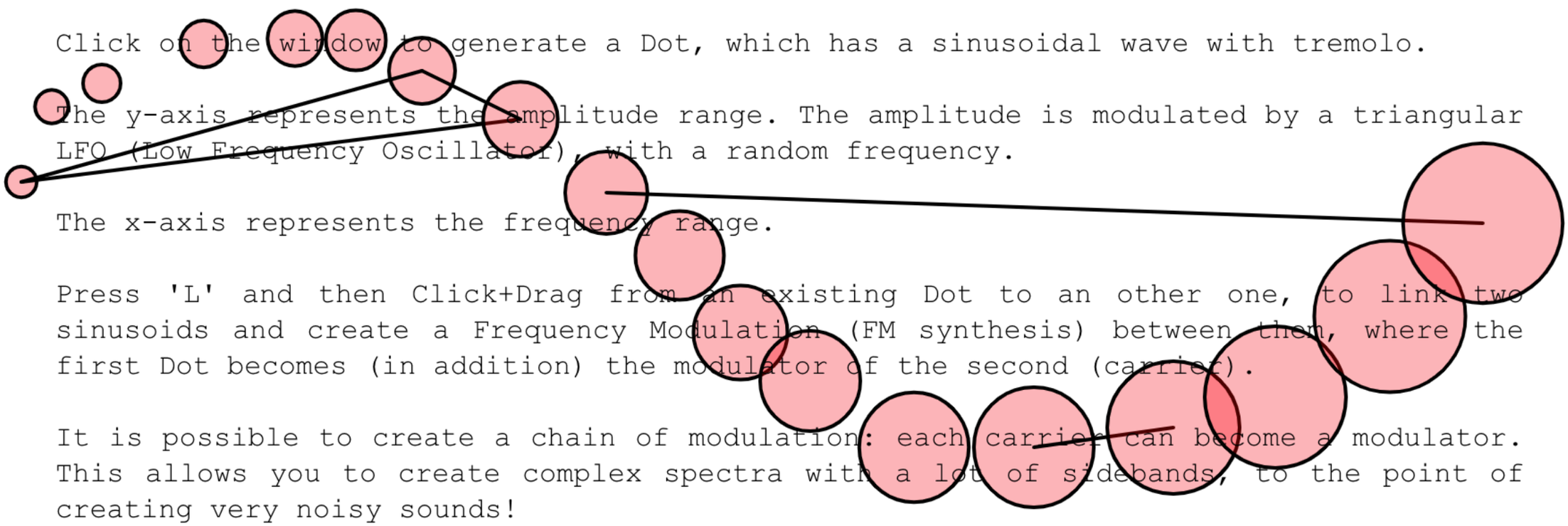
an introduction to the WEB AUDIO API

Day 2

highSCORE

Dot Drone Generator is a drone generator which allows you to create synthetic textures and chords directly on your browser.



Click on the window to generate a Dot, which has a sinusoidal wave with tremolo.

The y-axis represents the amplitude range. The amplitude is modulated by a triangular LFO (Low Frequency Oscillator), with a random frequency.

The x-axis represents the frequency range.

Press 'L' and then Click+Drag from an existing Dot to an other one, to link two sinusoids and create a Frequency Modulation (FM synthesis) between them, where the first Dot becomes (in addition) the modulator of the second (carrier).

It is possible to create a chain of modulation: each carrier can become a modulator. This allows you to create complex spectra with a lot of sidebands, to the point of creating very noisy sounds!

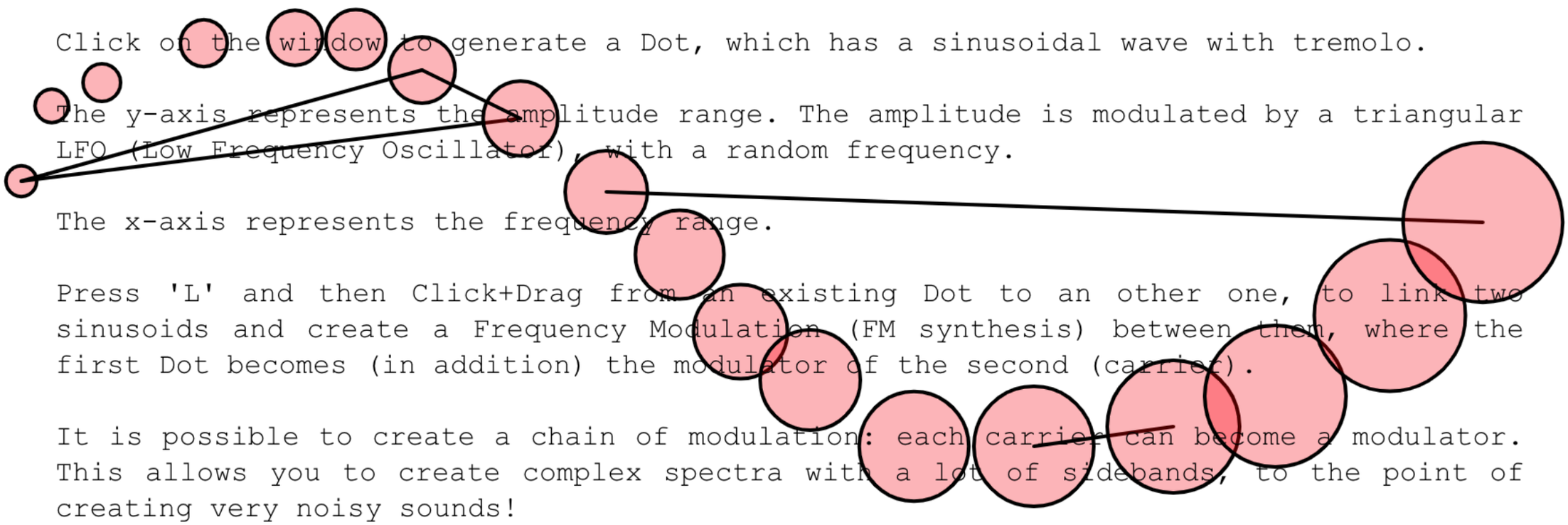
Click on an existing circle to delete it or to delete the modulation chain of which it is part.

Alberto Barberis

2021

3. WEB AUDIO BASIC CONCEPTS

Dot Drone Generator is a drone generator which allows you to create synthetic textures and chords directly on your browser.



Click on the window to generate a Dot, which has a sinusoidal wave with tremolo.

The y-axis represents the amplitude range. The amplitude is modulated by a triangular LFO (Low Frequency Oscillator), with a random frequency.

The x-axis represents the frequency range.

Press 'L' and then Click+Drag from an existing Dot to an other one, to link two sinusoids and create a Frequency Modulation (FM synthesis) between them, where the first Dot becomes (in addition) the modulator of the second (carrier).

It is possible to create a chain of modulation: each carrier can become a modulator. This allows you to create complex spectra with a lot of sidebands, to the point of creating very noisy sounds!

Click on an existing circle to delete it or to delete the modulation chain of which it is part.

3.1 MAIN FEATURES

- it uses a **modular routing logic**: with an architecture of type: source -> effects -> output. Very similar to that of visual programming software (such as Max/MSP) or modular analog synthesis environments.

- it works at **high bit-rate** (32-bit floats) for processing;

- it has an accurate **event scheduler** (with low latency);

- it offers the possibility to **automate parameters**, for the creation of envelopes, crossfades, LFOs, etc. ;

- it allows **real-time DSP** (Digital Signal Processing);

- it offers **various spatialization algorithms**;

3.1 MAIN FEATURES

- it has a **convolution module**, for creating effects such as: convolution reverbs, filters, cross synthesis, etc. ;

- it offers support for **sound visualization** in both time (waveform) and frequency domain (spectrum);

- it offers **standard effects modules** such as: filters, compressor, delay, reverb. etc. ;

- it has a **waveshaping module** for creating distortion and other non-linear effects;

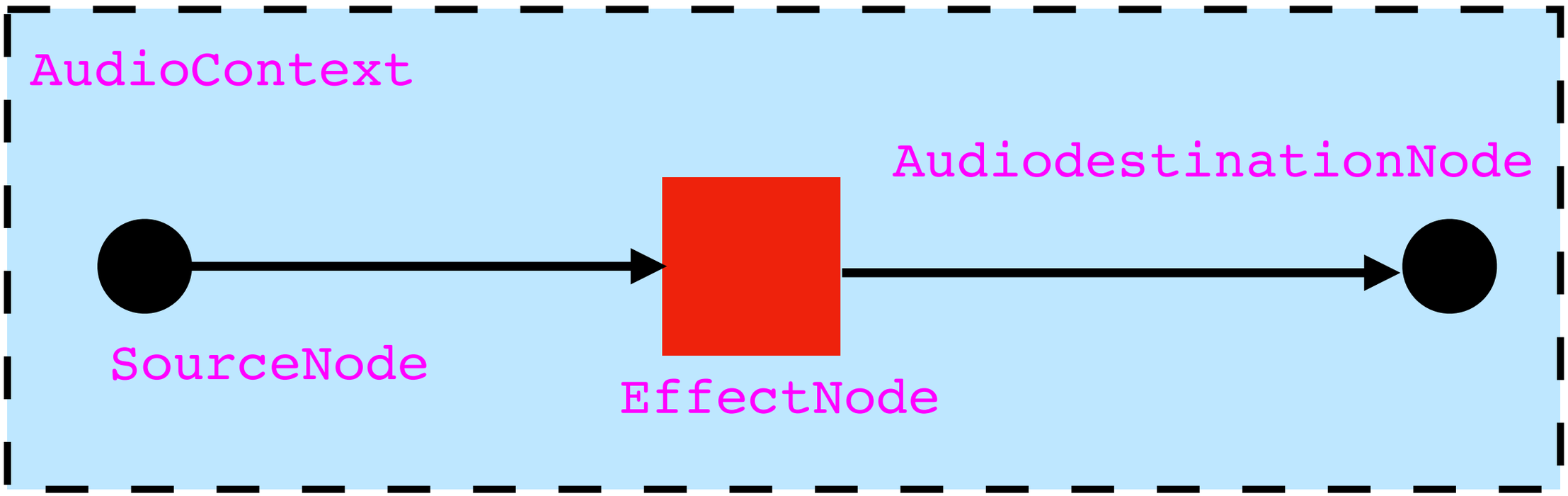
- it has a set of **standard waveform oscillators**;

- offers a **buffer module** for creating a memory array in which to store sample values.

3.1 MAIN FEATURES

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

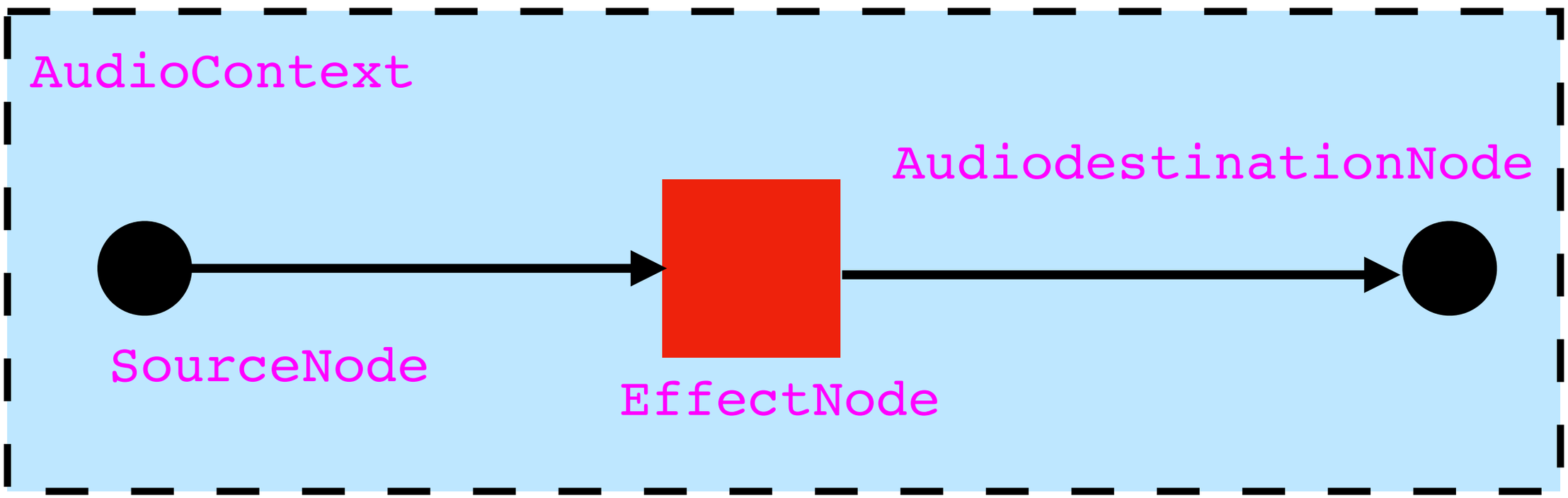
Audio operations are performed with **audio nodes**, which are linked together into **chains** to form an **audio routing graph**.



3.1 MAIN FEATURES

A **graph** typically starts with one or more **audio sources** (oscillators or samples). Outputs of these nodes could be linked to inputs of others.

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.



3.2 MODULARITY AND AUDIO CONTEXT

The Web Audio API works with **modular logic**, where **nodes** are connected together to create a stream of signals.

These nodes are created and connected to each other in an **audioContext**, that provides us with the audio rendering context.

Nodes can be of different types, for example *source nodes*, *effect nodes*, *parameter nodes*, *destination nodes*, etc..

EffectNode

AnalyzerNode

DestinationNode

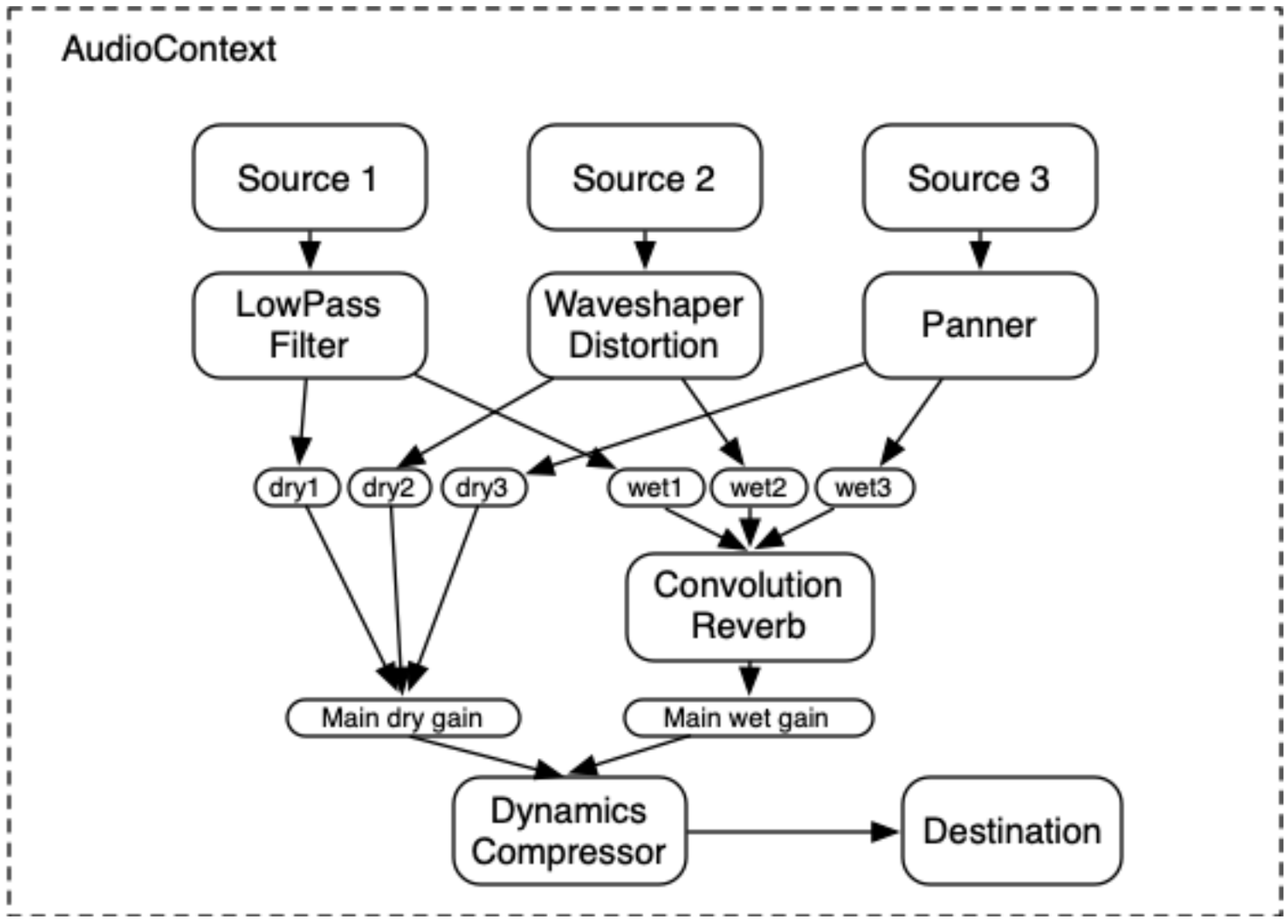
SourceNode

ParameterNode

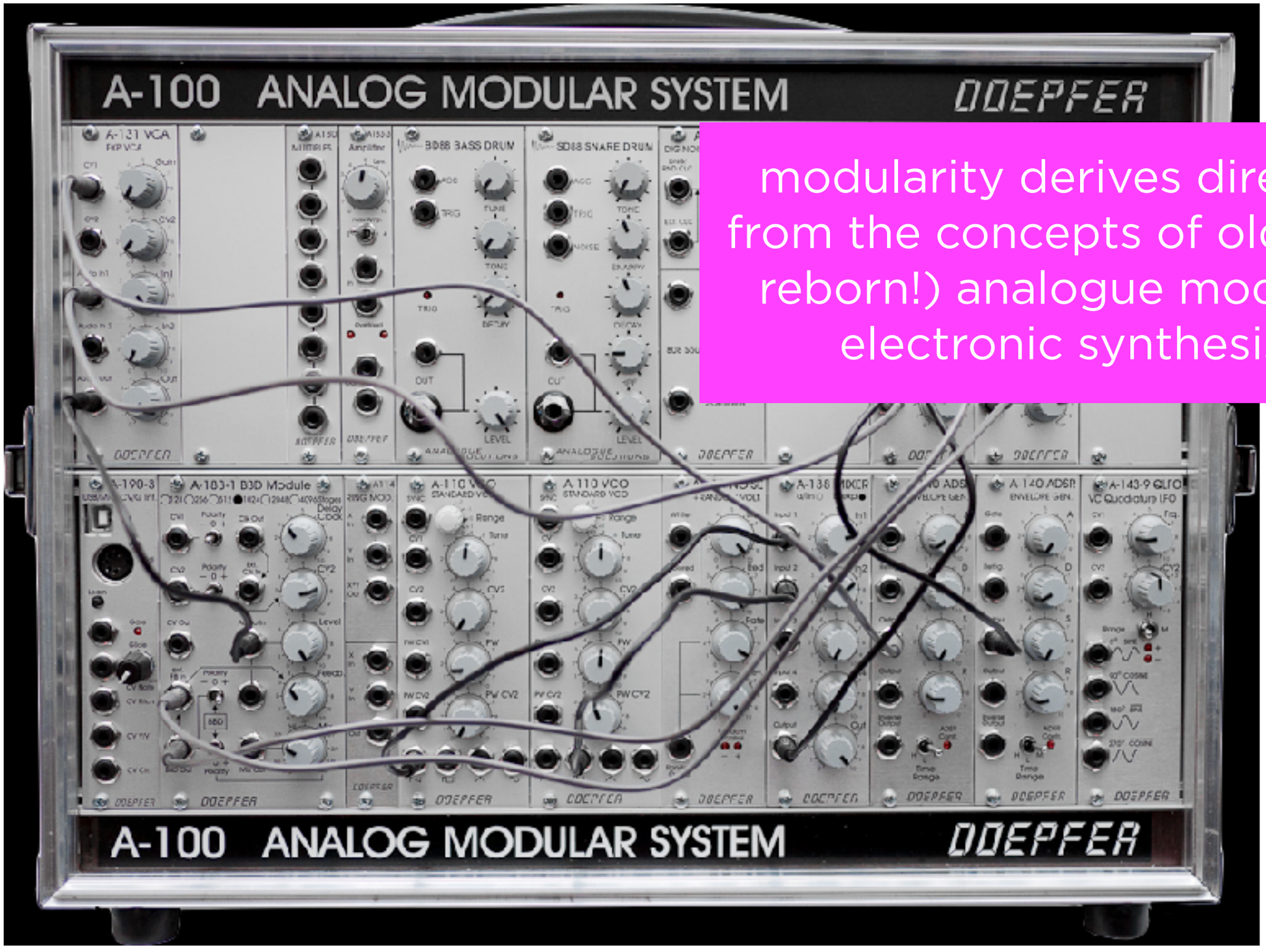
3.2 MODULARITY AND AUDIO CONTEXT

A **graph** is a collection of nodes. Any configuration of nodes (graphs) can be created in the audioContext.

Here is an example:



3.2 MODULARITY AND AUDIO CONTEXT



modularity derives directly from the concepts of old (but reborn!) analogue modular electronic synthesis

3.3 EXAMPLE OF WORKFLOW

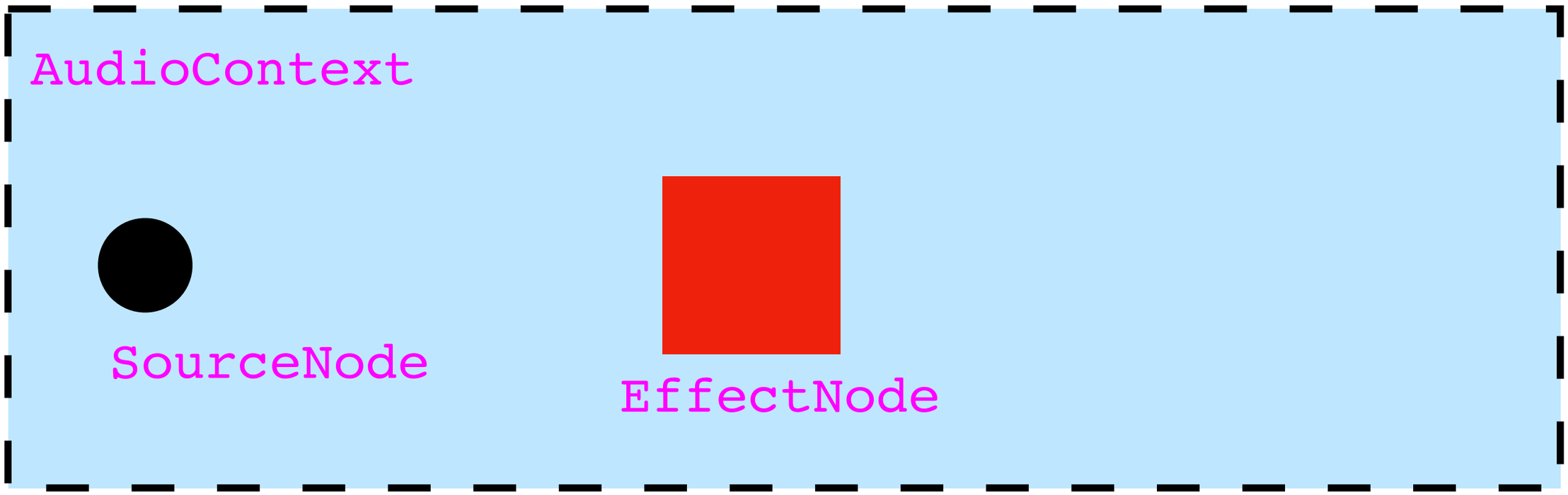
1. **Create an audio context**: an `AudioContext` is a kind of container for `AudioNode` objects, which allow different kind of audio sources and processing.

3.3 EXAMPLE OF WORKFLOW

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

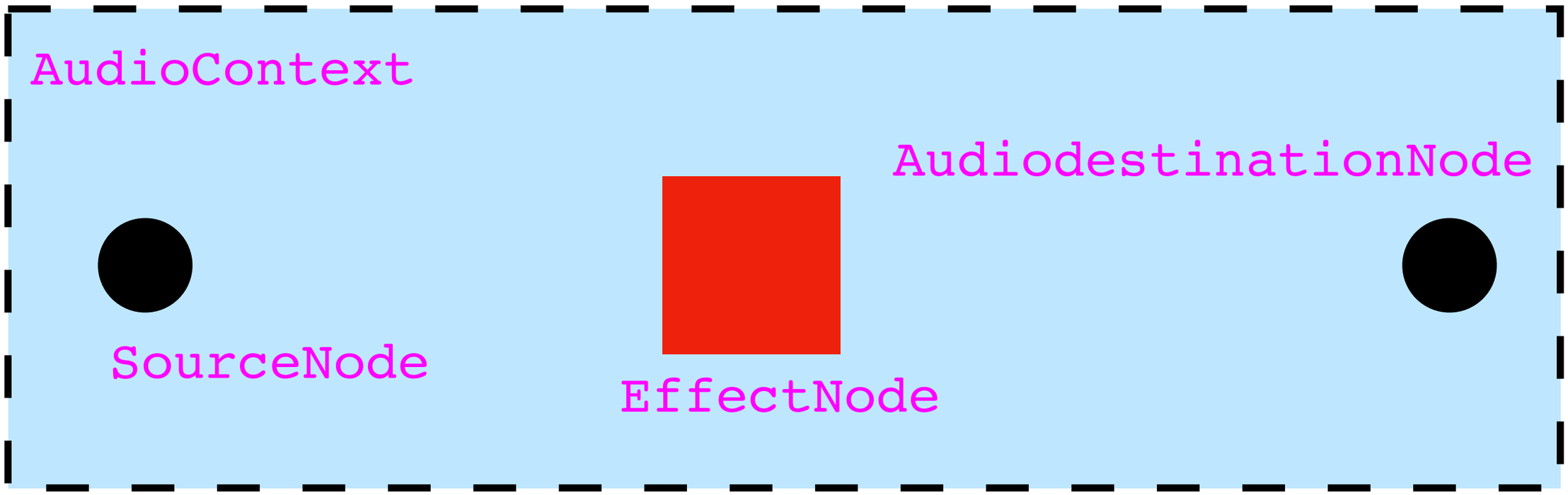
3.3 EXAMPLE OF WORKFLOW

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



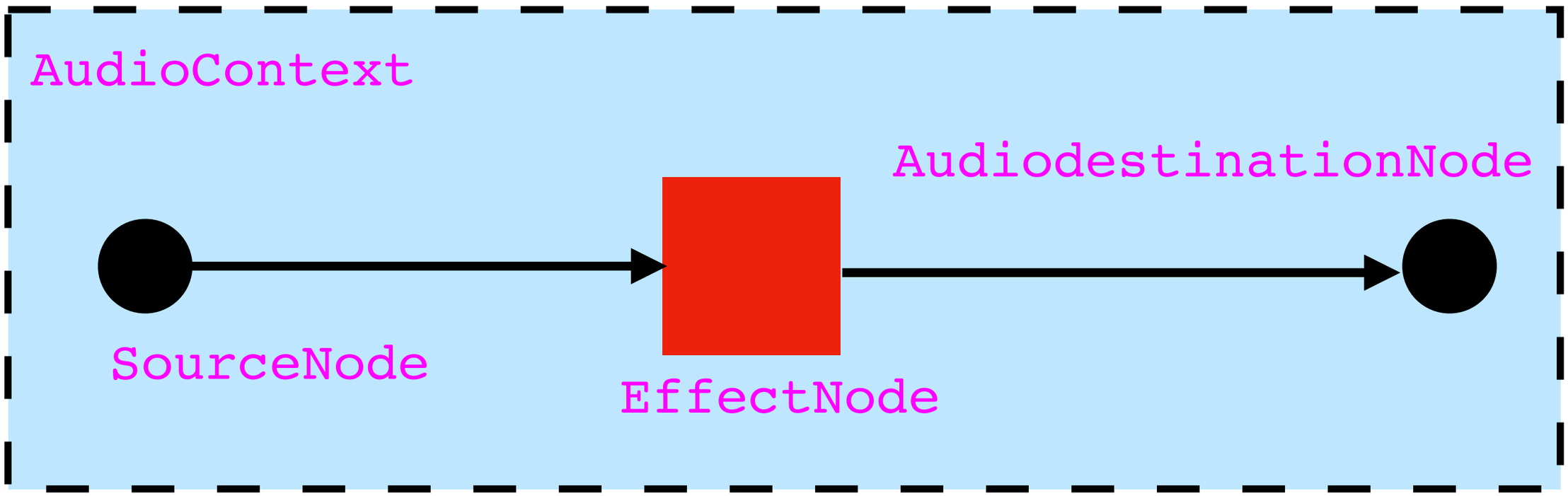
3.3 EXAMPLE OF WORKFLOW

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.

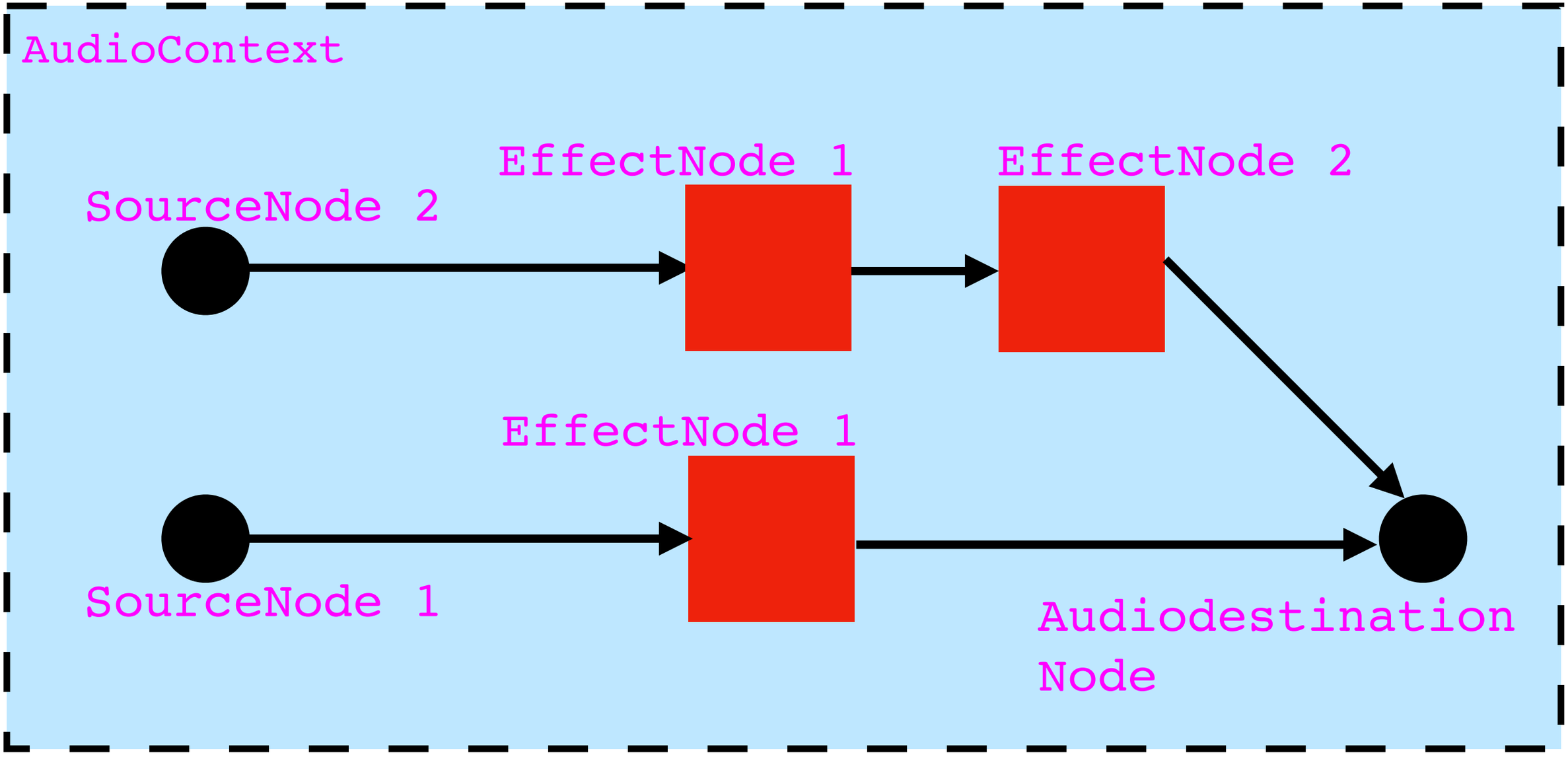


3.3 EXAMPLE OF WORKFLOW

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

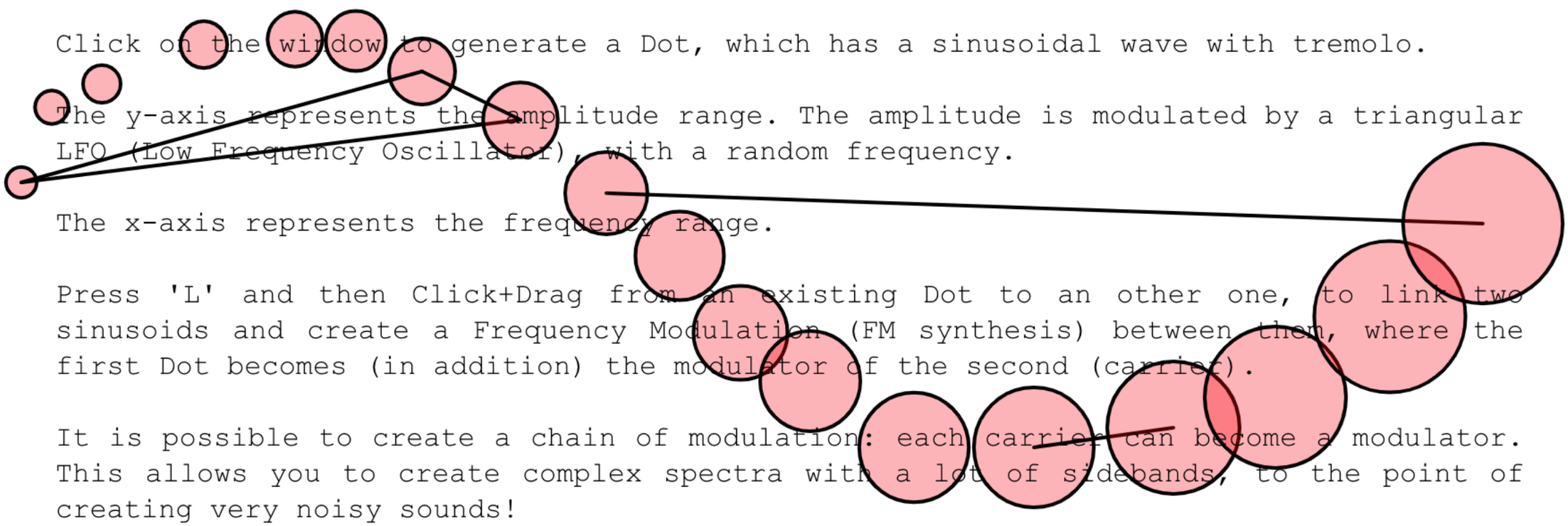


3.3 EXAMPLE OF WORKFLOW



4. WHAT TOOLS WE NEED

Dot Drone Generator is a drone generator which allows you to create synthetic textures and chords directly on your browser.



Click on the window to generate a Dot, which has a sinusoidal wave with tremolo.

The y-axis represents the amplitude range. The amplitude is modulated by a triangular LFO (Low Frequency Oscillator), with a random frequency.

The x-axis represents the frequency range.

Press 'L' and then Click+Drag from an existing Dot to an other one, to link two sinusoids and create a Frequency Modulation (FM synthesis) between them, where the first Dot becomes (in addition) the modulator of the second (carrier).

It is possible to create a chain of modulation: each carrier can become a modulator. This allows you to create complex spectra with a lot of sidebands, to the point of creating very noisy sounds!

Click on an existing circle to delete it or to delete the modulation chain of which it is part.

4.1 THE WEB AUDIO SKILLS

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

1. **web programming languages** (HTML, CSS, javascript, web audio API, p5.js, etc.);

2. **sound synthesis** methods (additive synthesis, modulation synthesis, granular synthesis, wavetable look-up synthesis, etc.);

3. **digital signal processing** algorithms (filters, delay, reverb, distortion, etc.);

4. **music theory and music creativity** (harmony, timbre, form, composition processes, ideas, etc.).

4.1 THE WEB AUDIO SKILLS

Let's have a look to:

- **HTML language (4.2)**

- **CSS language (4.3)**

(We will deal with **javascript** in the next lessons!)

And let's review the **sound synthesis method** that we will use in the app:

- **tremolo** (kind of **AM**) **(4.4)**

- **FM synthesis (4.5)**

4.2 HTML

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**;

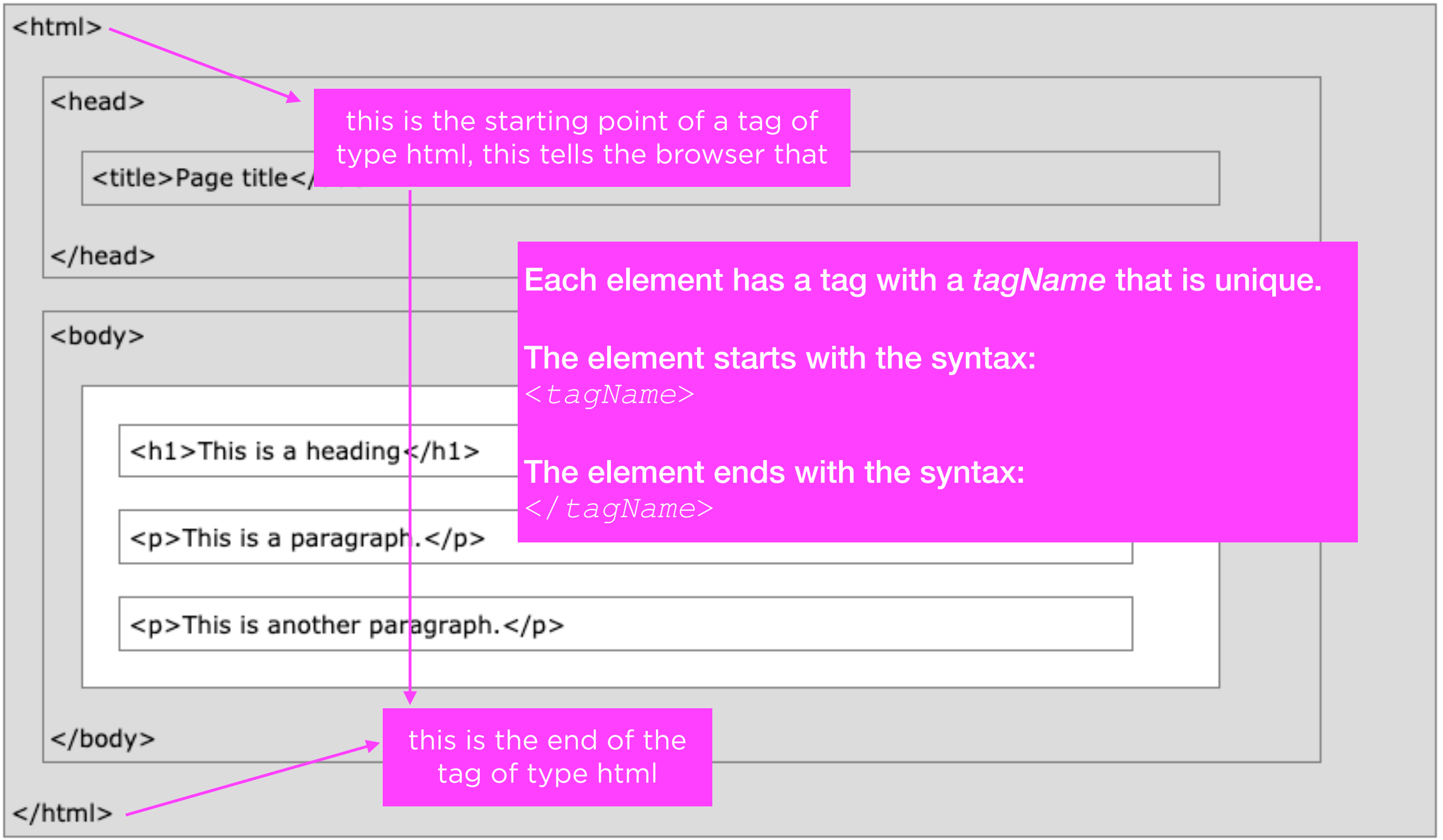
- HTML elements tell the browser **how to display the content**;

- HTML elements are represented by **tags**;

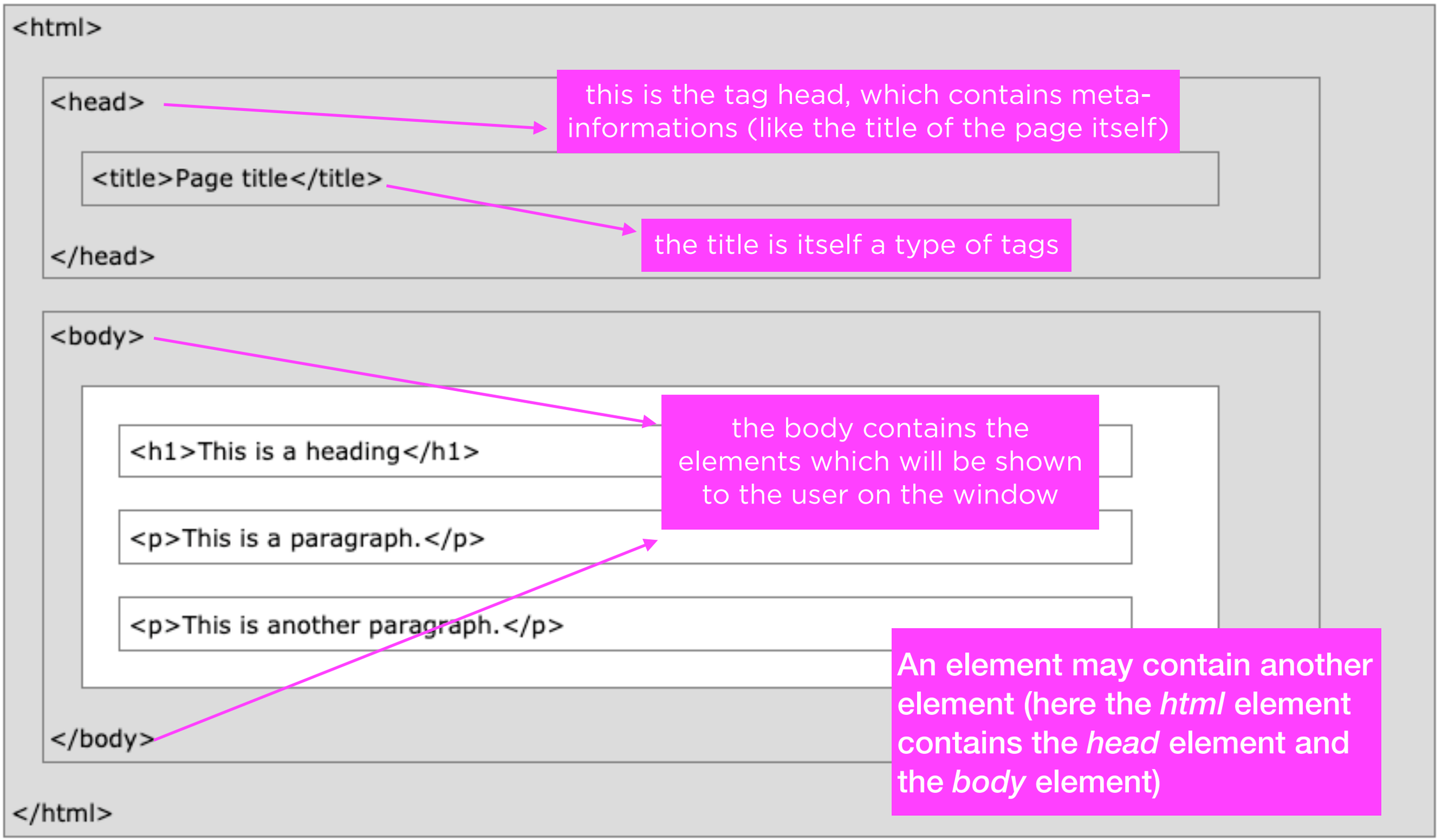
- HTML tags label pieces of content such as "heading", "paragraph", and so on;

- Browsers use HTML tags to render the content of the page;

4.2 HTML


<html>
<head>
<title>Page title</
</head>
<body>
<h1>This is a heading</h1>
This is a paragraph.
This is another paragraph.
</body>
</html>
this is the starting point of a tag of type html, this tells the browser that
this is the end of the tag of type html
Each element has a tag with a tagName that is unique.
The element starts with the syntax:
<tagName>
The element ends with the syntax:
</tagName>

4.2 HTML


<html>
<head>
this is the tag head, which contains meta-informations (like the title of the page itself)
<title>Page title</title>
the title is itself a type of tags
</head>
<body>
the body contains the elements which will be shown to the user on the window
<h1>This is a heading</h1>
This is a paragraph.
This is another paragraph.
An element may contain another element (here the *html* element contains the *head* element and the *body* element)
</body>
</html>

4.2 HTML

```
<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>
```

this is an element of type heading 1, that means that it will be displayed with a big font size (important, for a title)

this is an element that represents a normal paragraph (normal font size)

an introductio

Let's CODE!!

Let's open the file *index.html*

This is the HTML code that defines the structure of the web page

The tag `script` allows us to load a javascript file or library to our HTML

Here we load one library (p5.js) and the file hs2021.js that is the javascript file that we will code

```
<!-- WEB AUDIO highSCORE 2021-->

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

<html lang="en"> <!-- html tag is the root element of an HTML page -->

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

    <link rel="stylesheet" type="text/css" href="css/hs2021.css"> <!-- include css file -->

    <title>
      web audio hS2021
    </title> <!-- title tag specifies a title for the document -->

    <meta charset="utf-8">

    <!-- load the libraries we need / old versions that does not give
    <!-- <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.
    <script src="lib/p5.min.js"></script>

    <!-- load the javascript file -->
    <script src="js/hs2021.js"></script>

  </head>

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

    <h1> <!-- h1 tag defines a large heading -->
      Dot Drone Generator
    </h1>

    <p> <!-- p tag defines a defines a paragraph -->
      <a href="https://www.highscorefestival.com/" target="_blank" > highSCORE Festival 2021 </a>
      <a href="http://www.albertobarberis.it/" target="_blank" > Alberto Barberis </a>
    </p>

    <div id="infoText"> <!-- div tag defines a divi

      Dot Dron Generator is a drone generator
      <br><br>
      Click on the window to generate a Dot, which
      <br><br>
      The y-axis represent the amplitude range. The
      <br><br>
      The x-axis represents the frequency range.
      <br><br>
      Press 'L' and then Click+Drag
      <br><br>
      It is possible to create a chai
      <br><br>
      Click on an existing circle to

    </div>
  </body>
</html>
```

This is the CSS file that you find in the folder css. We use the link tag to link it to our HTML file

This is the JAVASCRIPT file that we are going to create soon

This is a div element, a general section of the HTML file where we insert a description text

`id` specifies a unique ID address (that we will use to refer to this specific element)

Let's CODE!!

Dot Drone Generator

highSCORE Festival 2021 | Alberto Barberis

Dot Drone Generator is a dr
textures and chords directly o

This is the web page that you should see on your browser when you double click on the index.html file

Click on the window to generate a Dot, which has a sinusoidal wave with tremolo.

The y-axis represents the amplitude range. The amplitude is modulated by a triangular LFO (Low Frequency Oscillator), with a random frequency.

The x-axis represents the frequency range.

Press 'L' and then Click+Drag from
sinusoids and create a Frequency Mo
first Dot becomes (in addition) the m

If you click on the page nothing happens yet because the JAVASCRIPT file that will manage the user interaction does not exist yet

It is possible to create a chain of
This allows you to create complex s
creating very noisy sounds!

Click on an existing circle to delete it or to delete the modulation chain of which it is part.

4.3 CSS

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

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 **logical separation** of presentation/sty ure (HTML).

HTML (contents and structure)

CSS (style and presentation)

4.3 CSS

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

CSS works **applying some style 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.

Some rules have a **higher priority** (specificity) than others and this priority depends on the manner in which the rules are defined.

https://developer.mozilla.org/en-US/docs/Web/CSS/Specificity

4.3 CSS

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

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

4.3 CSS

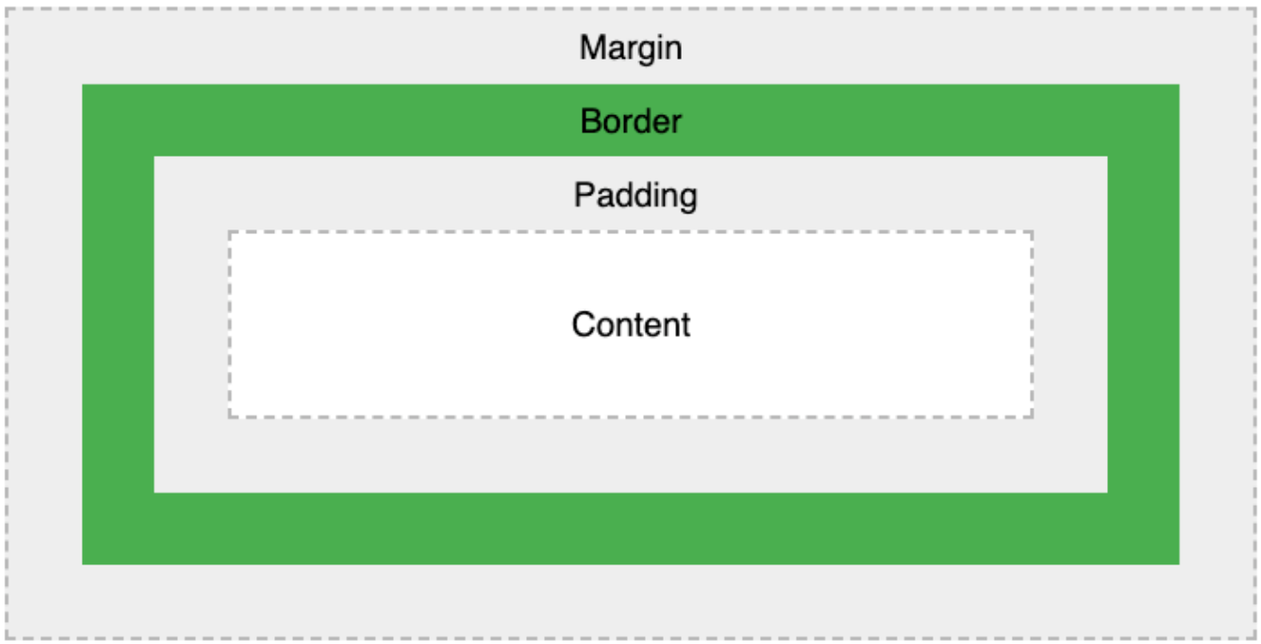
THE BOX MODEL

- **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

an introductio

Let's CODE!!

This is the CSS file that you find in the folder css. We use the `link` tag to link it to our HTML file

`rel` attribute specifies the relationship between the documents

`type` specifies the type of document

`href` sets the file name

```
<!-- WEB AUDIO highSCORE 2021-->

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

<html lang="en"> <!-- html tag is the root element of an HTML page -->

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

    <link rel="stylesheet" type="text/css" href="css/hs2021.css"> <!-- include css file -->

    <title>
      web audio hS2021
    </title> <!-- title tag specifies a title for the document -->

    <meta charset="utf-8">

    <!-- load the libraries we need / old versions that does not give us problem of security -->
    <!-- <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.4.0/p5.min.js"></script> -->
    <script src="lib/p5.min.js"></script>

    <!-- load the javascript file -->
    <script src="js/hs2021.js"></script>

  </head>

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

    <h1> <!-- h1 tag defines a large heading -->
      erator

      defines a defines a paragraph -->
      ://www.highscorefestival.com/" target="_blank" > highSCORE Festival 2021 </a>
      p://www.albertobarberis.it/" target="_blank" > Alberto Barberis </a>

      t"> <!-- div tag defines a division or a section in an HTML document; it is a general element -->

      erator is a drone generator which allows you to create synthetic textures and chords directly on you

      indow to generate a Dot, which has a sinusoidal wave with vibrato asociated to it.

      presents the amplitude range. The amplitude is modulated by a  triangular LFO (Low Frequency Oscilla
      <br><br>
      The x-axis represents the frequency range.
      <br><br>
      Press 'L' and then Click+Drag from an existing Dot to an other one, to link two sinusoids and create a Frequency
      <br><br>
      It is possible to create a chain of modulation: each carrier can become a modulator. This allows you to create c
      <br><br>
      Click on an existing circle to delete it or to delete the modulation chain of which it is part.

    </div>
  </body>
</html>
```

Let's CODE!!

4.3 CSS

```
css > # hs2021.css > ...
1   body { /* selector body */
2       margin: 0px;
3       font-family: 'Courier New';
4       user-select: none; /* user can not select the text */
5   }
```

Let's open the file hs2021.css

This is called SELECTOR: it specifies which elements the rules refer to

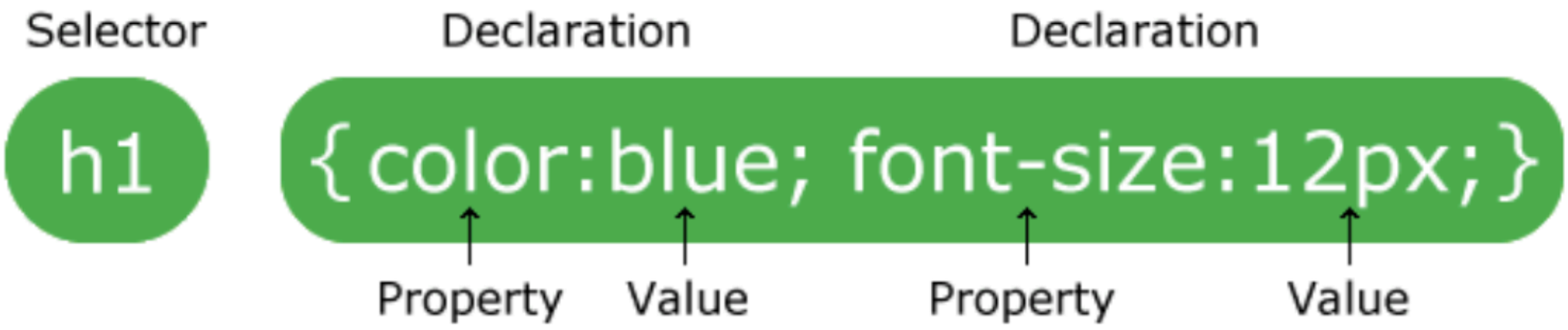
This are the rules in the form `property: value;`

The CSS language is made of different **selectors** followed by **rules**. There are different **types of selectors** and each selector refers to a specific html element or to a group of elements with some shared characteristics.

The **rules** are placed into brackets and are made of **properties** followed by a **value**.

For example here we say to the body element to have a margin of 10 px and to use the text font Courier New.

4.3 CSS



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.

```
css > # hs2021.css > ...
  1   body { /* selector body */
  2       margin: 0px;
  3       font-family: 'Courier New';
  4       user-select: none; /* user can not select the text */
  5   }
  6   h1 { /* selector h1 */
  7       margin: 0px;
  8       padding: 10px;
  9       background-color: rgb(255, 79, 79);
 10       color: white;
 11   }
 12   p { /* selector paragraph */
 13       margin: 0px;
 14       padding: 10px;
 15       color: rgb(0, 0, 0);
 16       border-bottom: solid;
 17       border-width: 1px;
 18   }
 19   #infoText { /* selector element with ID infoText */
 20       position: fixed;
 21       margin-top: 50px;
 22       margin-left: 100px;
 23       margin-right: 100px;
 24       text-align: justify;
 25       line-height: 130%;
 26   }
 27   a { /* selector element a */
 28       text-decoration: none;
 29       color: black;
 30   }
 31   a:hover { /* selector element a when mouse is over */
 32       color: rgb(255, 79, 79);
 33   }
```

This is the CSS file that sets the style and look of the window

These selectors refer to all the elements of a certain type (eg body, paragraph)

The selector that starts with # is a selector that refers to an element with a specific ID.

The div element that we use to present the text in the HTML file has a special attribute id="infoText".

This is the way in which we can match a specific element.

```
<div id="infoText">
```

This refers to the elements a but when the mouse is over the text

Here we use a lot of rules to set the position of the infoText:

position: fixed;

An element with `position: fixed;` is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled. The top, right, bottom, and left properties are used to position the element.

A fixed element does not leave a gap in the page where it would normally have been located.

Notice the fixed element in the lower-right corner of the page. Here is the CSS that is used:

Here we set some top / left / right margin

Here we justify the text

Here we set the line height to be a bit more than the default one

```
css > # hs2021.css > ...
  1   body { /* selector body */
  2       margin: 0px;
  3       font-family: 'Courier New';
  4       user-select: none; /* user can not select the text */
  5   }

                                    rgb(255, 79, 79);

                                  aph */

                                   0);
 16       border-bottom: solid;
 17       border-width: 1px;
 18   }
 19   #infoText { /* selector element with ID infoText */
 20       position: fixed;
 21       margin-top: 50px;
 22       margin-left: 100px;
 23       margin-right: 100px;
 24       text-align: justify;
 25       line-height: 130%;
 26   }
 27   a { /* selector element a */
 28       text-decoration: none;
 29       color: black;
 30   }
 31   a:hover { /* selector element a when mouse is over */
 32       color: rgb(255, 79, 79);
 33   }
```

4.4 AM SYNTHESIS (tremolo)

The **Dot Drone Generator** that we are going to develop is a digital web audio instrument that uses two modulation sound synthesis techniques called:

1. **amplitude modulation** (**AM**) for **a tremolo effect**.

2. **frequency modulation synthesis** (**FM**).

We speak of **modulation synthesis** when at least one of the 3 "free parameters" of an oscillator (amplitude / frequency / phase) varies continuously in time in relation to another signal (usually a periodic signal).

$$x(t) = A \cdot sin(2\pi ft + \varphi)$$

AM FM PM

$$x(t) = A(t) \cdot sin(2\pi f(t)t + \varphi(t))$$

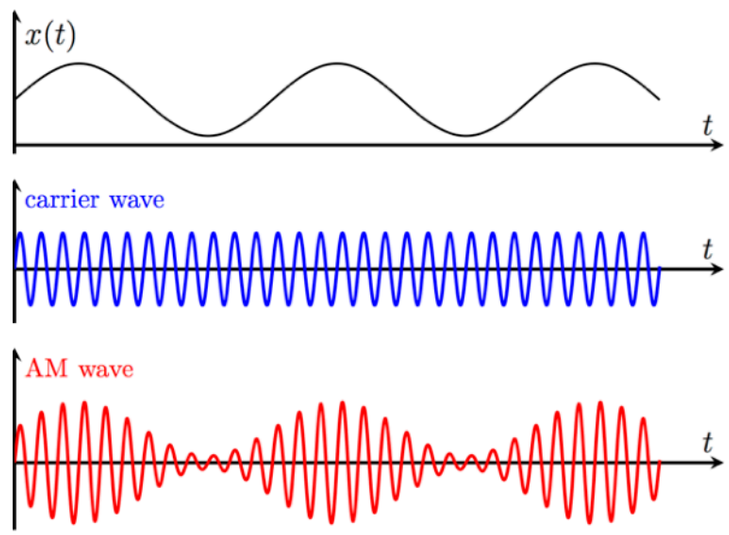
4.4 AM SYNTHESIS (tremolo)

To obtain a tremolo we will implement an AM.

A sinusoid (**carrier**) will be amplitude modulate by an other oscillator (a triangular wave **modulator**) that oscillates at Low Frequency (LFO).

If the frequency of the modulator is in the **sub audio range** (< 15 Hz) the timbre of the carrier does not change, but its volume (amplitude) change accordingly to the shape of the modulator.

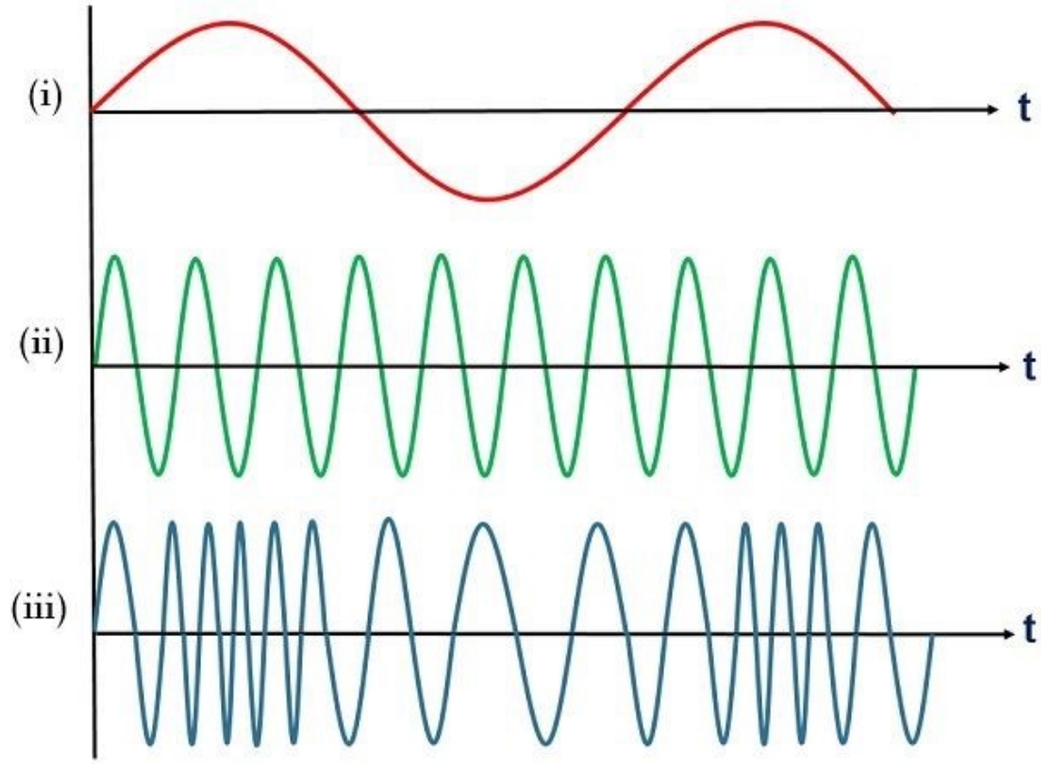
In our case the modulator will be a triangle waveform and not a sinusoid



$$x(t) = A_{tri}(t) \cdot sin(2\pi ft + \varphi)$$

4.5 FM SYNTHESIS

In the case of the **frequency modulation** the frequency of an oscillator is not fixed but it changes accordingly to an other periodic function of time (usually an other oscillator).



(i) Modulating signal
(ii) Carrier waveform
(iii) Frequency modulated signal

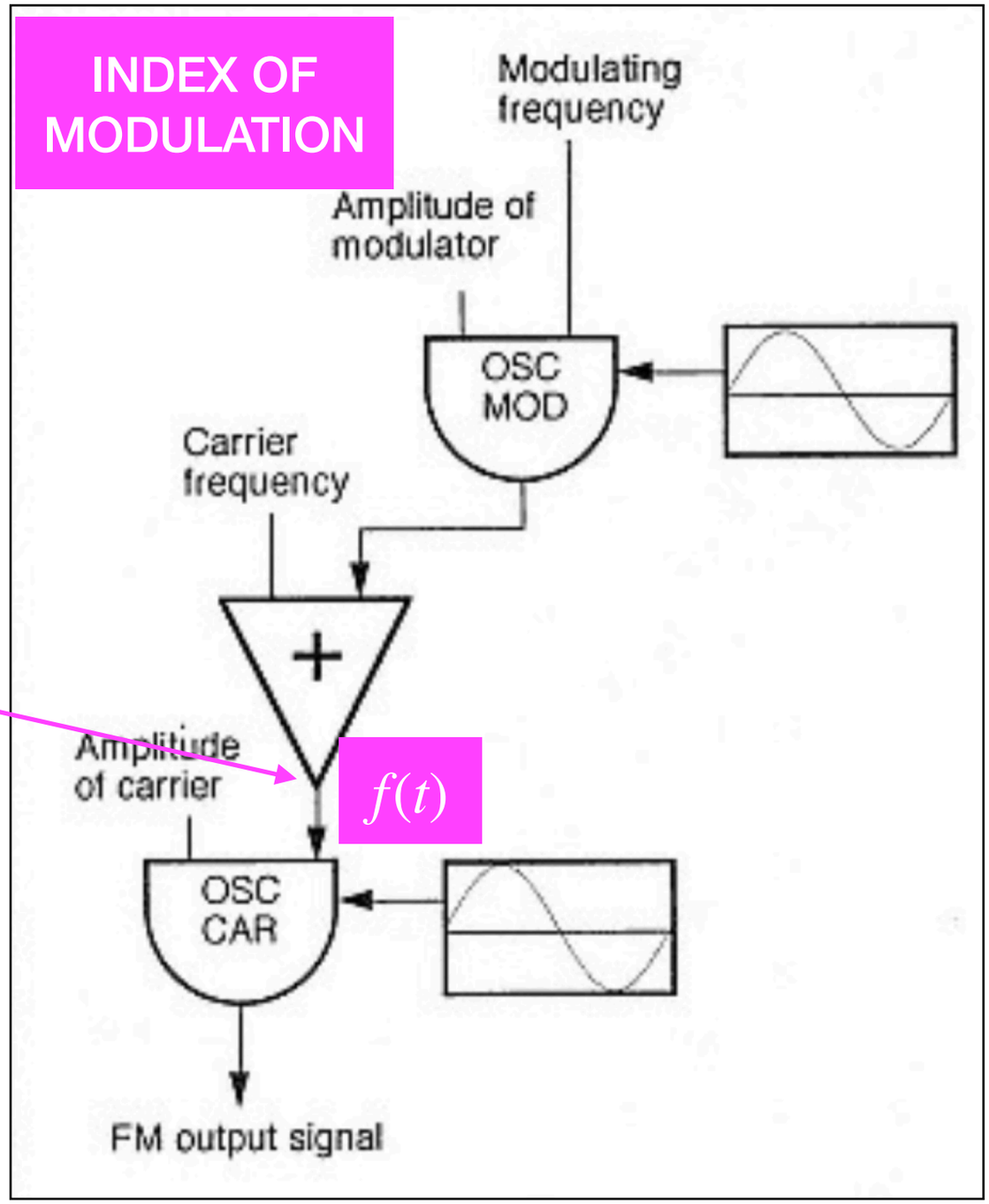
4.5 FM SYNTHESIS

In the linear FM the frequency of the original signal is no longer constant but varies over time periodically, according to the formula:

$$f(t) = f_c + A_m \cdot x_m(t)$$

Where:

- f_c = carrier frequency
- A_m = amplitude of the modulator (called **INDEX OF MODULATION**)
- $x_m(t)$ = sinusoidal modulating signal



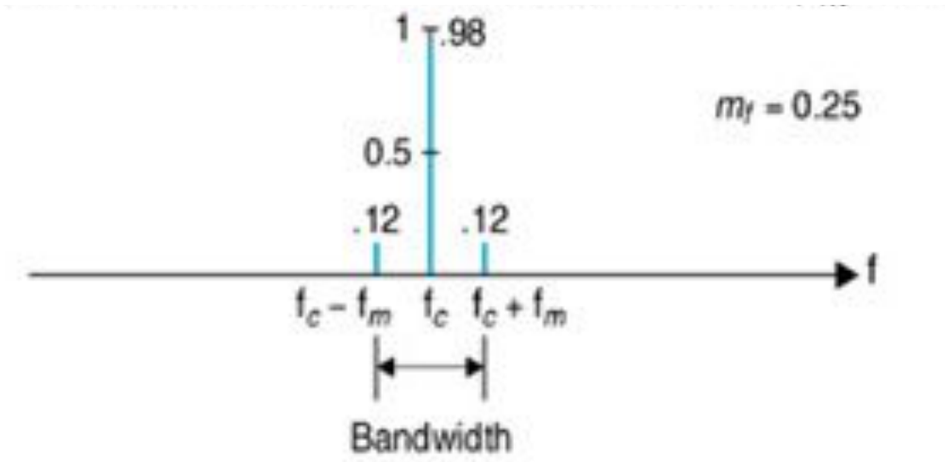
4.5 FM SYNTHESIS

The FM synthesis generates **complex spectra**, starting from simpler signals.

As the **amplitude of the modulator signal** A_m (INDEX OF MODULATION) increases, the following occurs:

- the **carrier decreases** in amplitude;
- **sidebands appear** at defined frequencies: $f_c \pm nf_m$ where n is the integer index of the order of the sidebands;
- the **spectral energy** is "stolen" from the carrier frequency and is re-distributed in some sidebands;
- after a certain threshold of A_m, the **carrier reappears** (this is because the amplitude of the components of the spectrum follows the Bessel functions).

4.5 FM SYNTHESIS



example of spectral evolution at different INDEX OF MODULATION