Conservatorio Superior de Música de Murcia
1-2 December 2021

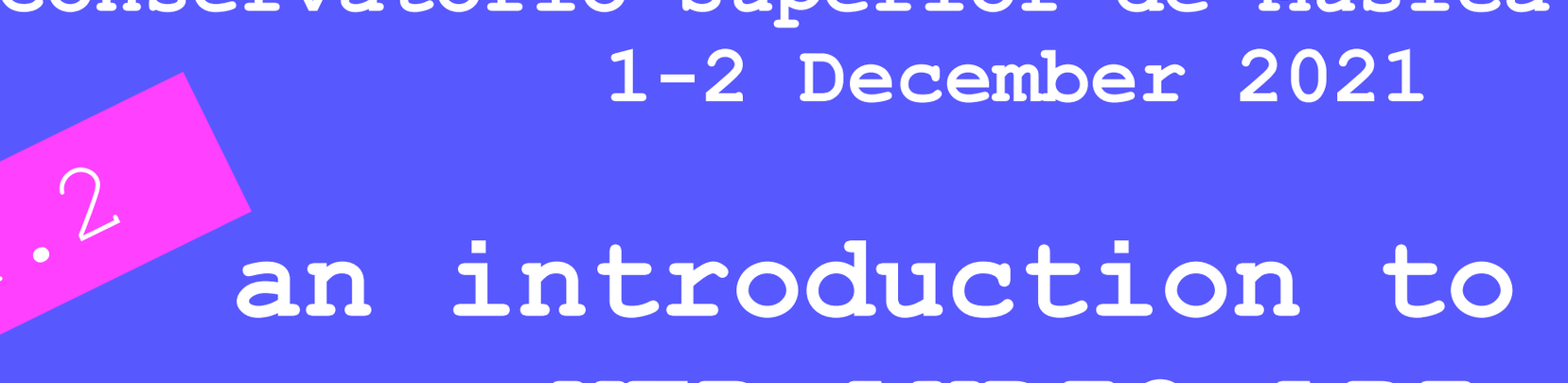
1.2

an introduction to the WEB AUDIO API

Dot Drone Generator

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

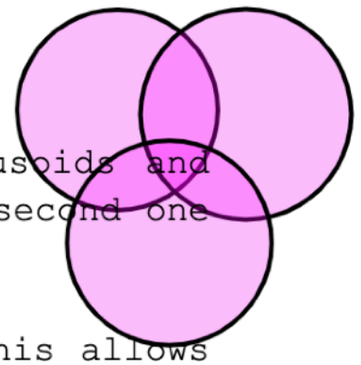
Click on the window to generate a **Dot**, a sinusoidal wave with tremolo.



The **y-axis** represents the amplitude range. The **amplitude** is modulated by a **triangular LFO** (Low Frequency Oscillator), with 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** between them. The first Dot becomes the modulator of the second one (the carrier).



It is possible to create a **chain of modulation**: each carrier can become a modulator. This allows you to create complex spectra, 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

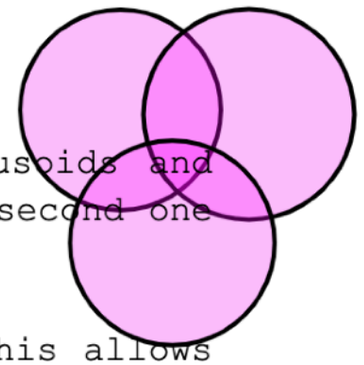
3. WEB AUDIO: BASIC CONCEPT

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

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

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

The **x-axis** represents the frequency range.



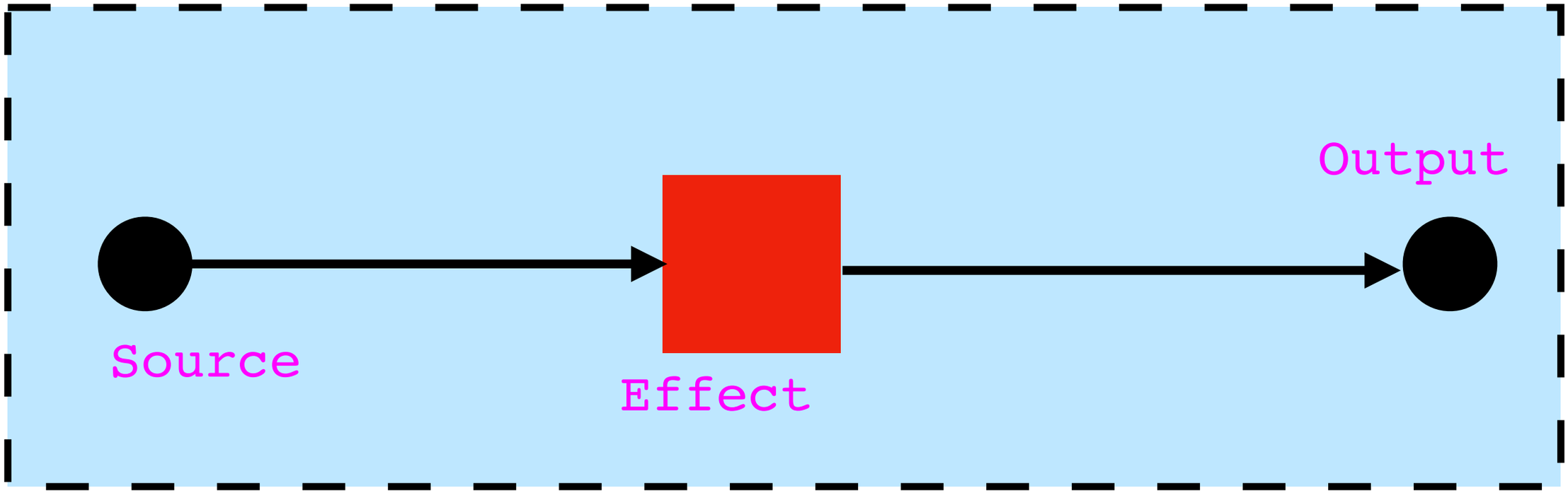
Press 'L' and then Click+Drag from an existing Dot to an other one, to link two sinusoids and create a **Frequency Modulation** between them. The first Dot becomes the modulator of the second one (the carrier).

It is possible to create a **chain of modulation**: each carrier can become a modulator. This allows you to create complex spectra, 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

1. 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.



3.1 MAIN FEATURES

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

3. it has an accurate **event scheduler** (with low latency), that allows to deal with events in time;

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

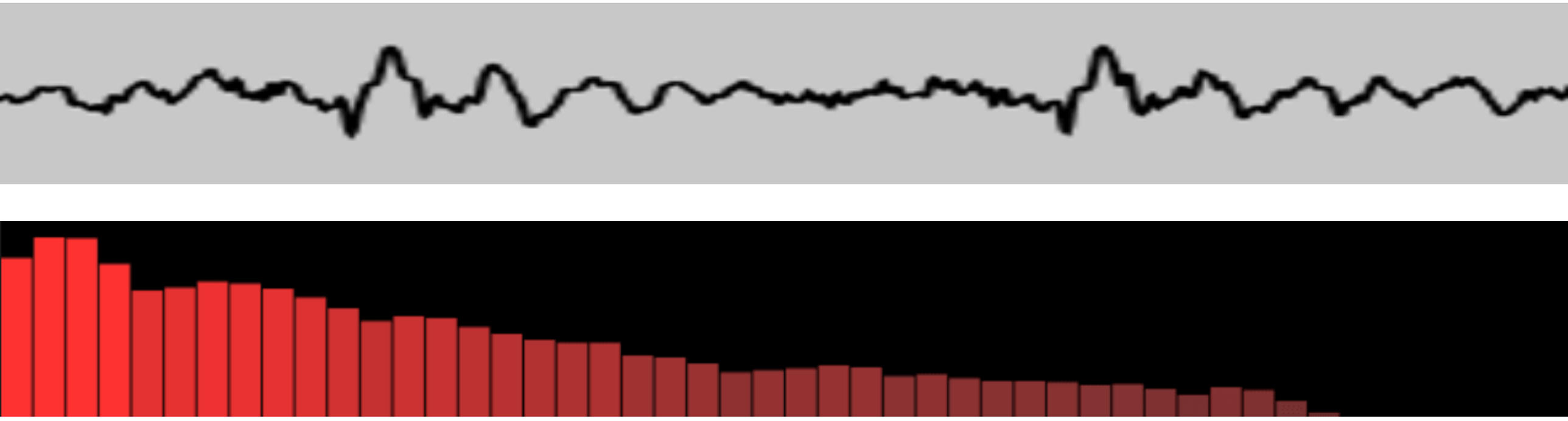
5. it allows **real-time DSP** (Digital Signal Processing);

6. it offers **various spatialization algorithms**;

3.1 MAIN FEATURES

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

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



3.1 MAIN FEATURES

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

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

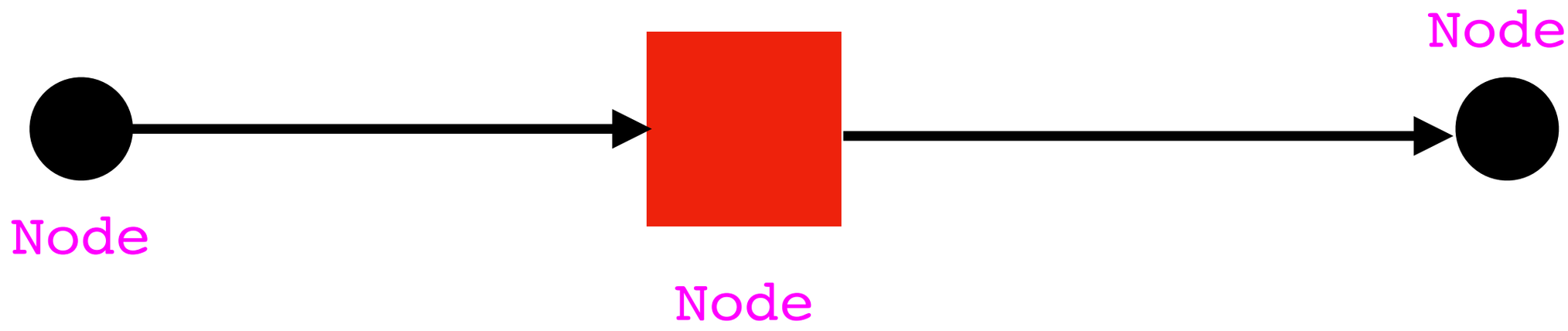
11. it has a set of **standard waveform oscillators**;

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

3.2 MODULAR DESIGN

The web audio works with **modular logic**, where **Nodes** are connected together to create a stream of signals.

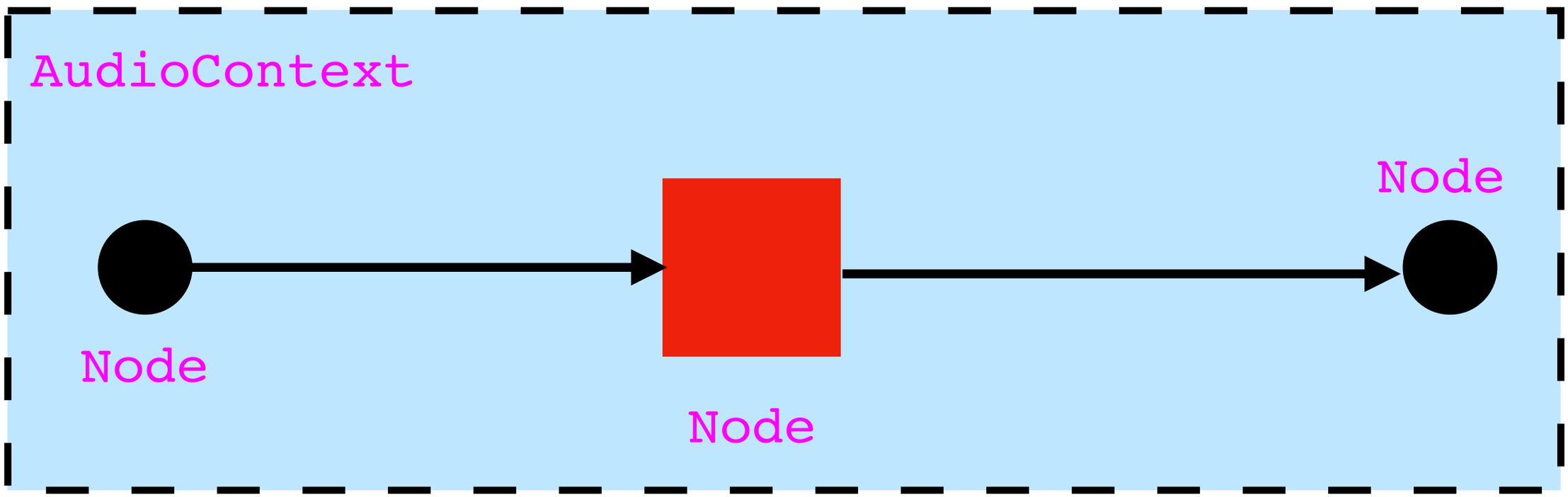
Each **Node** performs a specific audio operation on the signal. Nodes are linked together into **chains** to form an **audio routing graph**.



3.2 MODULAR DESIGN

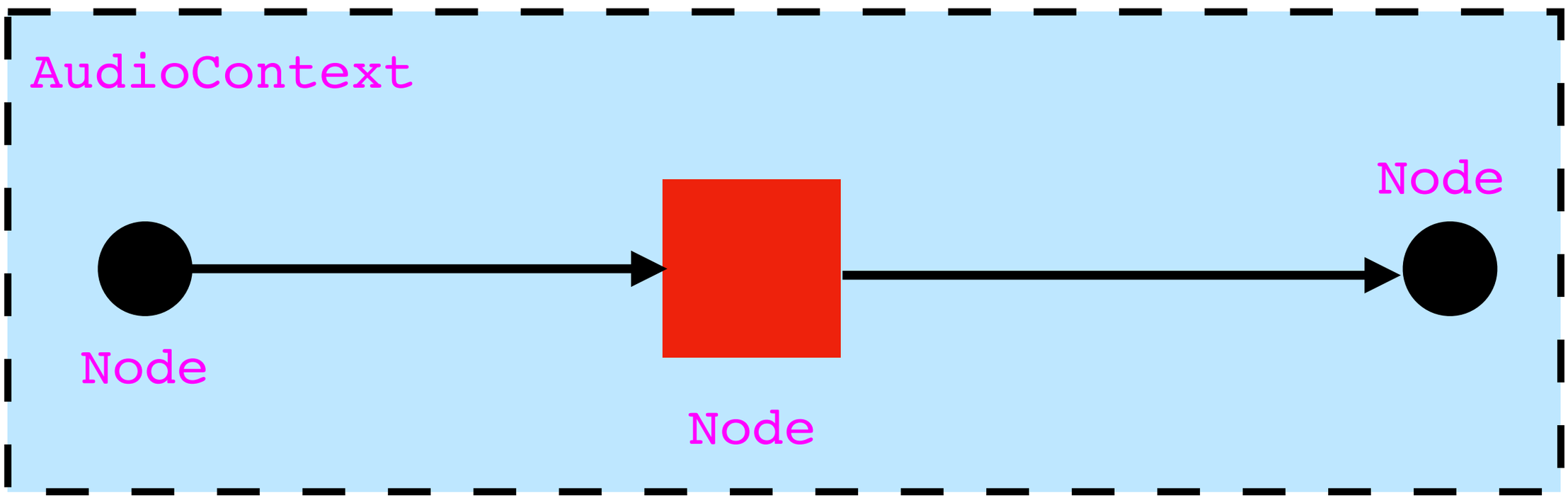
The **Nodes** are created and connected to each other in an **audioContext**, that provides us with the audio rendering context.

The web audio involves handling **audio operations** using **Nodes** inside an **audio context**.



3.2 MODULAR DESIGN

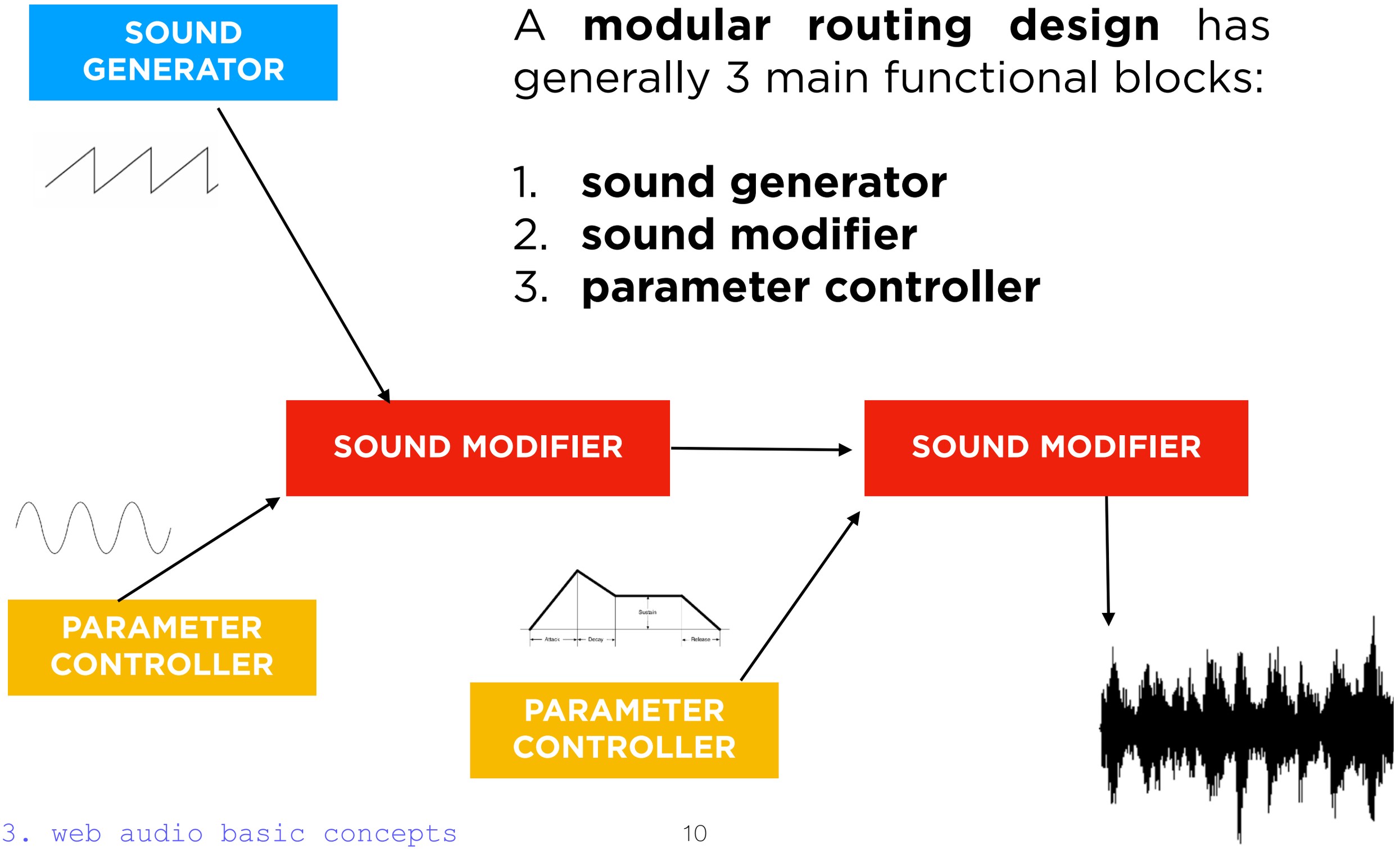
this is an **audio routing graph**
inside an **Audio Context**



3.2 MODULAR DESIGN

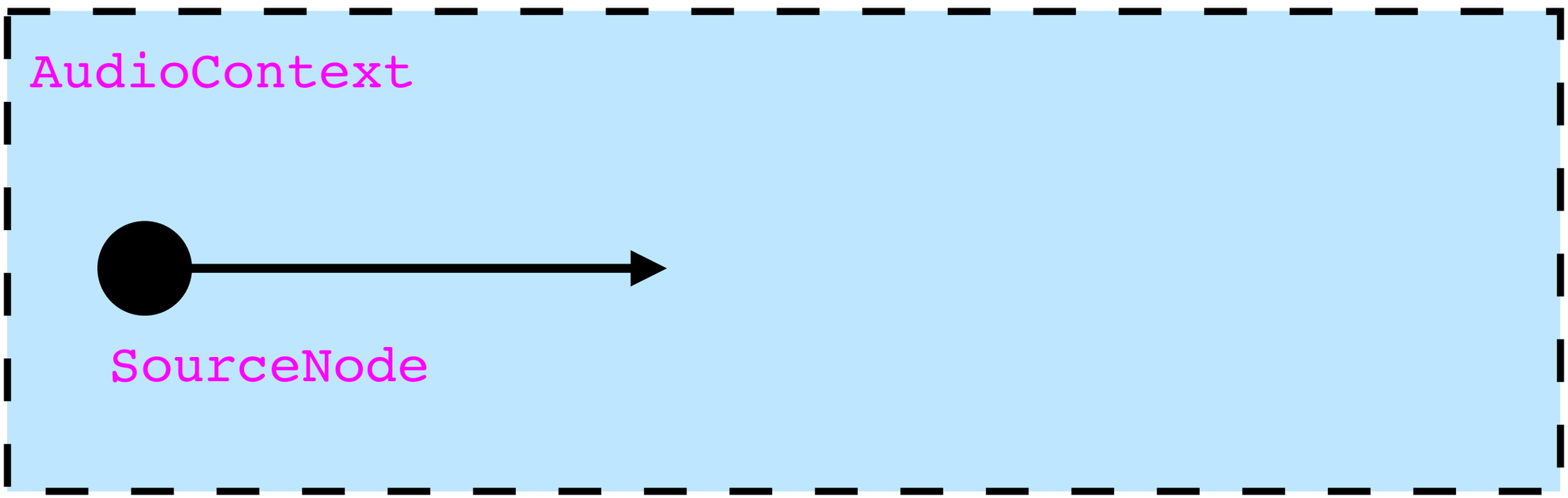
A **modular routing design** has generally 3 main functional blocks:

1. **sound generator**
2. **sound modifier**
3. **parameter controller**



3.2 MODULAR DESIGN

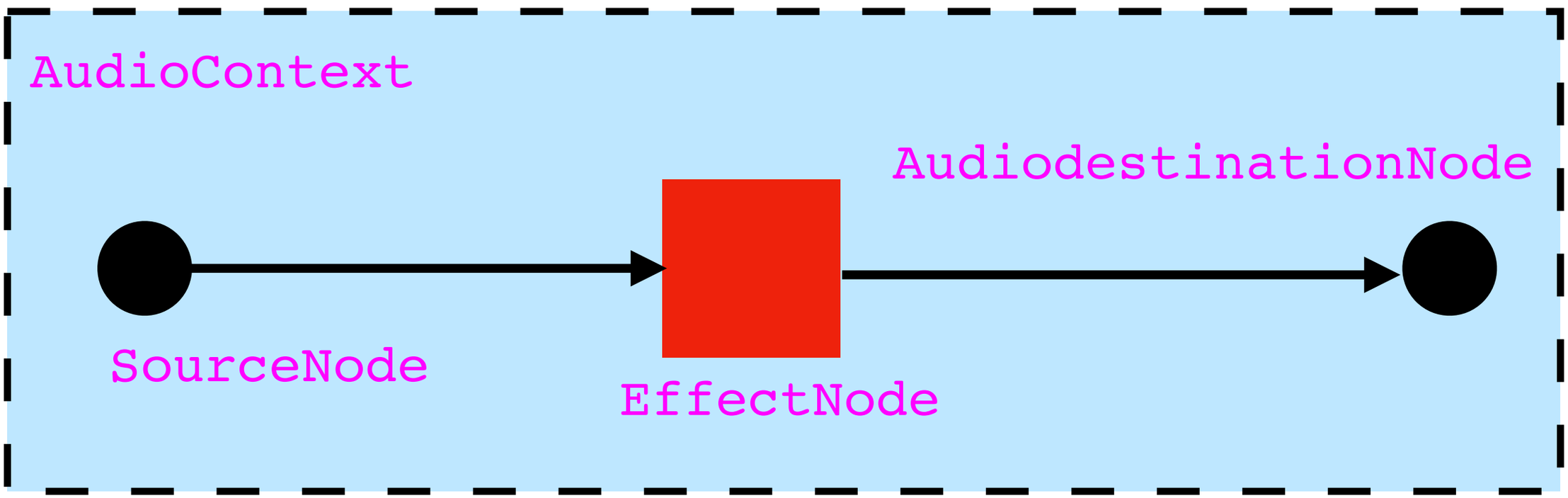
An **audio graph** typically starts with one **audio source Node** (a sound generator), like an oscillator (sine wave, triangle wave, impulse wave, sawtooth wave, wavetable, etc.), or a noise generator (white noise, pink noise), or a recorded sample.



3.2 MODULAR DESIGN

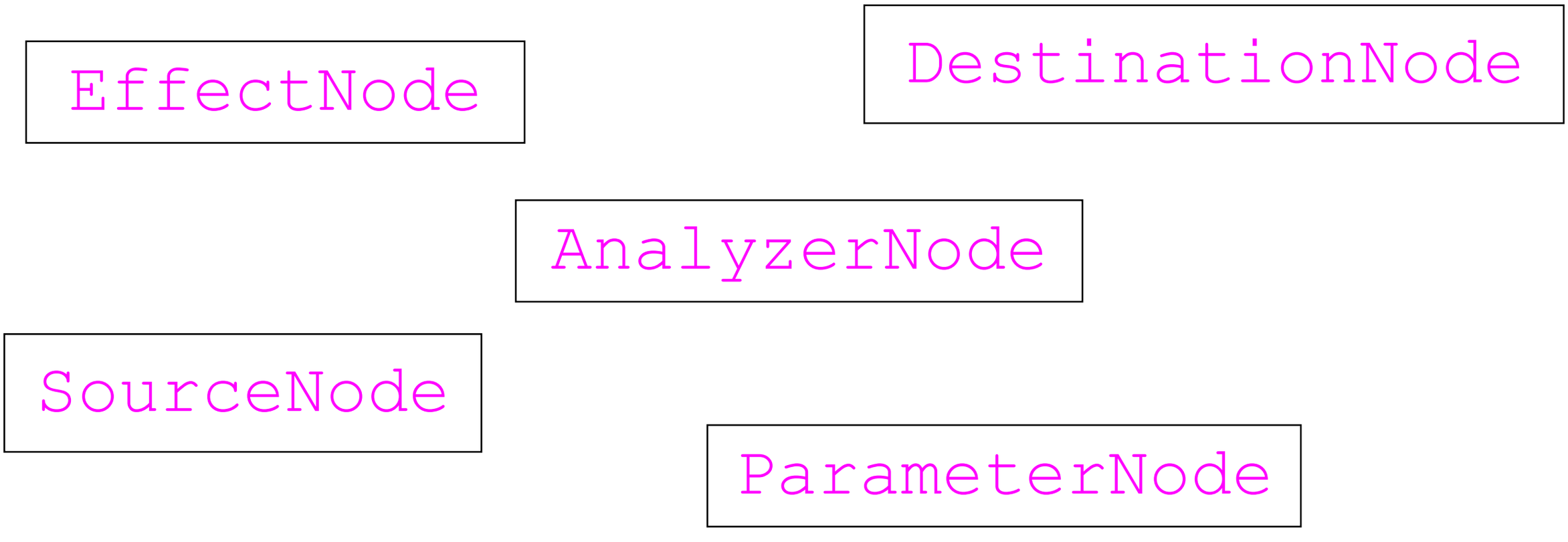
The output of a **source Node** can be linked to the input of an **audio effect Node** (a sound modifier), or more.

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



3.2 MODULAR DESIGN

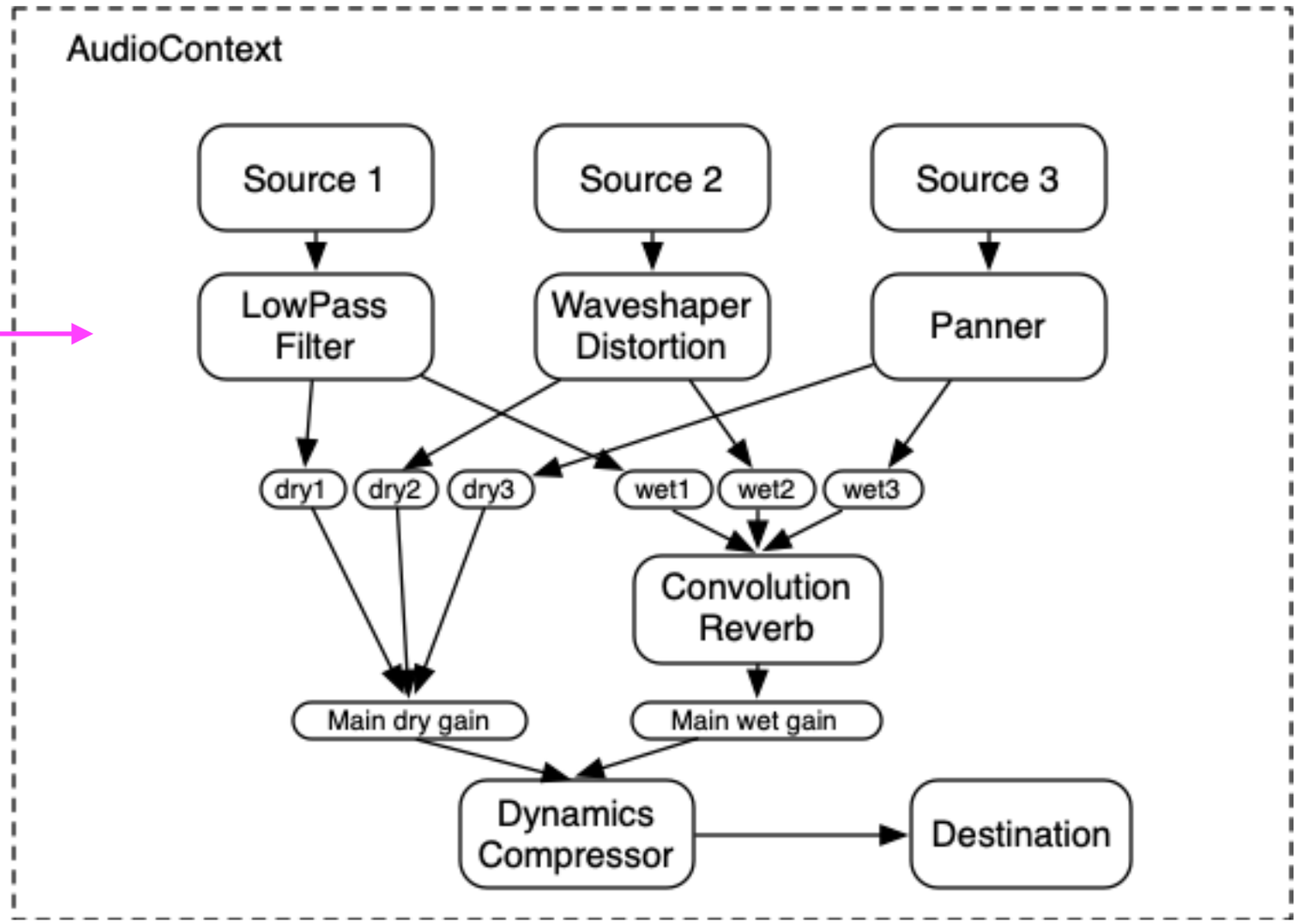
Nodes can be of different types, for example: *source nodes*, *effect nodes*, *parameter nodes*, *destination nodes*, *analyzer nodes*, etc..



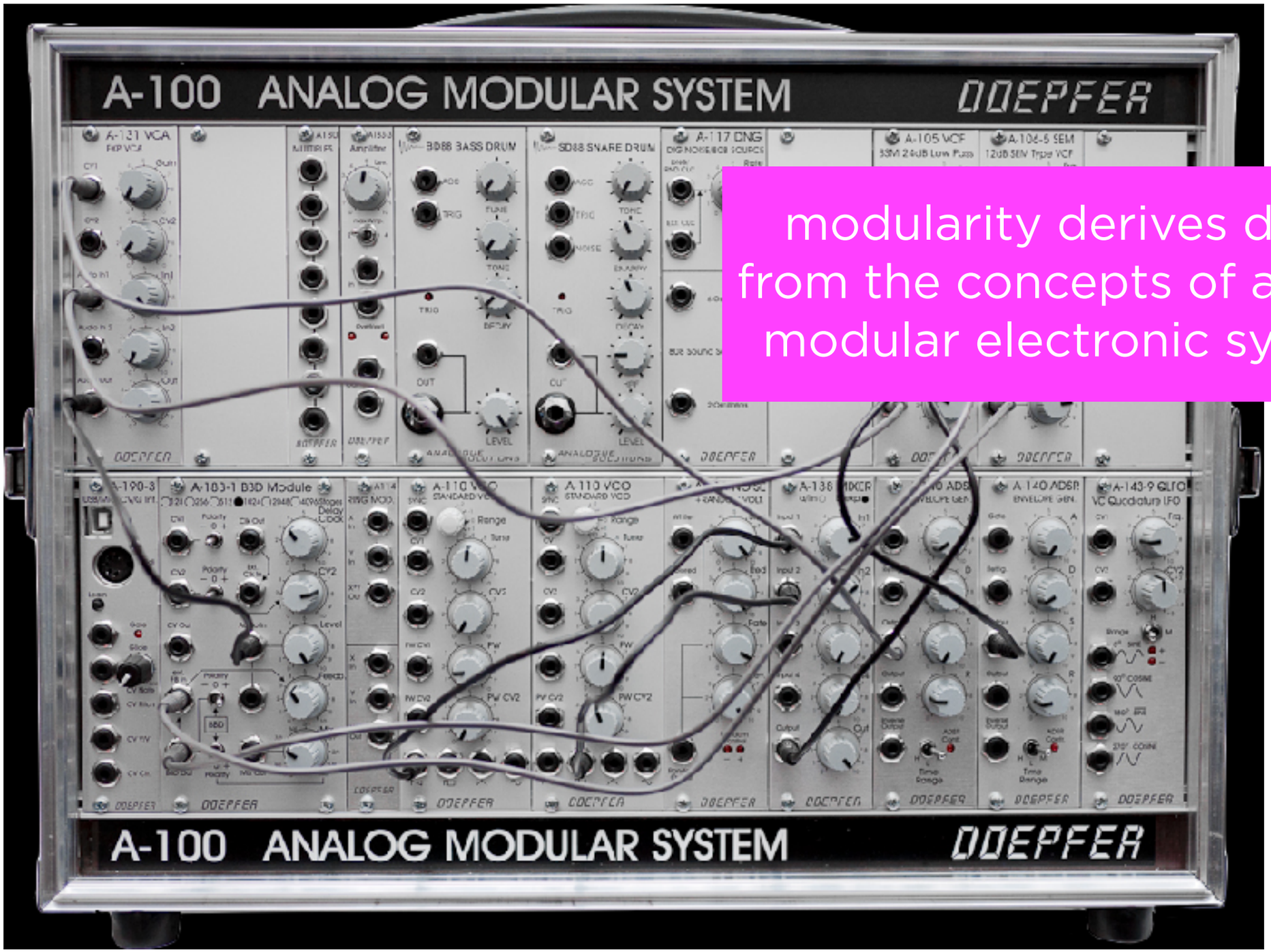
3.2 MODULAR DESIGN

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

example of the flow chart of a possible web audio graph



3.2 MODULAR DESIGN



modularity derives directly from the concepts of analogue modular electronic synthesis

3.3 EXAMPLE OF WORKFLOW

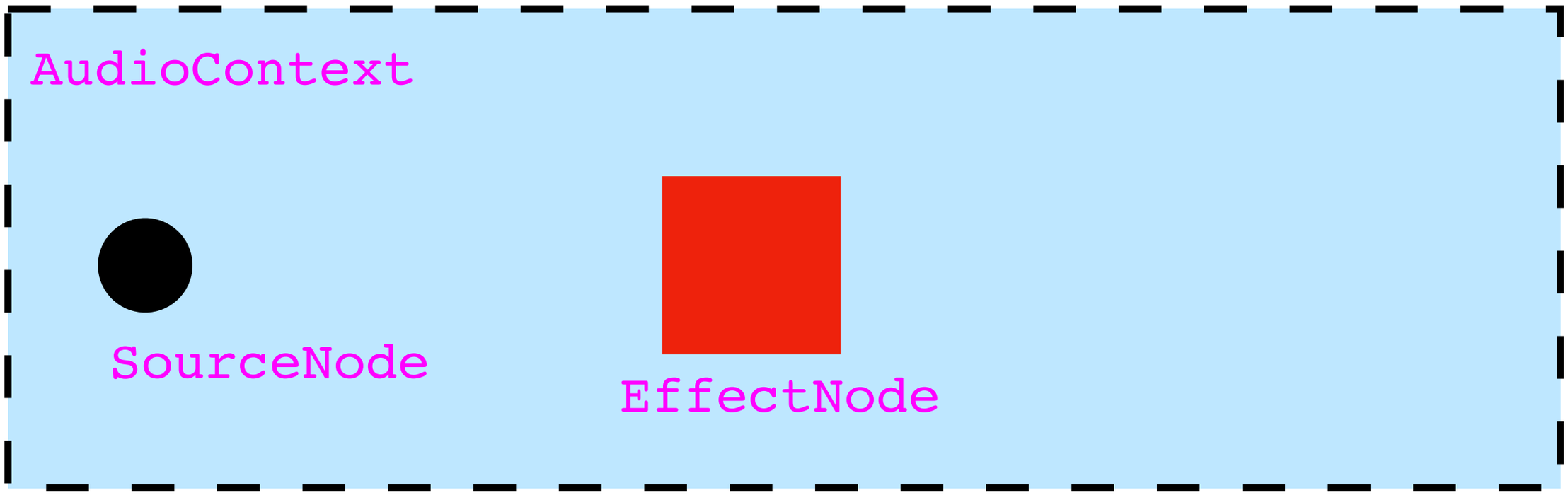
1. **Create an audio context**: an `AudioContext` is a kind of container for `AudioNode` objects, which allow different kind of audio sources and processing.

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).

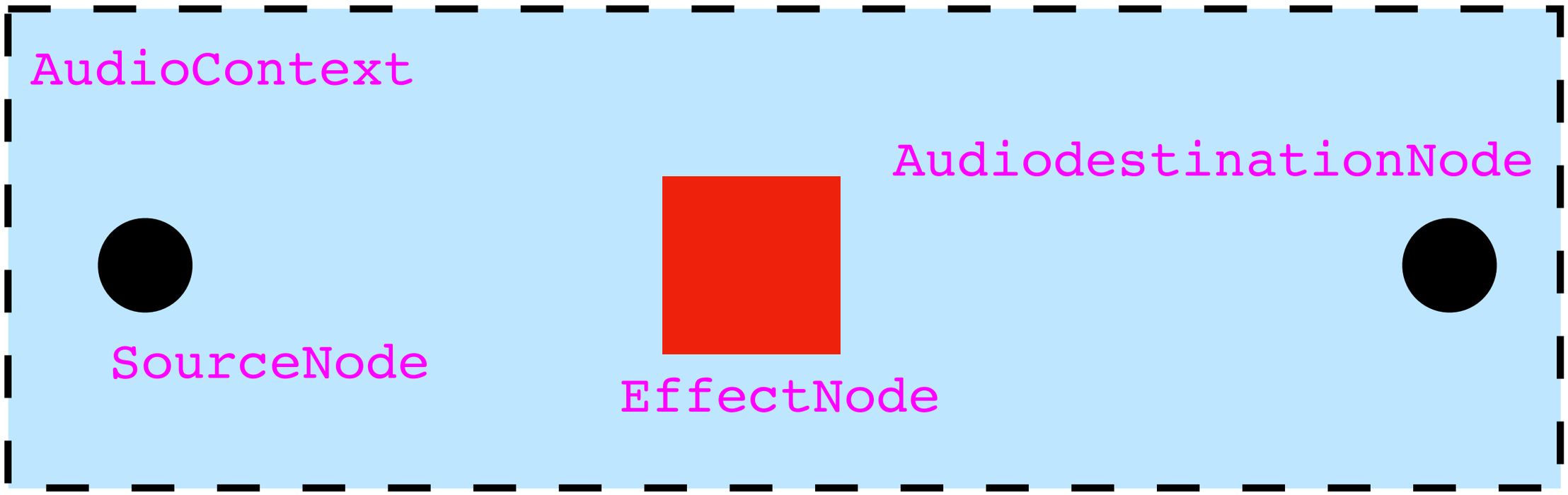
3.3 EXAMPLE OF WORKFLOW

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



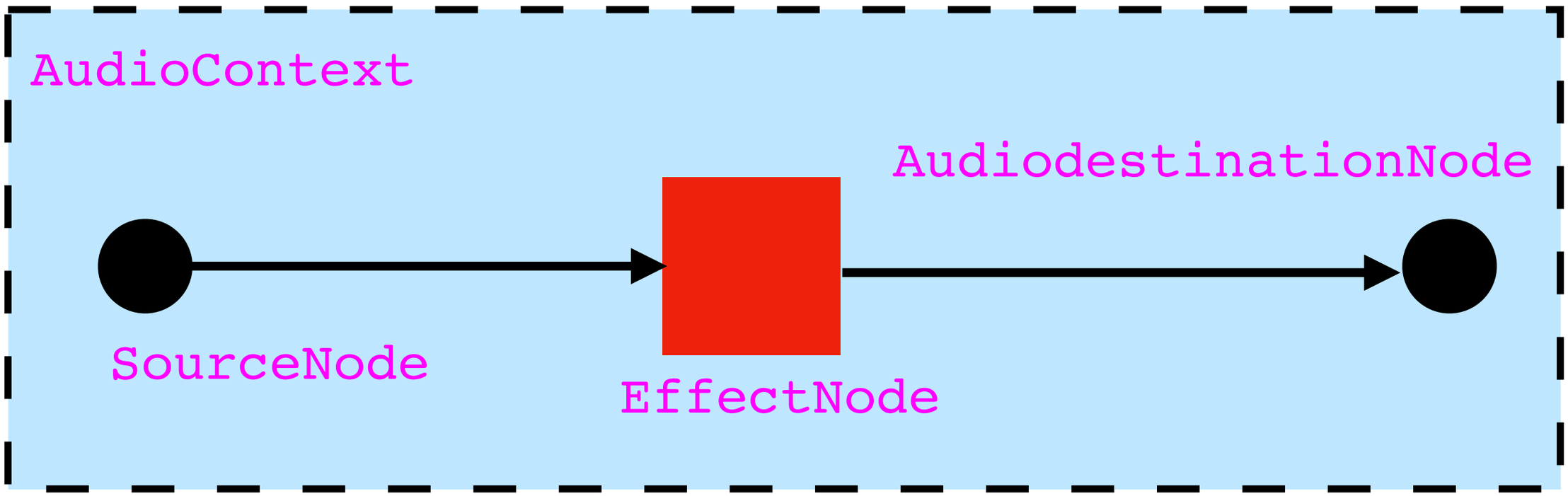
3.3 EXAMPLE OF WORKFLOW

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.



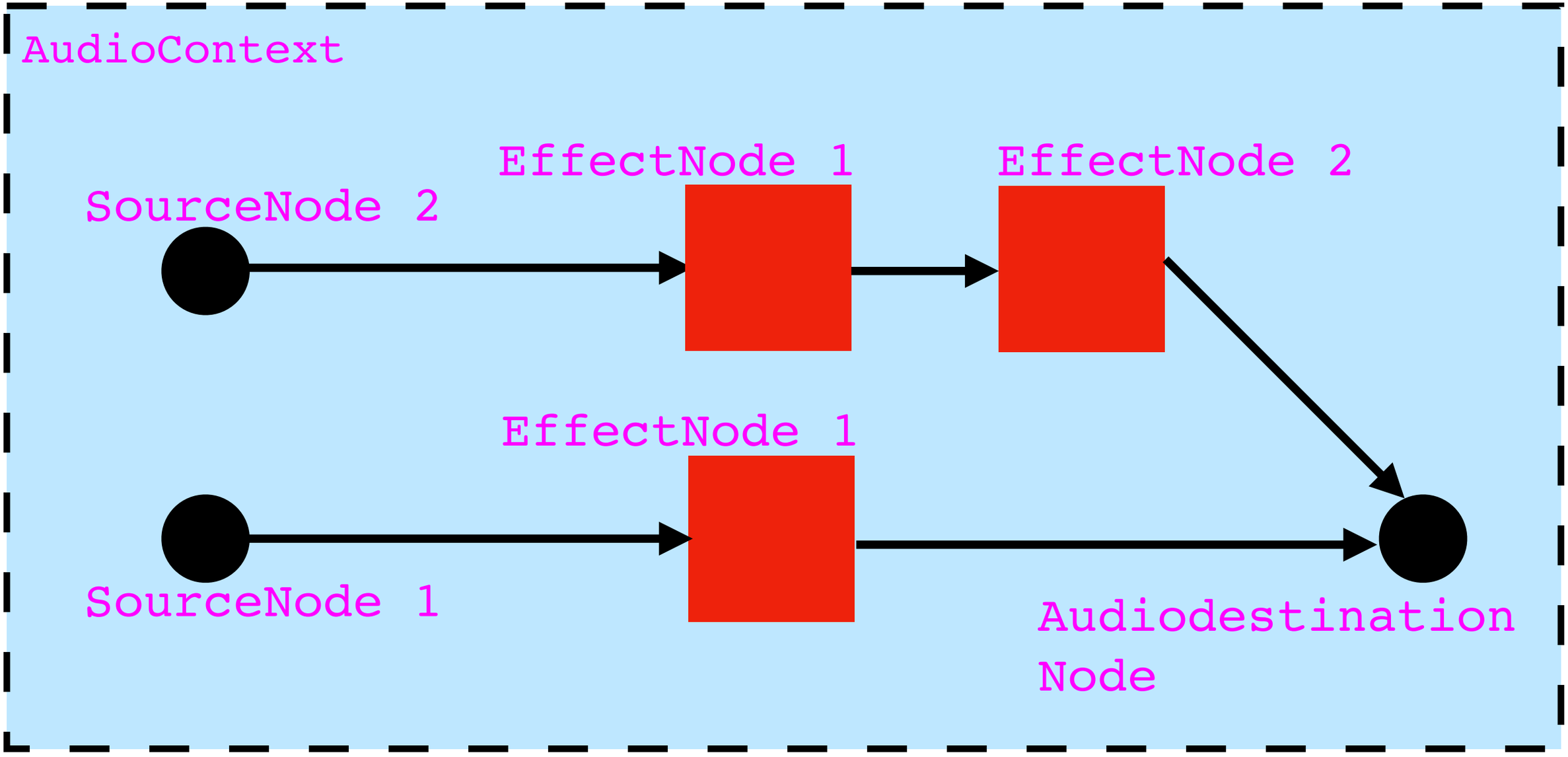
3.3 EXAMPLE OF WORKFLOW

5. **Connect** the **sources** to the **effects** and the **effects** to the **destination** (creating a chain).



3.3 EXAMPLE OF WORKFLOW

It is possible to create any kind of routing graph.



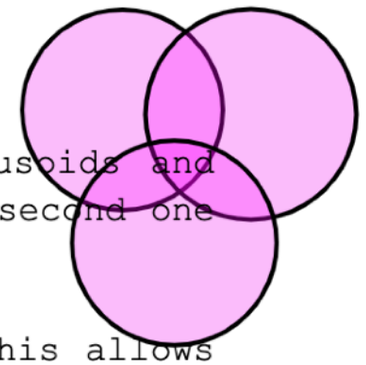
4. WHAT TOOLS WE NEED

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

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

The **y-axis** represents the amplitude range. The **amplitude** is modulated by a **triangular LFO** (Low Frequency Oscillator), with 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** between them. The first Dot becomes the modulator of the second one (the carrier).

It is possible to create a **chain of modulation**: each carrier can become a modulator. This allows you to create complex spectra, 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)**

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 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

Each **element** tells the Browser how to display a certain type of content.

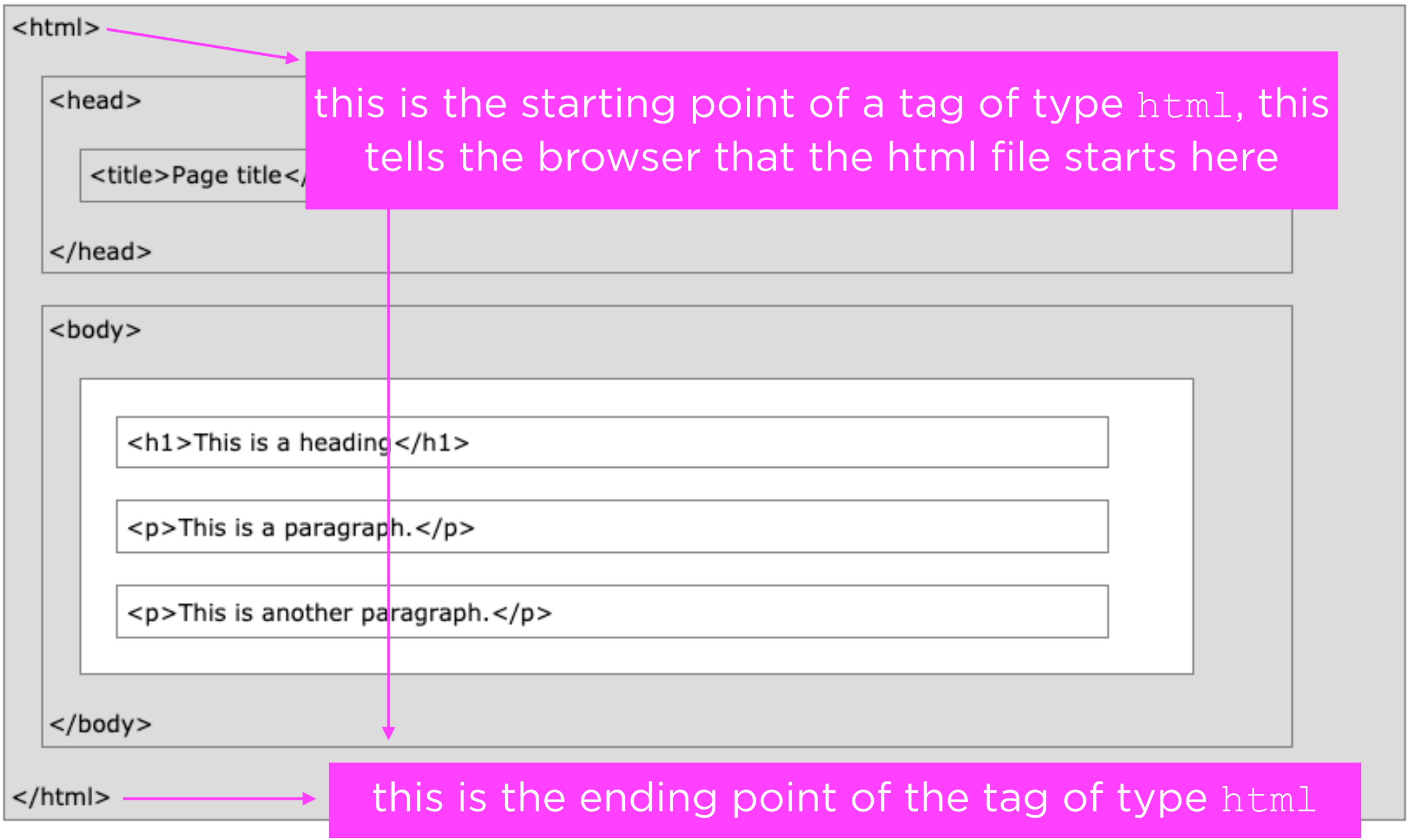
For each **content** of a web page there is a specific **element** (eg: the element for the images, the one for the text, the one for the hyperlinks, for the buttons, etc.).

Each element has a *tagName* that is unique.

The element starts with the syntax
<tagName>

The element ends with the syntax
</tagName>

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 the html file starts here
this is the ending point of the tag of type 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>
```

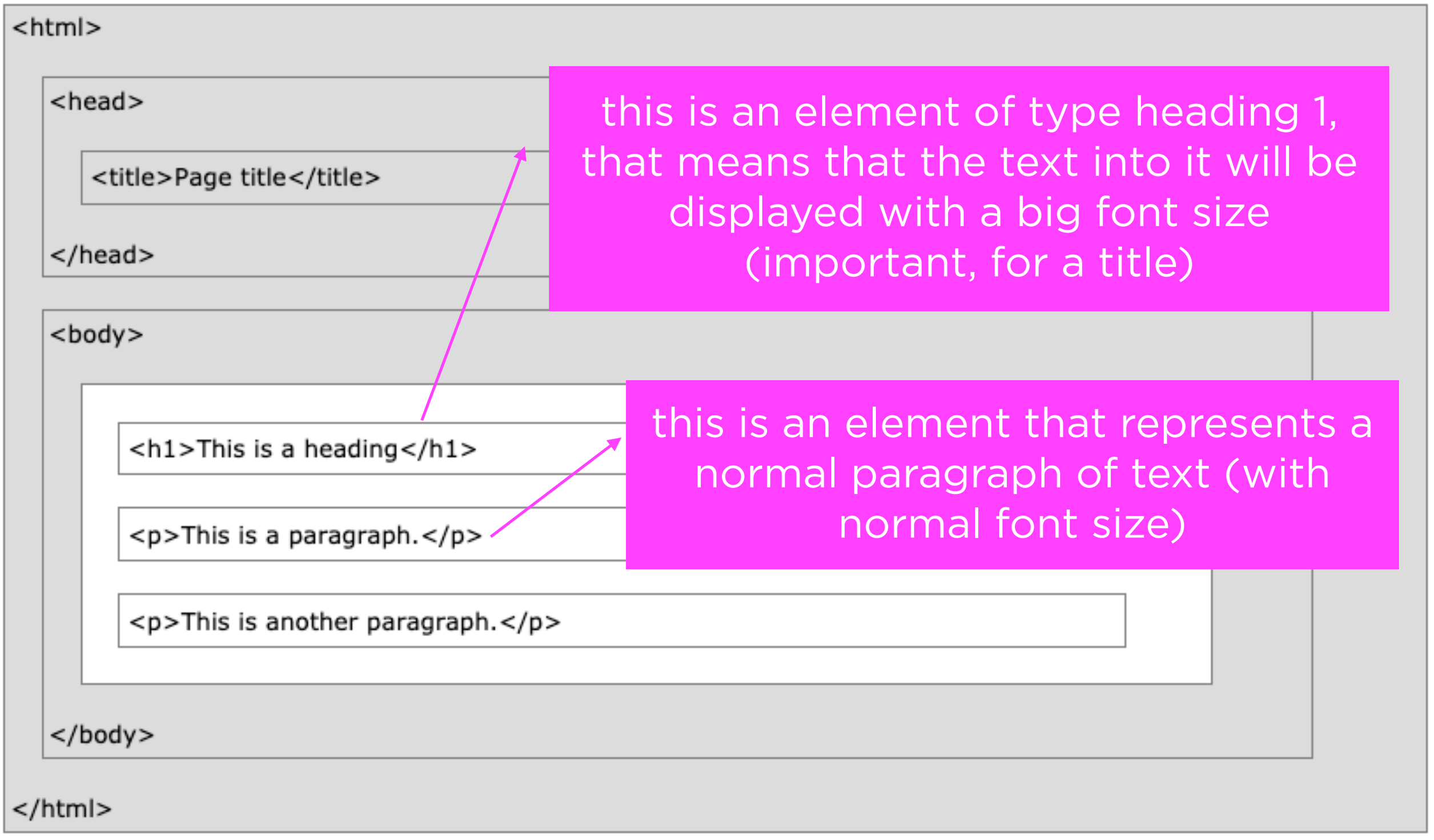
this is the tag `head`, which contains meta-informations (like the title of the page itself)

the `title` is itself a type of tags

the `body` contains the elements which will be shown to the user on the window

An element may contain another element (here the `html` element contains the `head` element and the `body` element)

4.2 HTML


<html>
<head>
<title>Page title</title>
</head>
<body>
<h1>This is a heading</h1>
This is a paragraph.
This is another paragraph.
</body>
</html>
this is an element of type heading 1, that means that the text into it will be displayed with a big font size (important, for a title)
this is an element that represents a normal paragraph of text (with normal font size)

Let's open the file *index.html*

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 murcia2021.js

`b` is the tag for the font style bold

```
1  <!DOCTYPE html> <!-- declaration defines this document to be HTML5 -->
2
3  <html lang="en"> <!-- html tag is the root element of an HTML page -->
4
5    <head> <!-- head tag contains meta information about the document -->
6
7      <link rel="stylesheet" type="text/css" href="css/murcia2021.css"> <!-- include css file -->
8
       <title>
         web audio Murcia 2021
       </title> <!-- title tag specifies a title for the document -->

       <script src="lib/p5.min.js"></script> <!-- load the libraries we need --
       <script src="js/murcia2021.js"></script> <!-- load the javascript file -

     </head>

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

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

       <p> <!-- p      defines a defines a paragraph -->
         <a href="http://www.csmmurcia.com/" target="_blank" > Conservatorio Superior de Música de Murcia 2021      </a>
         | <a href="http://www.albertobarberis.it/" target="_blank" > Alberto Barberis </a>
       </p>

29     <div id="infoText"> <!-- div tag defines a division or a section in an
30
31       Dot <b>Drone Generator</b> is a drone generator which allows you to
32       <b      <br>
33       lick on the window to generate a <b>Dot</b>, a sinusoidal wave with
34       <br><br>
35       The <b>y-axis</b> represents the amplitude range.
36       The <b>amplitude</b> is modulated by a <b>triangular LFO</b> (Low Frequency Oscillator), with random frequency.
         <br><br>
         The <b>x-axis</b> 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 <b>Frequency Modulation</b>
         between them. The first Dot becomes the modulator of the
42       <br><br>
43       It is possible to create a <b>chain of modulation</b>: e
44       This allows you to create complex spectra, to the point
45       <br><br>
46       Click on an existing circle to <b>delete</b> it or to delete the modulation chain of which it is part.
47
48     </div>
49   </body>
50 </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 an a element (anchor element) which creates a hyperlink to web pages

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 open the file *index.html*

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

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

  <head> <!-- head tag contains meta information abou the documer

    <link rel="stylesheet" type="text/css" href="css/murcia2021.cs

    <title>
      web audio Murcia 2021
    </title> <!-- title tag specifies a title for the document -->

    <script src="lib/p5.min.js"></script> <!-- load the libraries
    <script src="js/murcia2021.js"></script> <!-- load the javascr

  </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="http://www.csmmurcia.com/" target="_blank" >
      | <a href="http://www.albertobarberis.it/" target="_bl
    </p>

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

      Dot <b>Drone Generator</b> is a drone genera                    tly on your browser.
      <br><br>
      Click on the window to generate a <b>Dot</b>
      <br><br>
      The <b>y-axis</b> represents the amplitude range.
      The <b>amplitude</b> is modulated by a <b>triangular LFO</b> (Low Frequency Oscillator), with random frequency.
      <br><br>
      The <b>x-axis</b> represents the frequency range.
      <br><br>
      Press 'L' and then Click+Drag from an e                    ency Modulation</b>
      between them. The first Dot becomes the
      <br><br>
      It is possible to create a <b>chain of
      This allows you to create complex spectra, to the point of creating very noisy sounds!
      <br><br>
      Click on an existing circle to <b>delete</b> it or to delete the modulation chain of which it is part.

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

each `tag` can contain different **attributes.** HTML attributes provide additional information about HTML elements.

An attribute follows the syntax:

```
attributeName = "information"
```

`src` is the attribute to specify the source path of a javascript file

`href` is the attribute to specify the URL of an hyperlink

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

Dot Drone Generator

Conservatorio Superior de Música de Murcia 2021 | Alberto Barberis

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

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

The **y-axis** represents the amplitude range. The **amplitude** is modulated by a **triangular LFO** (Low Frequency Oscillator), with 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** between them. The first Dot becomes the modulator of the second one (the carrier).

It is possible to create a **chain of modulation**: each carrier can become a modulator. This allows you to create complex spectra, 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.

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

If you click on the page nothing happens yet because the **javascript** file that will manage the user interaction and the sound synthesis methods does not exist (yet)!

4.3 CSS

WHAT IS IT FOR?

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/style** (CSS) and **content/structure** (HTML).

HTML (contents and structure)

CSS (style and presentation)

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

4.3 CSS

STYLE RULES AND SPECIFICITY

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.

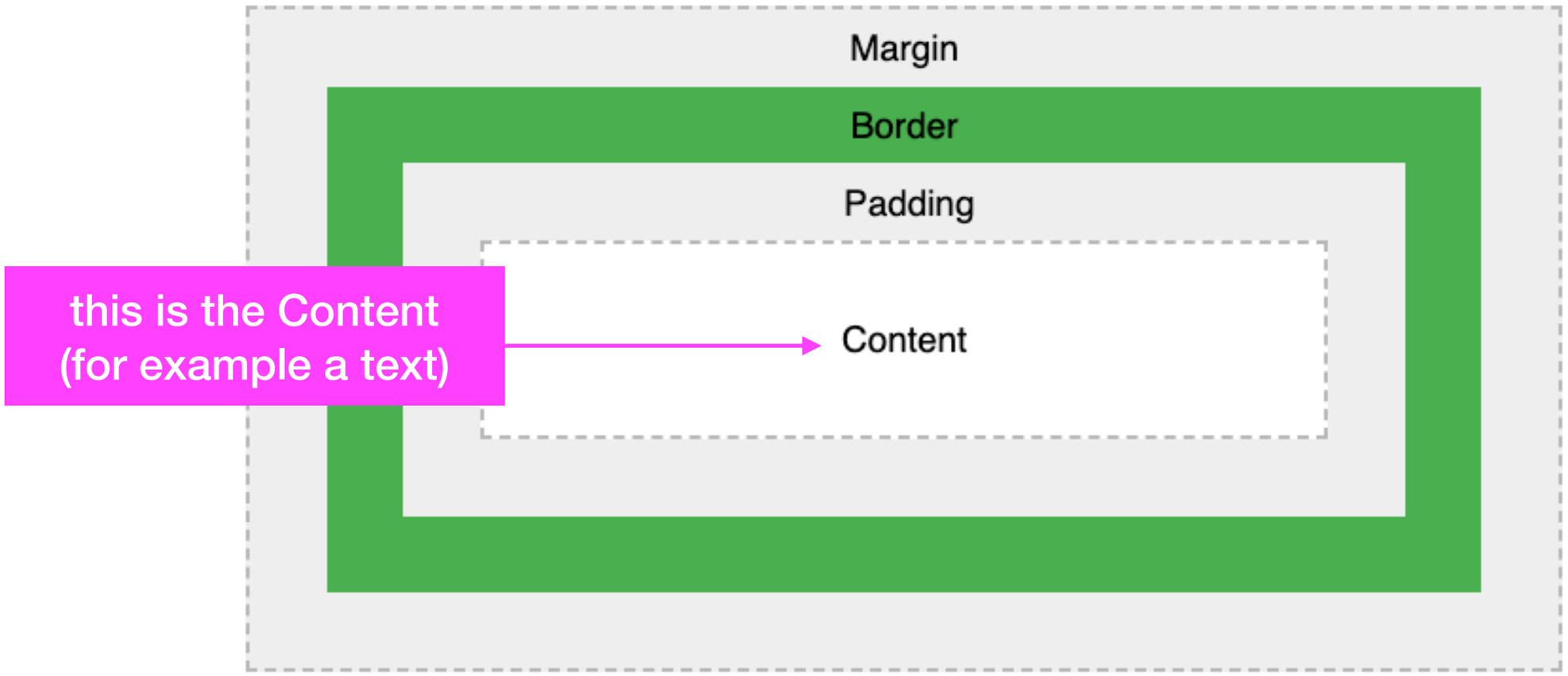
Some rules have a **higher priority** (specificity) than others and **this priority depends on the manner in which the rules are defined**.

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

4.3 CSS

THE BOX MODEL

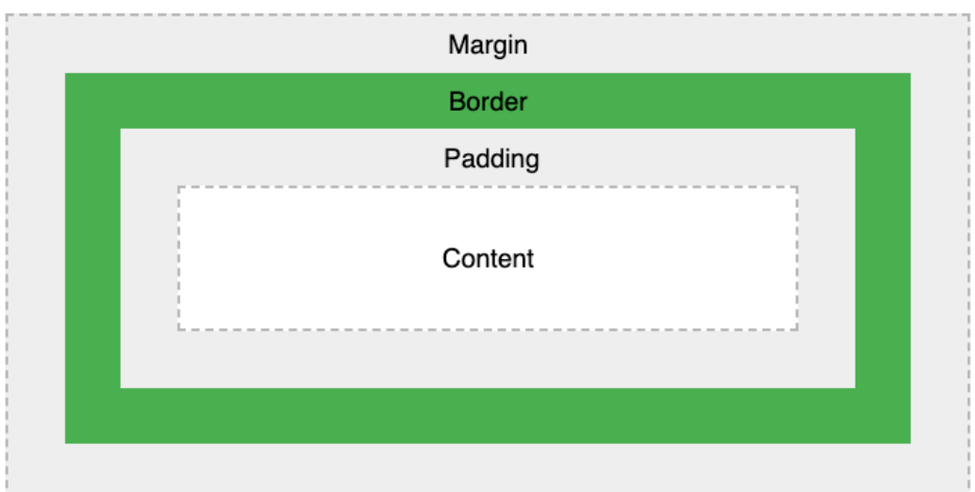
All HTML elements can be considered as **boxes**. The CSS box model is essentially a box that wraps around every HTML element.



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

4.3 CSS

THE BOX MODEL



- **Content** : 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 the content (can be *visible*);
- **Margin** - clears an area outside the border. The margin is *transparent*.

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

Let's open the file *index.html*

```
<!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/murcia2021.css"> <!-- include css file -->

    <title>
      web audio Murcia 2021
    </title> <!-- title tag specifies a title for the document -->

    <script src="lib/p5.min.js"></script> <!-- load the libraries we need -->
    <script src="js/murcia2021.js"></script> <!-- load the javascript file -->

  </head>

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

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

    <p> <!-- p tag defines a defines a paragraph -->
      <a href="http://www.csmmurcia.com/" target="_blank" > Conservatorio Superior de Música de Murcia 2021      </a>
      | <a href="http://www.albertobarberis.it/" target="_blank" > Alberto Barberis </a>
    </p>

    <div id="infoText"> <!-- div tag defines a division or a section in an HTML document. it is a general element -->

      Dot <b>Drone Generator</b> is a drone gener                                                    irectly on your browser.
      <br><br>
      Click on the window to generate a <b>Dot</b
      <br><br>
      The <b>y-axis</b> represents the amplitude
      The <b>amplitude</b> is modulated by a <b>t                                                    requency.
      <br><br>
      The <b>x-axis</b> represents the frequency
      <br><br>
      Press 'L' and then Click+Drag from an exist                                                    <b>Frequency Modulation</b>
      between them. The first Dot becomes the mod
      <br><br>
      It is possible to create a <b>chain of modu
      This allows you to create complex spectra,
      <br><br>
      Click on an existing circle to <b>delete</b> it or to delete the modulation chain of which it is part.

    </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

`rel` attribute specifies the relationship between the documents

`type` specifies the type of document

`href` sets the file name

Let's open the file css.*html*

4.3 CSS

THE SELECTORS

The CSS language is made of different **selectors** and **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 they are defined with the syntax : **`properties: value;`**

For example here we say to the `body` element to have a `margin` of 10px, to use the `text-family` Courier New, and that the text it contains can not be selected (`user-select` property).

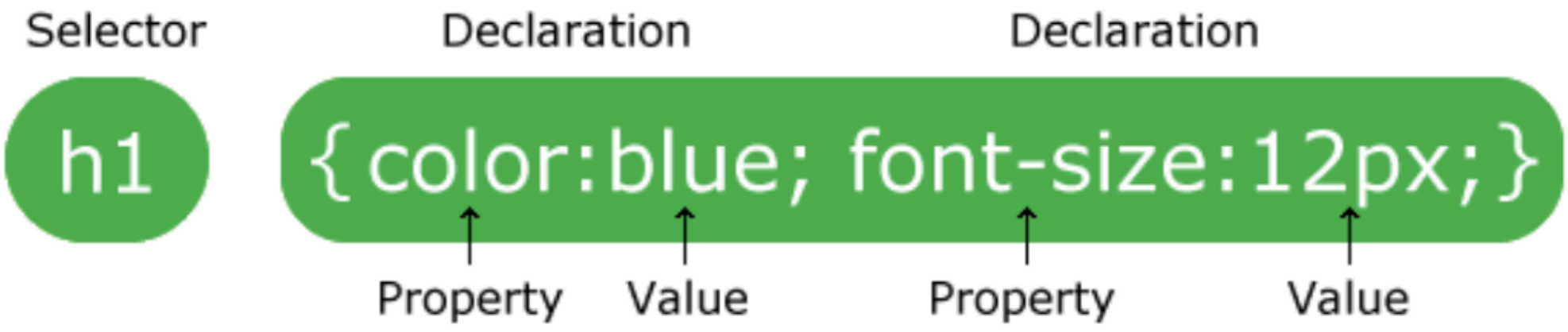
```
1   body { /* selector body */
2       margin: 0px;
3       font-family: 'Courier New';
4       user-select: none; /* user can not select the text */
5   }
```

This is aSELECTOR: it specifies which elements the rules refer to

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

4.3 CSS

THE SELECTORS



A CSS rule-set consists of:

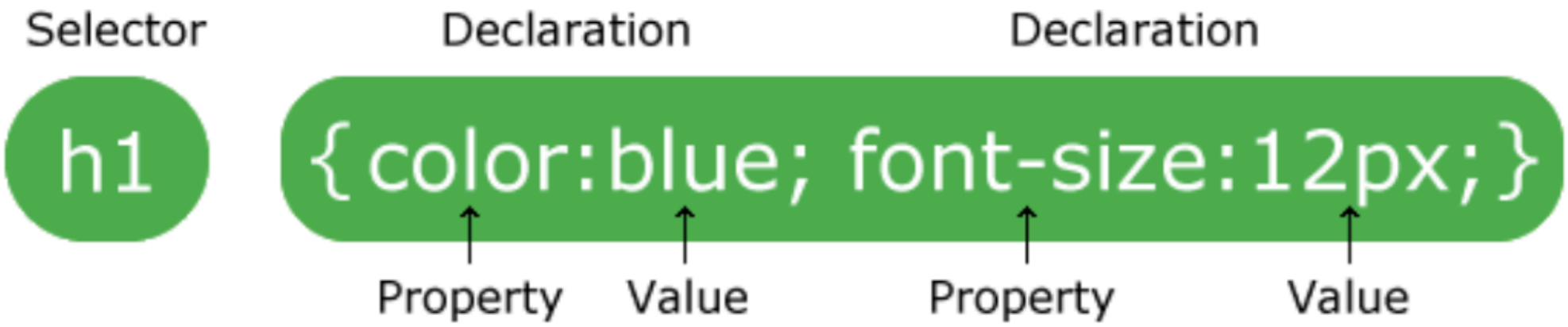
1. a **selector**, that points to the HTML element you want to style.

2. a **declaration block** surrounded by curly braces, that contains one or more declarations separated by semicolons. Each declaration includes a **CSS property name** and a **value**, separated by a colon.

4.3 CSS

THE SELECTORS

There are different types of selectors!



Selector	Example	Example description
.class	.intro	Selects all elements with class="intro"
.class1.class2	.name1.name2	Selects all elements with both *name1* and *name2* set within its class attribute
.class1 .class2	.name1 .name2	Selects all elements with *name2* that is a descendant of an element with *name1*
#id	#firstname	Selects the element with id="firstname"
*	*	Selects all elements
element	p	Selects all <p> elements
element.class	p.intro	Selects all <p> elements with class="intro"

some examples of selectors

https://www.w3schools.com/cssref/css_selectors.asp

```css
body { /* selector body */
    margin: 0px;
    font-family: 'Courier New';
    user-select: none; /* user can not select the text */
}
h1 { /* selector h1 */
    margin: 0px;
    padding: 10px;
    background-color: rgb(255, 79, 79);
    color: white;
}
p { /* selector paragraph */
    margin: 0px;
    padding: 10px;
    color: rgb(0, 0, 0);
    border-bottom: solid;
    border-width: 1px;
}
#infoText { /* selector element with ID infoText */
    position: fixed;
    margin-top: 50px;
    margin-left: 100px;
    margin-right: 100px;
    text-align: justify;
    line-height: 130%;
}
a { /* selector element a */
    text-decoration: none;
    color: black;
}
a:hover { /* selector element a when mouse is over */
    color: rgb(255, 79, 79);
}
```

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 set the `position` fixed.

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:

```
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
32      color: rgb(255, 79, 79);
33  }
```

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

You should check the other rules and properties by yourself: https://www.w3schools.com/cssref/css_selectors.asp

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** (**FM**).

We speak of **modulation synthesis** when at least one of the 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)

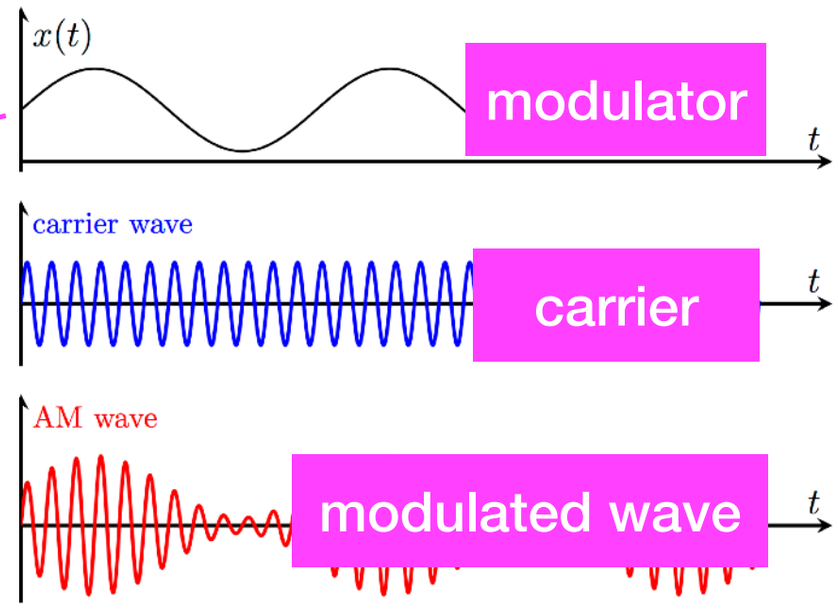
To obtain a **tremolo** we implement an AM.

A sinusoid (**carrier**) will be amplitude modulate by an other oscillator (a triangular wave **modulator**) that oscillates at Low Frequency (for this reason, we call it 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.

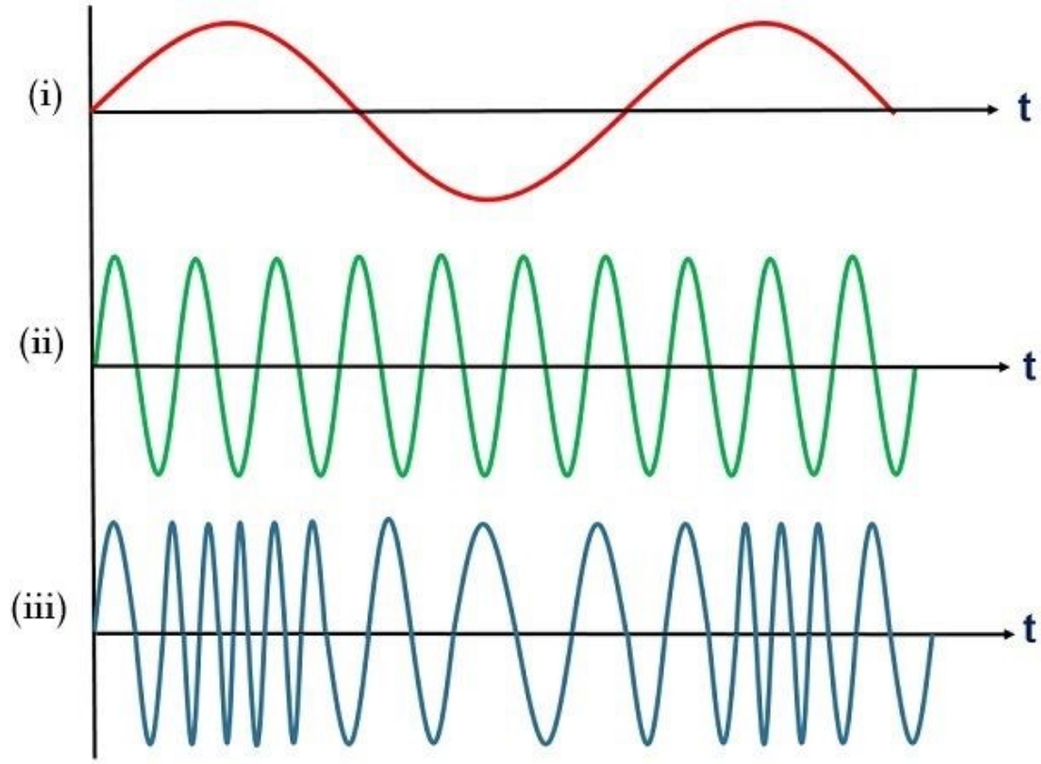
In our case the modulator will be a triangle waveform and not a sinusoid

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



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.



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

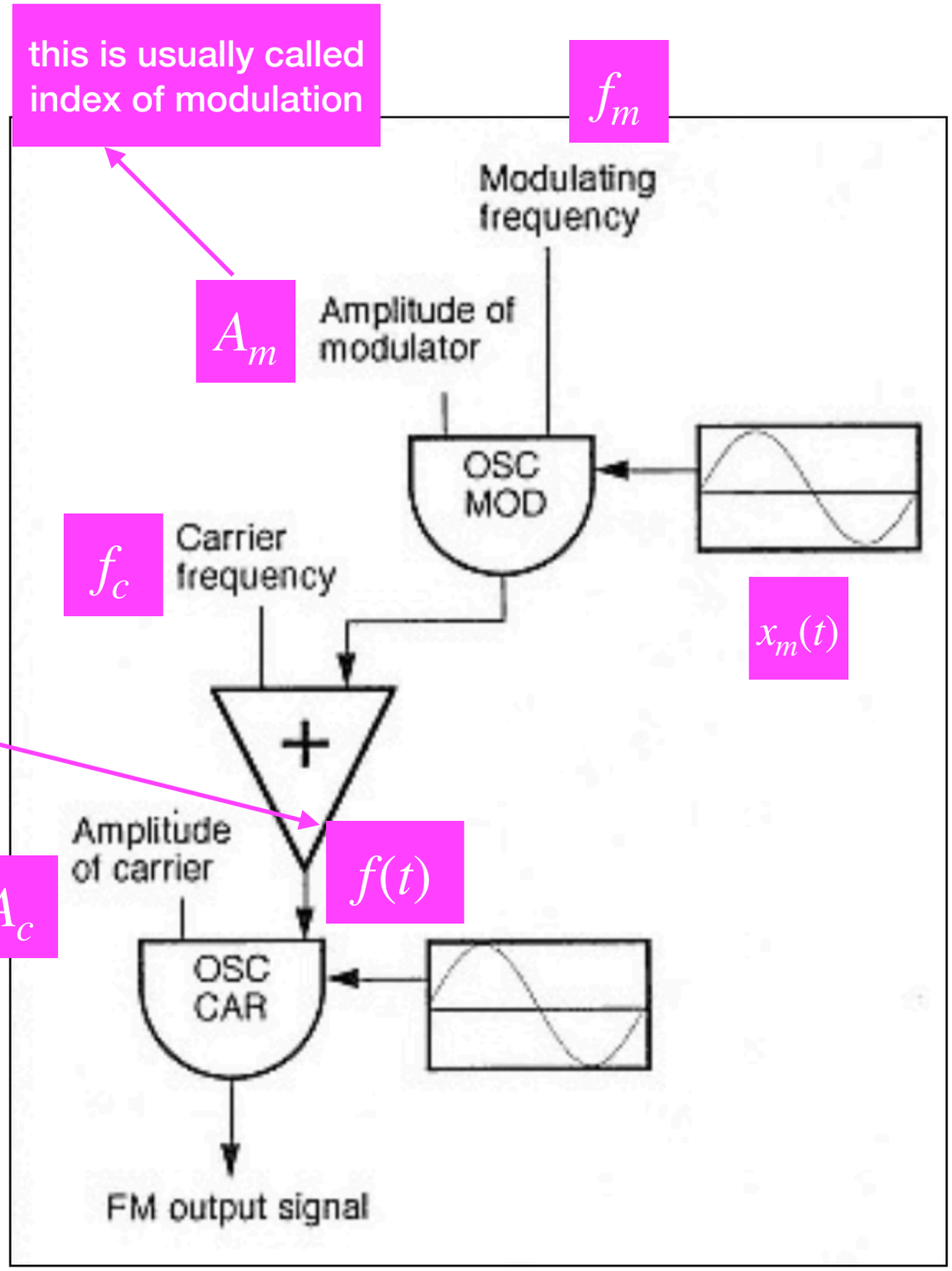
4.5 FM SYNTHESIS

In the **linear FM** the frequency of the original signal is no longer constant but it 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



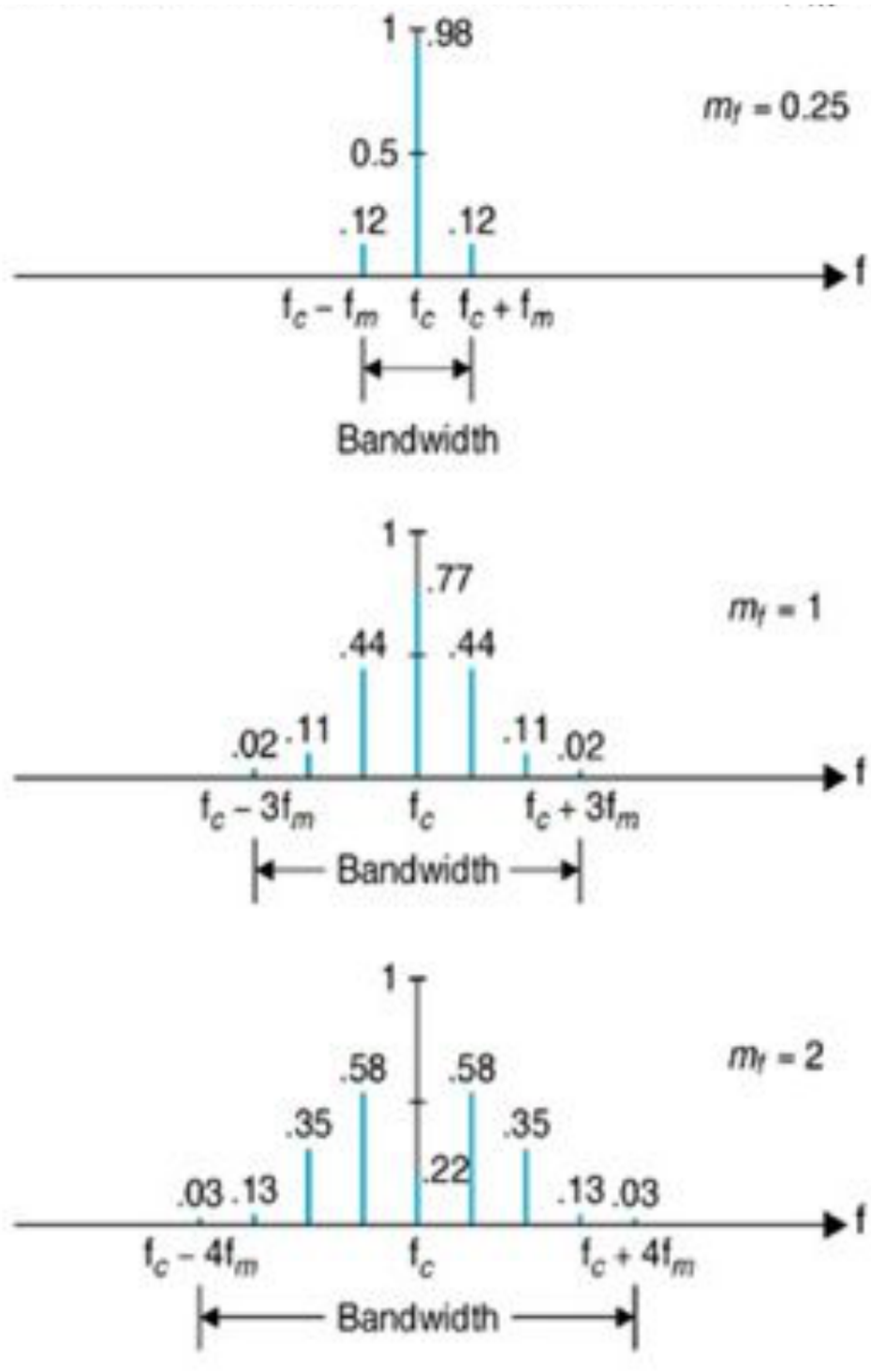
4.5 FM SYNTHESIS

The FM synthesis generates **complex spectra**, starting from simple 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