

# Web Audio

# Introduction

# Technologies web - *introduction*

## *Internet*

Ensemble de protocoles de transfert de données (TCP, IP, ...) pour l'interconnection de réseaux

**1972** - Première démonstration officielle d'*ARPANET*  
*envoi d'un message entre UCLA et Stanford*

**1983** - *ARPANET* est officiellement renommé *Internet*

## *World Wide Web*

Système de partage d'informations hypertexte fonctionnant sur Internet

**1989-1992** - Développement au CERN par T. Berners Lee et *Robert Caillau (logiciels et protocoles)*

**1993** - Le CERN ouvre les technologies au domaine public

# Basics

**HTTP**

*HyperText Transfert Protocol*

**URL**

*Uniform Resource Locator*

**HTML**

*HyperText Markup Language*



# Languages

*html / css / javascript*

```
(function hide() {  
    $list.forEach(($el, index) => {  
        $el.classList.remove('hide');  
  
        if (index === counter) {  
            $el.classList.add('hide');  
        }  
    });  
  
    counter = (counter += 1) % $list.length;  
    timeoutId = setTimeout(hide, 1000);  
}());
```

<http://127.0.0.1:3000>

# Web as a Creative Platform & Web Audio API

# web as creative platform

## *ubiquity*

almost every device implements web standards

## *interactive multimedia*

HTML5/CSS, Web GL, Canvas, Web Audio API,  
DeviceMotion/Orientation, Geolocation

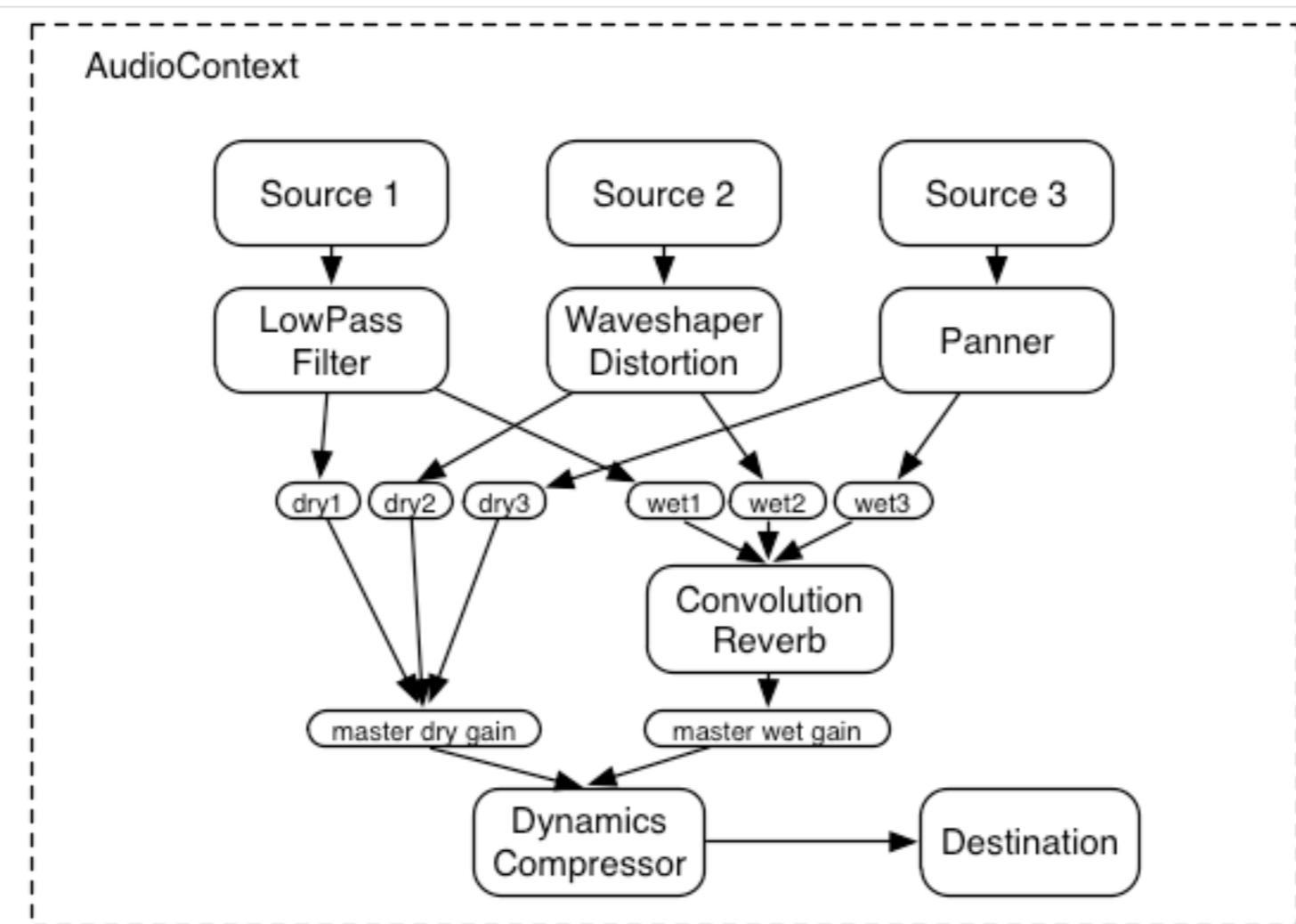
## *networking*

HTTP, WebSockets, WebRTC

## *rapid prototyping & interoperability* (using Node.js)

very rapid development / deployment cycles

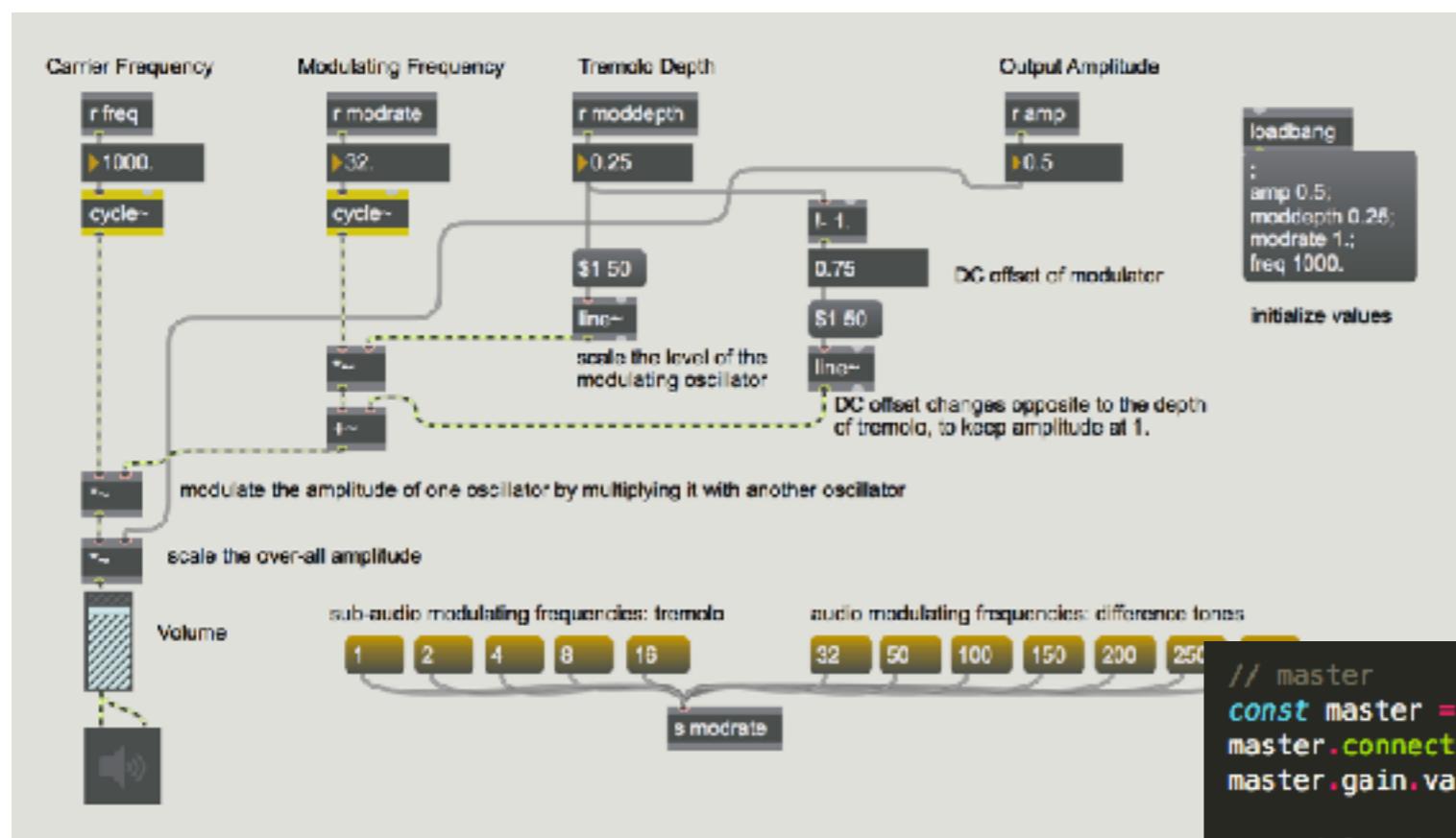
# WebAudio



*specification*

<https://webaudio.github.io/web-audio-api/>

# WebAudio



```
// master
const master = audioContext.createGain();
master.connect(audioContext.destination);
master.gain.value = 0; // default to muted

// modulated amplitude
const amplitude = audioContext.createGain();
amplitude.connect(master);
amplitude.gain.value = 1 - defaultDepth;
amplitude.gain.setValueAtTime(1 - defaultDepth, audioContext.currentTime);

// carrier
const carrier = audioContext.createOscillator();
carrier.connect(amplitude);
carrier.frequency.value = 1000;

// modulation
const depth = audioContext.createGain();
depth.connect(amplitude.gain);
depth.gain.value = defaultDepth
depth.gain.setValueAtTime(defaultDepth, audioContext.currentTime);

const mod = audioContext.createOscillator();
mod.frequency.value = 1;
mod.connect(depth);

carrier.start(audioContext.currentTime);
mod.start(audioContext.currentTime);
```

# TP 1 - Basics

`./TP/1-04-trigger-sine-with-env`

# Examples

# Librairies

# Waves

## WebAudio Visualisation Edition and Synthesis

<https://github.com/wavesjs/>

# waves-ui

<http://wavesjs.github.io/waves-ui/examples/layer-waveform.html>

<http://wavesjs.github.io/waves-ui/examples/layer-axis.html>

<http://wavesjs.github.io/waves-ui/examples/layer-segment.html>

<http://wavesjs.github.io/waves-ui/examples/states-zoom.html>

# waves-audio

<https://rawgit.com/wavesjs/waves-audio/master/examples/player-engine/index.html>

<https://rawgit.com/wavesjs/waves-audio/master/examples/granular-engine/index.html>

# waves-lfo

<https://cdn.rawgit.com/wavesjs/waves-lfo/master/examples/sink-vu-meter-display/index.html>

<https://cdn.rawgit.com/wavesjs/waves-lfo/master/examples/sink-waveform-display/index.html>

<https://cdn.rawgit.com/wavesjs/waves-lfo/master/examples/operator-segmenter/index.html>

<https://cdn.rawgit.com/wavesjs/waves-lfo/master/examples/mosaicking/index.html>

# démonstrateurs

<https://wave.ircam.fr/demo/phoenix-1901/>

<https://wave.ircam.fr/demo/bachothèque/>

<https://wave.ircam.fr/demo/leroux-voirex/>

# Artistic projects

# Collective Sound Checks

2014



# 88 Fingers

*AudioMostly 2017*

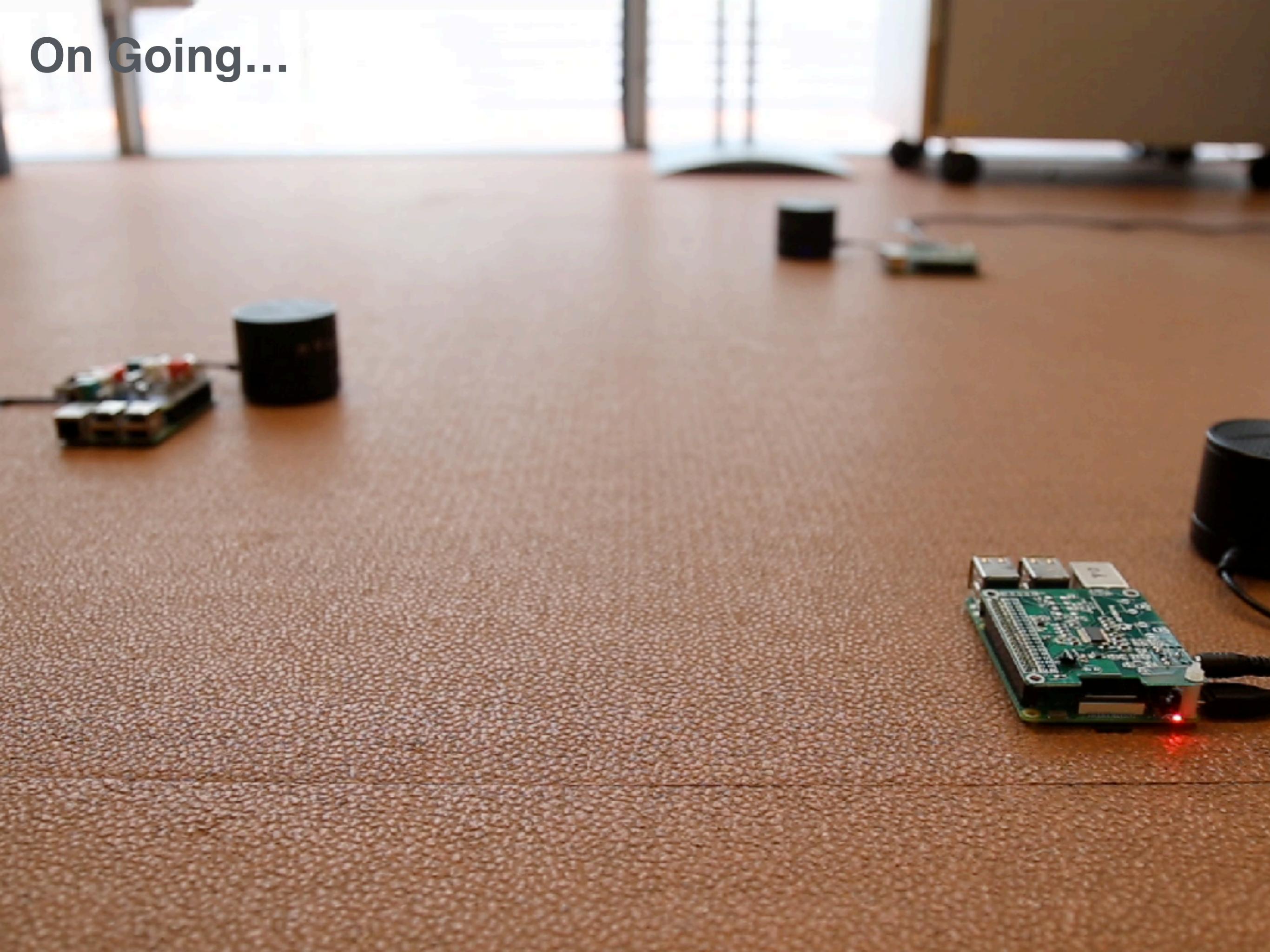


# GrainField

Audio Mostly, London, 2017



On Going...



# *Biotope* - Jean-Luc Hervé

2019 - exposition *La fabrique du vivant* - CGP

# MISC

# *Design Patterns*

In **software engineering**, a **software design pattern** is a general, reusable solution to a commonly occurring problem within a given context in **software design**. It is not a finished design that can be transformed directly into **source** or **machine code**. Rather, it is a description or template for how to solve a problem that can be used in many different situations.

[https://en.wikipedia.org/wiki/Software\\_design\\_pattern](https://en.wikipedia.org/wiki/Software_design_pattern)

**Gang of Four:** Design Patterns: *Elements of Reusable Object-Oriented Software* (1994)

**History:** Christopher Alexander, *Notes on the Synthesis of Form* (1964)

→ creating / inventing / designing a *form* adapted to a *context*

# *Design Patterns - 2 examples*

## Singleton

Why?

Implementation

## EventEmitter / PubSub

Why?

Implementation

# Questions ?