html> <!-- html tag is the root element of an HTML page -->

<head> <!-- head tag contains meta information about the document -->
    each tag can have different attributes
        <!-- include css file -->
        <link rel="stylesheet" type="text/css" href="hs2019_style.css">
</head>

<body> <!-- body tag defines the main content -->
    <h1> <!-- h1 tag defines a heading -->
        The
    </h1>
    <p> <!-- p tag defines a paragraph -->
        highSCORE Festival, Pavia, August 2019
    </p>
    <div id="infoText"> <!-- div tag defines a generic HTML elements -->
    </div>
</body>
</html>
```

rel attribute specifies the relationship between the documents

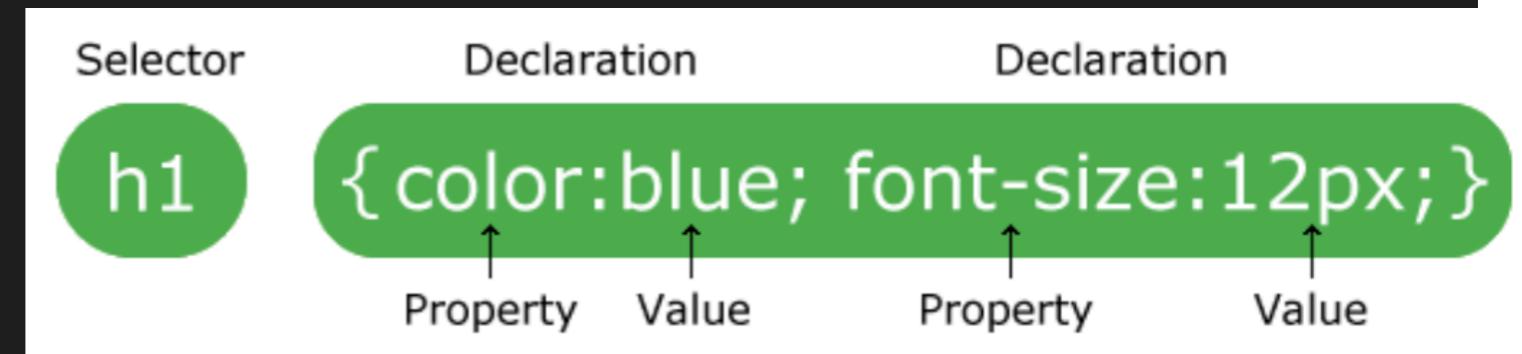
type specifies the type of document

href sets the file name

let's create our first .css file and link it to the .html file

5.2 CSS: BASIC STYLING

```
body {  
    margin: 0px;  
    font-family: 'Courier New', Courier, monospace;  
}  
  
h1 {  
    margin: 0px;  
    padding: 10px;  
    background-color: rgb(88, 88, 255);  
    color: white;  
    -webkit-user-select: none;  
}  
  
p {  
    margin: 0px;  
    padding: 10px;  
    color: rgb(0, 0, 0);  
    border-bottom: solid;  
    border-width: 1px;  
    -webkit-user-select: none;  
}
```

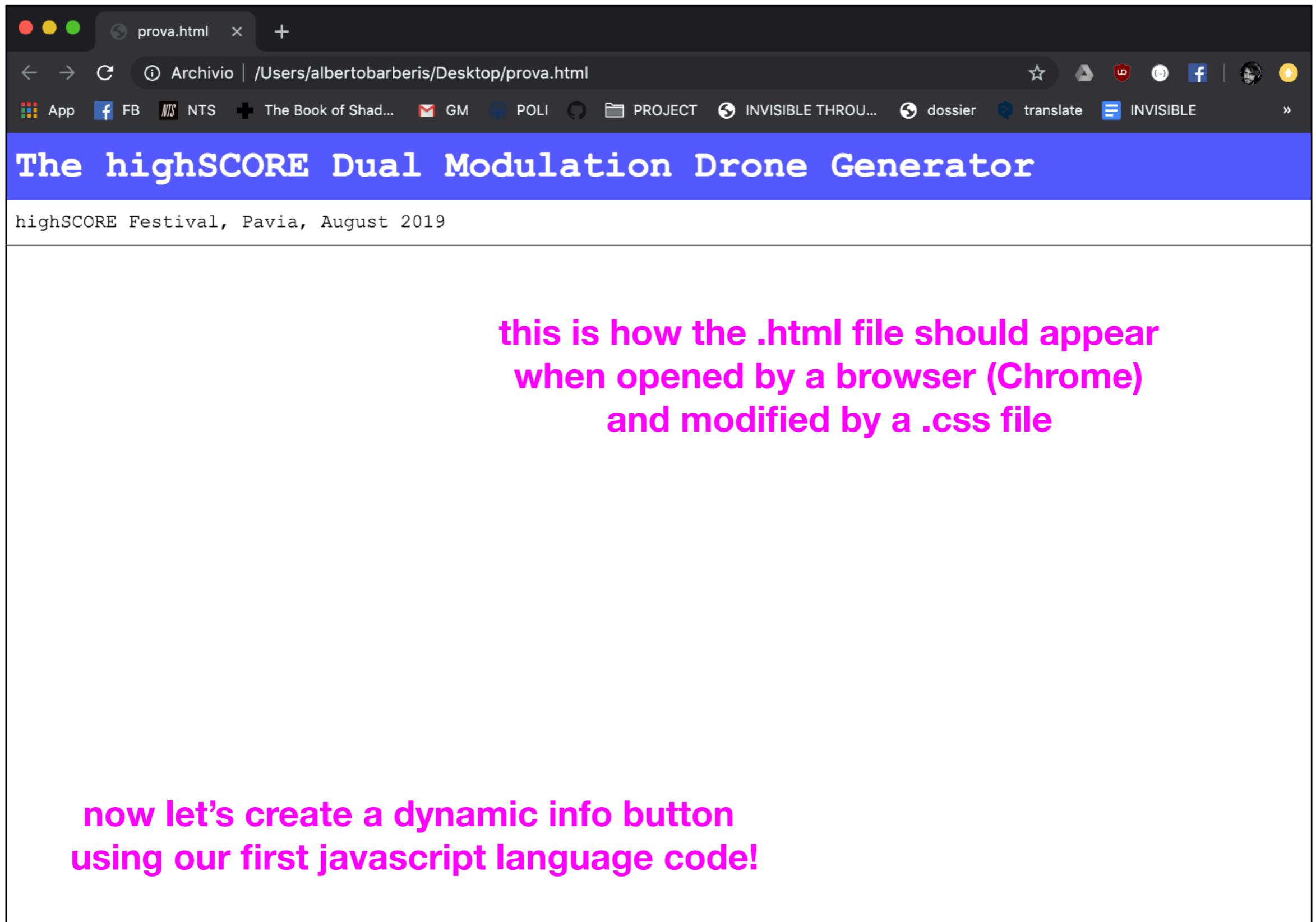


A CSS rule-set consists of a selector and a declaration block. The selector points to the HTML element you want to style.

The declaration block contains one or more declarations separated by semicolons. Each declaration includes a CSS property name and a value, separated by a colon.

Declaration blocks are surrounded by curly braces.

5.2 CSS: BASIC STYLING



5.3 USER INTERACTION : create a button

```
<!-- WEB AUDIO highSCORE 2019-->

<!DOCTYPE html> <!-- declaration defines this document to be HTML5 -->

<html> <!-- html tag is the root element of an HTML page -->

<head> <!-- head tag contains meta information about the document -->
    <!-- include css file -->
    <link rel="stylesheet" type="text/css" href="hs2019_style.css">

</head>

<body> <!-- body tag contains the visible page content -->

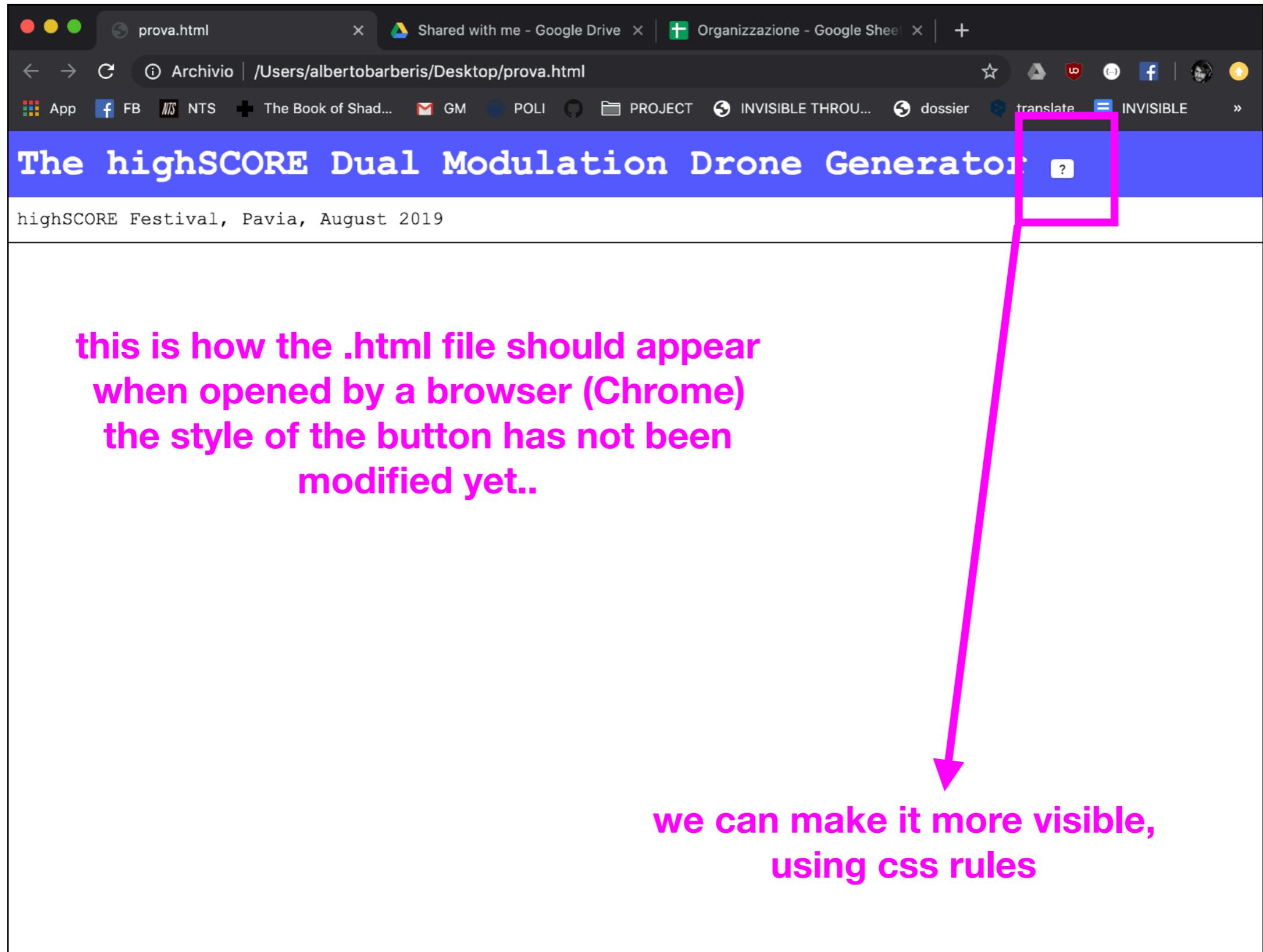
    <h1> <!-- h1 tag defines a large heading -->
        The highSCORE Dual Modulation Drone Generator
        <button type="button" id="infoButton" onclick="showInfo()"> ?</button>
    </h1>

    <p> <!-- p tag defines a defines a paragraph -->
        highSCORE Festival, Pavia, August 2019
    </p>

    <div id="infoTe<b>id specifies a unique id address (that we will use to refer to</b>
        <b>this element in the css file and later in the javascript file</b>
    </div>

</body>
</html>
```

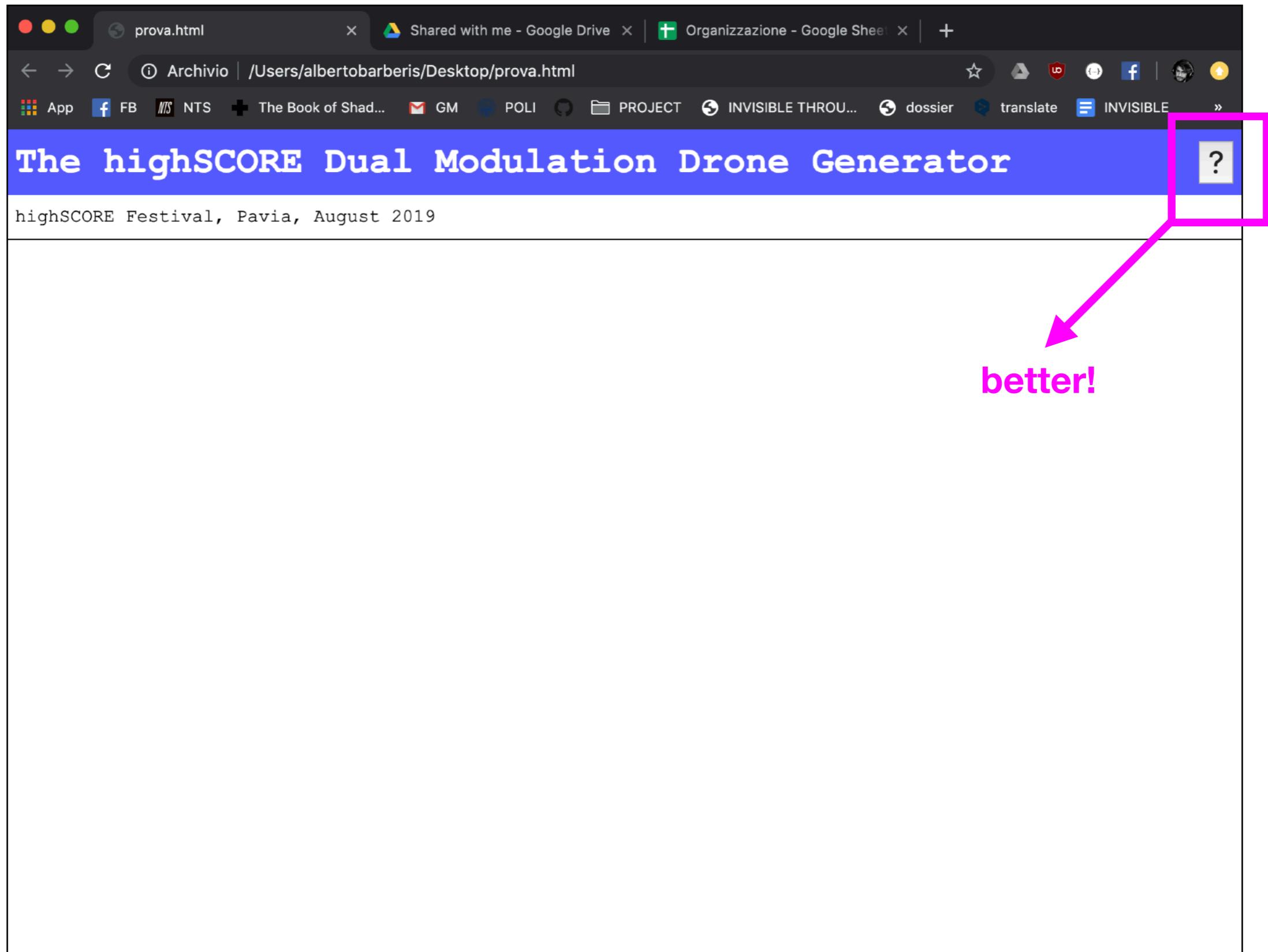
5.3 USER INTERACTION : create a button



5.3 USER INTERACTION : create a button

```
body {  
    margin: 0px;  
    font-family: 'Courier New', Courier, monospace;  
}  
  
h1 {  
    margin: 0px;  
    padding: 10px;  
    background-color: rgb(88, 88, 255);  
    color: white;  
    -webkit-user-select: none;  
}  
  
p {  
    margin: 0px;  
    padding: 10px;  
    color: rgb(0, 0, 0);  
    border-bottom: solid;  
    border-width: 1px;  
    -webkit-user-select: none;  
}  
  
#infoButton {  
    float: right;  
    font-size: 27px;  
}
```

5.3 USER INTERACTION : create a button



5.4 USER INTERACTION : using JAVASCRIPT

```
#infoButton {  
    float: right;  
    font-size: 27px;  
}
```

```
#infoText {  
    position: fixed;  
    margin-top: 50px;  
    margin-left: 100px;  
    margin-right: 100px;  
    float: center;  
    text-align: justify;  
    line-height: 150%;  
    -webkit-user-select: none;  
}
```

now we set some more rules in the .css file for the empty div that we created in the .html file

we want it to appear in the center of the window with an info text



we want that the info text appears when the user clicks on the button we just created

5.4 USER INTERACTION : using JAVASCRIPT

Javascript is a **high-level interpreted scripting language**.

It is the programming language used to modify dynamically the content of a web page and for the user-interaction.

Javascript characteristics are: curly-bracket syntax, dynamic typing, object-orientation.

The **web audio API is an API for javascript**. All the features it offers apply to this language.

It is possible to link an external .js file to an HTML file using the tag:

```
<script src="nameOfJavascriptFile.js"></script>
```

<https://www.w3schools.com/js/default.asp>

an introduction to WEB audio API and p5.js sound library

```
<!-- WEB AUDIO highSCORE 2019-->

<!DOCTYPE html> <!-- declaration defines this document to be HTML5 -->

<html> <!-- html tag is the root element of an HTML page -->

<head> <!-- head tag contains meta information about the document -->
    <!-- include css file -->
    <link rel="stylesheet" type="text/css" href="hs2019_style.css">

    <script src="hs2019_javascript_workshop.js"></script>

</head>      the tag script is used to link a javascript file to the .html one

<body> <!-- body tag contains the visible page content -->

    <h1> <!-- h1 tag defines a large heading -->
        The highSCORE Dual Modulation Drone Generator
        <button type="button" id="infoButton" onclick="showInfo()"> ?</button>
    </h1>

    <p> <!-- p tag defines a defines a paragraph -->
        highSCORE Festival, Pavia, August 2019
    </p>

    <div id="infoText">
    </div>

</body>

</html>
```

so, let's create a javascript file!

5.4 USER INTERACTION : using JAVASCRIPT

```
// GLOBAL VARIABLES  
var questionmark = false;  
  
// SHOW INFO FUNCTION  
function showInfo(){  
    if(questionmark==false){  
  
        document.getElementById("infoText").innerHTML =  
  
"Dual Modulation Drone Generator :  
textures directly on your browser.  
means of random triangular LF0s, a  
oscillators. Click on the window  
the frequencies, from 5Hz to 2000Hz.  
sinusoid is amplitude modulated by  
maximum of 0.2Hz. Press 'L' and the  
sinusoids and create a mutual doubl  
treated as a carrier frequency modulated by a  
frequency of the other sinusoid of the pair. Click on an existing circle to delete it or to delete the  
modulation pair of which it is part. And Enjoy."  
        questionmark=true;  
  
    } else {  
  
        document.getElementById("infoText").innerHTML = "";  
        questionmark=false;  
  
    }  
}
```

WHAT IS A VARIABLE?

with the keyword **var** we declare a variable, that is a sort of container which holds the data value, that can be changed anytime. In this case we use a boolean type, that can be **false** or **true**.

5.4 USER INTERACTION : using JAVASCRIPT

```
// GLOBAL VARIABLES  
var questionmark = false;  
  
// SHOW INFO FUNCTION  
function showInfo(){  
    if(questionmark==false){  
        document.getElementById("infoText").innerHTML =  
        "Dual Modulation  
textures do not mean  
means of random oscillators  
the frequency of a  
sinusoid is treated as  
maximum of two  
sinusoids and is  
treated as a frequency  
modulation";  
        questionmark=true;  
    } else {  
        document.getElementById("infoText").innerHTML = "";  
        questionmark=false;  
    }  
}
```

WHAT IS A FUNCTION?

with the key word `function` we declare a function, that is a block of code designed to perform a particular task. To invoke the function we use its name, that is `showInfo()`.

The block of code is inserted between the two curly brackets

```
function showInfo() {  
    ...  
}
```

synthetic
l waves by
n pairs of
axis represents
he LFO. Each
f 0.001Hz to a
ink two
sinusoid is
he same
elete the

5.4 USER INTERACTION : using JAVASCRIPT

```
// GLOBAL VARIABLES
var questionmark = false;

// SHOW INFO FUNCTION
function showInfo(){
    if(questionmark==false){
        document.getElementById("infoText").innerHTML =
        "Dual Modulation Drone Generator is a polyphonic drone generator which allows you to create synthetic textures directly on your browser. It exploits means of random triangular LFOs, and a double a oscillators. Click on the window to create a si ...  
the frequencies, from 5Hz to 2000Hz. The y-axis sinusoid is amplitude modulated by a random tri maximum of 0.2Hz. Press 'L' and then Click+Drag sinusoids and create a mutual double frequency treated as a carrier frequency modulated by a m frequency of the other sinusoid of the pair. Cl modulation pair of which it is part. And Enjoy.  
        questionmark=true;
    } else {
        document.getElementById("infoText").innerHTML = "";
        questionmark=false;
    }
}
```

WHAT IS A CONDITIONAL STATEMENT?

```
if () {  
...  
} else {  
...  
}  
}
```

is a conditional statement, used to perform different actions based on different conditions

l waves by n pairs of axis represents he LFO. Each f 0.001Hz to a ink two sinusoid is he same elete the

5.4 USER INTERACTION : using JAVASCRIPT

```
// GLOBAL VARIABLES  
var questionmark = false;  
  
// SHOW INFO FUNCTION  
function showInfo(){  
    if(questionmark==false){  
  
        document.getElementById("infoText").innerHTML =  
  
    "Dual Modulation Drone Generator is a polyphonic drone generator which allows you to create synthetic  
textures directly on your browser. It exploits both an AM (amplitude modulation) of sinusoidal waves by  
means of random triangular LFOs, and a double and reciprocal FM (frequency modulation) between pairs of  
oscillators. Click on the window to create a sinusoidal wave (symbolized by a circle). The x-axis represents  
the frequencies, from 5Hz to 2000Hz. The y-axis represents the maximum amplitude reached by the LFO. Each  
sinusoid is amplitude modulated by a random triangular LFO, with a frequency from a minimum of 0.001Hz to a  
maximum of 0.2Hz. Press 'L' and then Click+Drag from an existing circle to an other one, to link two  
sinusoids and create a mutual double frequency modulation. The carrier frequency of one sinusoid is treated as a carrier frequency modulated by the LFO of the other sinusoid, and the frequency of the other sinusoid of the pair is treated as a carrier frequency modulated by the LFO of the first sinusoid. The frequency of the LFO of each sinusoid is modulated by a random triangular LFO, with a frequency from a minimum of 0.001Hz to a maximum of 0.2Hz. Press 'L' and then Click+Drag from an existing circle to an other one, to link two  
sinusoids and create a mutual double frequency modulation. The carrier frequency of one sinusoid is treated as a carrier frequency modulated by the LFO of the other sinusoid, and the frequency of the other sinusoid of the pair is treated as a carrier frequency modulated by the LFO of the first sinusoid. And E  
        questionmark=true;  
    } else {  
        document.getElementById("infoText").innerHTML = "";  
    }  
}
```

WHAT IS A METHOD?

if the variable **questionmark** is false we get the html element with the id **infoText**, using the method **getElementById**, and then we change its HTML content with a string of text (our info text).

a javascript method is an action that can be performed on javascript objects (in this case the object that represents the html document itself)

5.4 USER INTERACTION : using JAVASCRIPT

The screenshot shows a web browser window titled "web audio x". The address bar displays the URL: "/Users/albertobarberis/Desktop/WEB_AUDIO/hs2019_app/hs2019.html". The page content is titled "The highSCORE Dual Modulation Drone Generator" and includes the text "highSCORE Festival, Pavia, August 2019". Below this, there is descriptive text about the application's functionality, including AM and FM modulation, and instructions for creating and linking sinusoidal waves using mouse clicks. A callout box with a pink border and text is overlaid on the bottom right of the page content.

Dual Modulation Drone Generator is a polifonic drone generator which allows you to create synthetic textures directly on your browser.

It exploits both an AM (Amplitude Modulation) of sinusoidal waves by means of random triangular LFOs (Low Frequency Oscillators), and a double and reciprocal FM (Frequency Modulation) between pairs of oscillators.

Click on the window to create a sinusoidal wave (symbolized by a circle). The y-axis represents the maximum amplitude reached by the LFO. Each sinusoid is amplitude modulated by a random triangular LFO, with a frequency from a minimum of 0.001Hz to a maximum of 0.2Hz. The x-axis represents the frequencies, from 5Hz to 2000Hz.

Press 'L' and then Click+Drag from an existing circle to an other one, to link two sinusoids and create a mutual double frequency modulation (symbolized by a line), where each sinusoid is treated as a carrier frequency modulated by a modulation Pulse (with random Pulse Width) at the same frequency of the other sinusoid of the pair.

Click on an existing circle to delete it or to And Enjoy.

this is how the .html file should appear when the user click on the question mark button

5.5 p5.js JAVASCRIPT LIBRARY

«*p5.js is a javascript library that starts with the original goal of Processing, to **make coding accessible for artists, designers, educators, and reinterprets this for today's web.***

p5.sound library extends p5 with Web Audio functionality including audio input, playback, analysis and synthesis.»

p5.js sound library objects and methods **simplify a lot the web audio API workflow** described before.

<https://p5js.org/reference/#/libraries/p5.sound>

5.5 p5.js JAVASCRIPT LIBRARY

```
<head> <!-- head tag contains meta information about the document -->  
<!-- include css file -->  
<link rel="stylesheet" type="text/css" href="hs2019_style_workshop.css">
```

```
<!-- include JS file --> https://cdnjs.com/libraries/p5.js  
<script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.9.0/p5.js"></script>  
<script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.9.0/addons/p5.sound.js"></script>  
<script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.6.1/addons/p5.dom.js"></script>
```

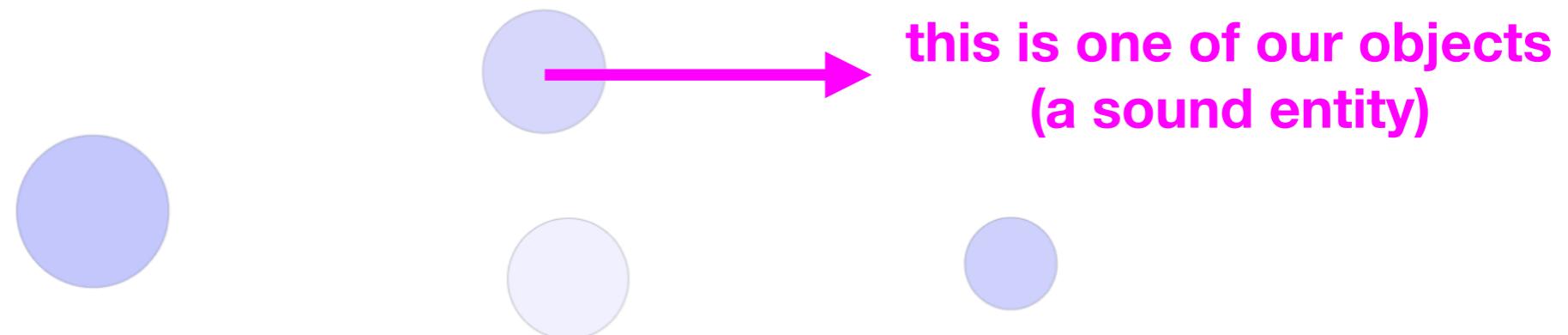
```
<script src="hs2019_javascript_workshop.js"></script>  
<script src=".p5.min.js"> </script>  
<script src=".p5.dom.min.js"> </script>  
<script src=".p5.sound.min.js"> </script>
```

```
<script src="hs2019_javascript_workshop.js"></script>  
</head>
```

We include in the html file
the p5.js libraries that we need.
In this case the libraries that we
have downloaded.

Or you can link to a server
where the libraries are stored
(but you need a connection)

5.5 OBJECTS



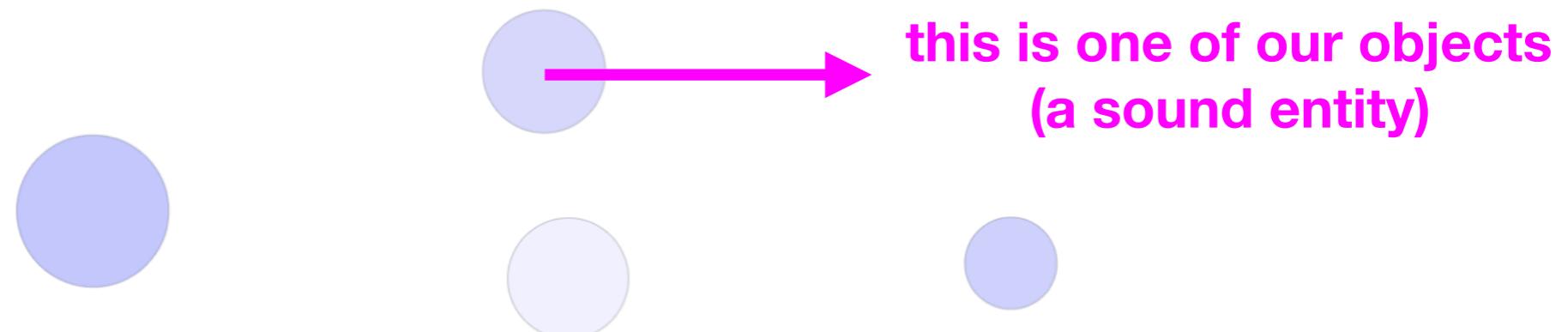
The object-oriented languages are based on **objects**. An objects is an **entity** with properties and methods.

For example a car is an **object** with some **properties** (like color and weight) and some **methods** (like start or stop).

In our situation we want to create a **sound entity** called Dot. So we create a Dot object with some properties.

And then we want to create one of this object each time we click on the window of the browser (to create a new object we use the keyword **new**).

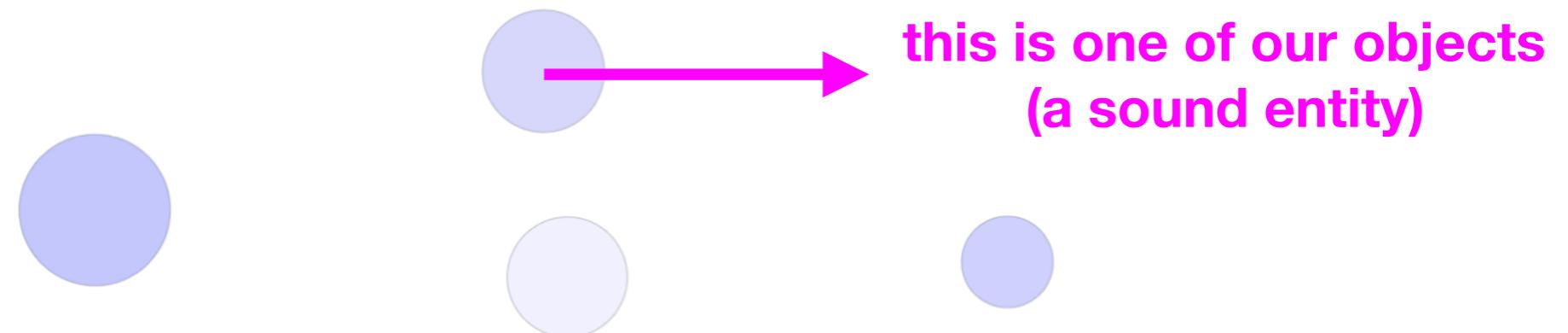
5.5 OBJECTS



A **Dot object** is a sound entity with the following properties:

- a **position** (X-Y) on the browser window;
- an associated **frequency** (defined by the position of the point on the X axis);
- a **diameter** (related to the frequency: lower the frequency, bigger the diameter);
- an associated **sine Oscillator** object;
- an associated **amplitude** (defined by the position of the point on the Y axis);
- an associated **triangle LFO** (Low Frequency Oscillator) object, for the Amplitude Modulation;

5.5 OBJECTS



You can access the properties of an object by specifying the name of the object, followed by a dot (.) followed by the property name. This is the syntax:

objectName.propertyName

5.5 OBJECTS

LET'S CREATE THE DOT CONSTRUCTOR

The Dot constructor is a function invoked by the `new` keyword that creates an object of the type Dot. Inside it you define all the properties you want to define.

```
// DOT CONSTRUCTOR
function Dot(x_axis,y_axis){ →
    // position of the Dot
    this.x=x_axis;
    this.y=y_axis;
}
```

we assign to the property `x` of the created object the `x_axis` value

name of the property (in this case `x`)

When we invoke the constructor we use two parameters, `x_axis` and `y_axis`, that indicate the position of the object in the browser window

this is a keyword that refers to the new created object.

5.5 OBJECTS

```
// CLASS DOT
function Dot(x_axis,y_axis){
    // position of the Dot
    this.x=x_axis;
    this.y=y_axis;

    // define amplitude and frequency of the associated sinusoid
    this.freq=map(this.x,0>windowWidth,5,2000);
    this.amp=map(this.y>windowHeight-97,0,0.0001,0.1);

    // set the dimension of the ellipse
    this.dimension=map(this.freq,2000,5,3,100);
}
```

map is a function of the p5.js library used for mapping. We take the x position of the current object and we scale it from 5 to 2000, that represent Hz.

here we use the frequency to define the dimension of the diameter of the dot

5.5 OBJECTS

```
// CLASS DOT
function Dot(x_axis,y_axis){
    // position of the Dot
    this.x=x_axis;
    this.y=y_axis;

    // define amplitude and frequency of the associated sinusoid
    this.freq=map(this.x,0>windowWidth,5,2000);
    this.amp=map(this.y>windowHeight-97,0,0.0001,0.1);

    // set the dimension of the ellipse
    this.dimension=map(this.freq,2000,5,3,100);

    // oscillator associated to Dot
    this.osc = new p5.Oscillator();
    this.osc.setType('sine'); →
    this.osc.freq(this.freq);
    this.osc.amp(0);
    this.osc.start(); →
}
```

new invokes the constructor of the Oscillator object (that is defined in the p5.js library) and we create the property osc of our general object Dot.

here we use different methods of the object Oscillator do set some properties of it (like the type, the frequency, the amplitude).

the method start() activates the Oscillator.

5.5 OBJECTS

```
// CLASS DOT
function Dot(x_axis,y_axis){

    // position of the Dot
    this.x=x_axis;
    this.y=y_axis;

    // define amplitude and frequency of the associated sinusoid
    this.freq=map(this.x,0>windowWidth,5,2000);
    this.amp=map(this.y>windowHeight-97,0,0.0001,0.1);

    // set the dimension of the ellipse
    this.dimension=map(this.freq,2000,5,3,100);

    // oscillator associated to Dot
    this.osc = new p5.Oscillator();
    this.osc.setType('sine');
    this.osc.freq(this.freq);
    this.osc.amp(0);
    this.osc.start();

    // LFO triangle for Amplitude Modulation
    this.ampModulator = new p5.Oscillator('triangle');
    this.ampModulator.disconnect(); // disconnect the ampModulator from master output
    this.ampModulator.freq(random(0.2)+0.001);
    this.ampModulator.amp(1);
    this.ampModulator.start();
}
```

We create an other Oscillator object associated to a Dot (LFO). This is going to be used for Amplitude Modulation.

the random method provides a random number between 0 and the number used as parameter of the function.

```
// CLASS DOT
function Dot(x_axis,y_axis){
    // position of the Dot
    this.x=x_axis;
    this.y=y_axis;

    // define amplitude and frequency of the associated sinusoid
    this.freq=map(this.x,0>windowWidth,5,2000);
    this.amp=map(this.y>windowHeight-97,0,0.0001,0.1);

    // set the dimension of the ellipse
    this.dimension=map(this.freq,2000,5,3,100);

    // oscillator associated to Dot
    this.osc = new p5.Oscillator();
    this.osc.setType('sine');
    this.osc.freq(this.freq);
    this.osc.amp(0);
    this.osc.start();

    // LFO triangle for Amplitude Modulation
    this.ampModulator = new p5.Oscillator('triangle');
    this.ampModulator.disconnect(); // disconnected
    this.ampModulator.freq(random(0.2)+0.001);
    this.ampModulator.amp(1);
    this.ampModulator.start();

    // apply Amplitude Modulation
    this.osc.amp(this.ampModulator.scale(-1,1,0.0,this.amp));
}
```

5.5 OBJECTS

we use the LFO triangle oscillator as a amplitude modulator for the Amplitude of the sine oscillator.

we use the scale method for scaling the amplitude between -1, 1, to 0 and the amplitude of the current sine oscillator

5.6 p5.js OPERATIVE FUNCTIONS

```
function setup() { ... }  
function windowResized() { ... }  
function draw() { ... }  
function mousePressed() { ... }
```

setup() is a function invoked just one time when the code is loaded.

windowResized() is a function invoked every time the user changes the dimension of the browser window

draw() is called directly after **setup()**. It continuously executes the lines of code contained inside its block until the program is stopped (60 frames per second).

mousePressed() is called once after every time a mouse button is pressed

5.6 p5.js OPERATIVE FUNCTIONS

createCanvas create the canvas where is possible to draw the p5.js elements (like our Dot)

windowWidth: system variable that stores the width of the inner window

windowHeight: system variable that stores the height of the inner window

```
function setup() {  
  createCanvas(windowWidth, windowHeight-97);  
}  
function windowResized() {  
  resizeCanvas(windowWidth, windowHeight-97);  
}
```

resizeCanvas resize the canvas created before

5.7 ARRAY

```
var array0fDots=[];
```

we create an empty array of Dots

An array is a special variable, which can hold more than one value at a time.

```
var array_name = [item1, item2, ...];
```

You access an array element by referring to the **index number**.

array_name[0] is the item1.

array_name[1] is the item2.

5.8 MOUSEPRESSED

```
// MOUSE PRESSED FUNCTION -> ADD A DOT OR DELETE A DOT
function mousePressed() {
    var exist=false; // we use this variable as a flag
    for(var i=0; i<arrayOfDots.length; i++){
        if(arrayOfDots[i]){
            var d = dist(arrayOfDots[i].x,arrayOfDots[i].y, mouseX, mouseY);
            if(d<arrayOfDots[i].d)
                exist=true;
            if( !keyIsDown(76) ) // delete key
                arrayOfDots[i] = null; // delete the object from the array
        }
    }
    if(!exist&&!keyIsPressed&&mouseY>0){
        var newDot = new Dot(mouseX, mouseY);
        arrayOfDots.push(newDot);
    }
}
```

we check if the object is not null

length is a property of the Arrays. It returns the number of items stored into the array. In this case we process every object in the Array of Dots.

WHAT IS A FOR LOOP?

dot

Loops can execute a block of code a number of times.

The For Loop

The `for` loop has the following syntax:

```
for (statement 1; statement 2; statement 3) {
    // code block to be executed
}
```

Statement 1 is executed (one time) before the execution of the code block.

Statement 2 defines the condition for executing the code block.

Statement 3 is executed (every time) after the code block has been executed.

5.8 MOUSEPRESSED

```
// MOUSE PRESSED FUNCTION -> ADD A DOT OR DELETE A DOT
function mousePressed() {
    var exist=false; // we use
    for(var i=0; i<arrayOfDots.length;
        if(arrayOfDots[i]){
            var d = dist(arrayOfDots[i].x,arrayOfDots[i].y,mouseX,mouseY);
            if(d<arrayOfDots[i].dimension/2){ // if pressing on an existing dot
                exist=true;
                if(!keyIsDown(76)&&mouseY>0){
                    arrayOfDots[i].ampModulator.amp(0,0.2);
                    arrayOfDots[i]=null; // delete the object from the array
                }
            }
        }
    if(!exist&&!keyIsDown(76)){
        var newDot = {
            arrayOfDots.push(new Dot());
    }
}
```

we calculate the distance between each Dot in the array and the position of the mouse. `dist()` is a method of p5.js that calculate the distance between two points.

if we are pressing into the circle of the current Dot, it means that it exists, and in this case we **DELETE it, sending the `ampModulator` amplitude to 0 in 0.2 seconds and than delete the object from the array (`null`).**

$$d(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

5.8 MOUSEPRESSED

```
// MOUSE PRESSED FUNCTION -> ADD A DOT OR DELETE A DOT
function mousePressed() {

    var exist=false; // we use this variable as a flag
    for(var i=0; i<arrayOfDots.length; i++){

        if(arrayOfDots[i]){

            var d = dist(arrayOfDots[i].x,arrayOfDots[i].y,mouseX,mouseY);

            if(d<arrayOfDots[i].dimension/2){ // if pressing on an existing dot
                exist=true;
                if( !keyIsDown(76)&&mouseY>0){
                    arrayOfDots[i].ampModulator.amp(0,0.2);
if there is no Dot in the position of the mouse, we create a new Dot (using the constructor we created before) and then we push it into the Array of Dots with the method push.
                }
            }
        }
    }

    if(!exist&&!keyIsPressed&&mouseY>0){
        var newDot = new Dot(mouseX,mouseY);
        arrayOfDots.push(newDot);
    }
}
```

The push () method adds new items to the end of an array, and returns the new length.

5.9 DRAW

```
// DRAW FUNCTION
function draw(){
  clear(); // c
  for(var i=0; i<arrayOfDots.length; i++){
    if(arrayOfDots[i]){
      var alpha=0.2;
      fill('rgba(0,0,255,' + alpha + ')');
      strokeWeight(0.1);
      ellipse(arrayOfDots[i].x, arrayOfDots[i].y,
arrayOfDots[i].dimension, arrayOfDots[i].dimension);
    }
  }
}
```

strokeWeight() sets the width of the stroke used for lines, points, and the border around shapes. All widths are set in units of pixels.

fill() sets the color used to fill shapes. The shapes drawn after the fill command will be filled with the decided color. The color is either specified in terms of the **RGB** or **HSB** color depending on the current **colorMode()**.

ellipse() draws an ellipse to the screen. An ellipse with equal width and height is a circle. The first two parameters set the location (x,y), and the third and fourth parameters set the shape's width and height.

thank you <3

REFERENCES

1. <https://www.howtogeek.com/343877/what-is-an-api/>
2. <https://www.techopedia.com/definition/23898/web-programming>
3. https://developer.mozilla.org/it/docs/Web/API/Web_Audio_API
4. <https://developer.mozilla.org/en-US/docs/Learn/JavaScript>
5. <https://interlucid.com/web-audio-api/>
6. https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API/Basic_concepts_behind_Web_Audio_API
7. <https://cdnjs.com/libraries/p5.js>
8. <https://p5js.org/reference/#/libraries/p5.sound>