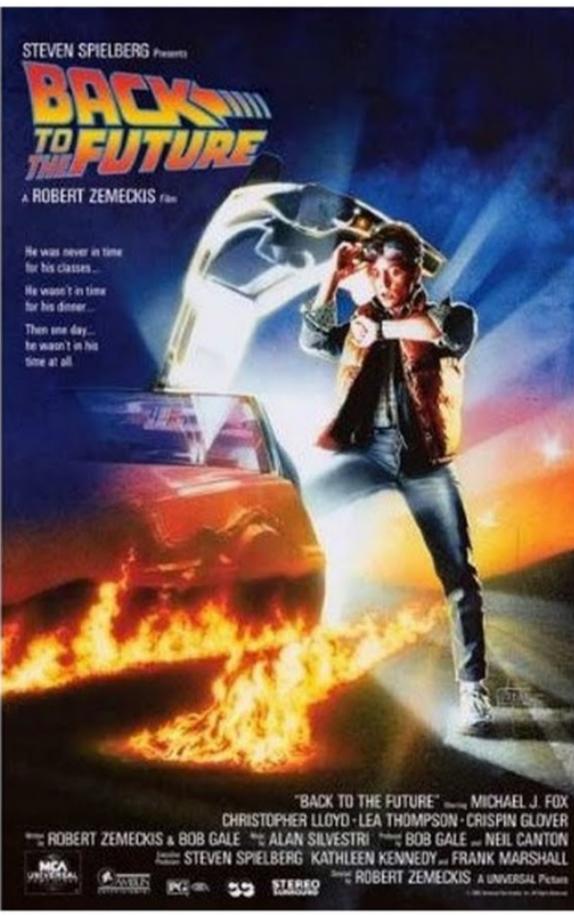
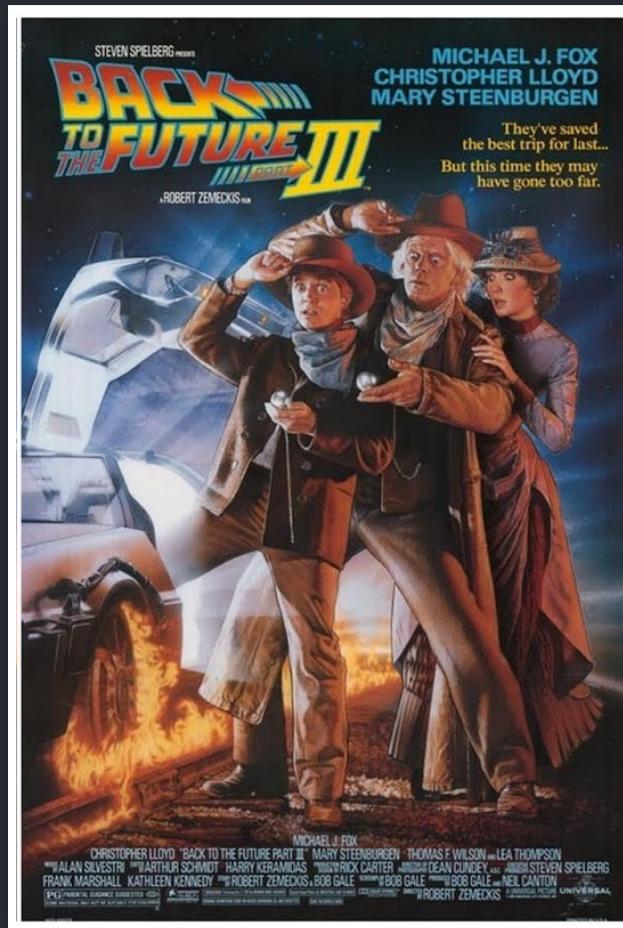


Come JavaScript ha

CAMBIATO

IL CSS

Davide Di Pumbo



The background of the image is a collage of two photographs. The top photograph shows a light-colored cat sitting on a dark surface next to a turntable with a vinyl record. The bottom photograph shows a black and white cat standing on a turntable, its front paws on the platter, with a DJ mixer and another turntable visible in the background.

LET'S START MEAOW

Parte 1

IL PASSATO



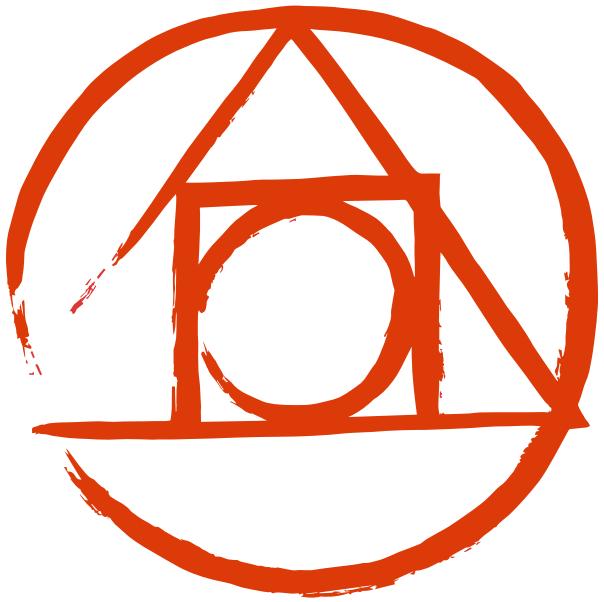
Disclaimer: family guy css gif is in there for satirical purposes, I actually love css in each of its forms

Sass

GRAZIE A SASS ABBIAMO AVUTO:

- Le variabili
- Le funzioni
- I mixin





CON SASS ERAVAMO DIPENDENTI DA RUBY



POI, UNA NOTTE DI SETTEMBRE MI SVEGLIAI

Era maggio 2009, ma volevo fare la citazione 



what the heck
do I need node
for on the
frontend?



what the he
do I need n
for on the
frontend?





JavaScript

Build Tools



il che ci porta a...

Parte 2

IL PRESENTE

Any application that can be written
in JavaScript, will eventually be
written in JavaScript.

ATWOOD'S LAW

No, non sto parlando di CSS in JS
Lo faremo presto 😊

La vera domanda è, perchè ho bisogno di JS
per scrivere del CSS?

PER RISOLVERCI QUALCHE
PROBLEMA

Problema:

SCOPING

```
.title {  
    color: red;  
}  
.title {  
    color: green;  
}  
.title {  
    color: blue;  
}
```

Lo abbiamo risolto con:

BEM, SMACSS, ECC... ECC...



GENTLEMEN'S **AGREEMENT**

**CSS
MODULES**

```
|- Accordion
  |- Accordion.html
  |- Accordion.module.css
|- BadgeStep
  |- BadgeStep.html
  |- BadgeStep.module.css
|- BlogPosts
  |- BlogPosts.html
  |- BlogPosts.module.css
```

```
/* Accordion.module.css */
```

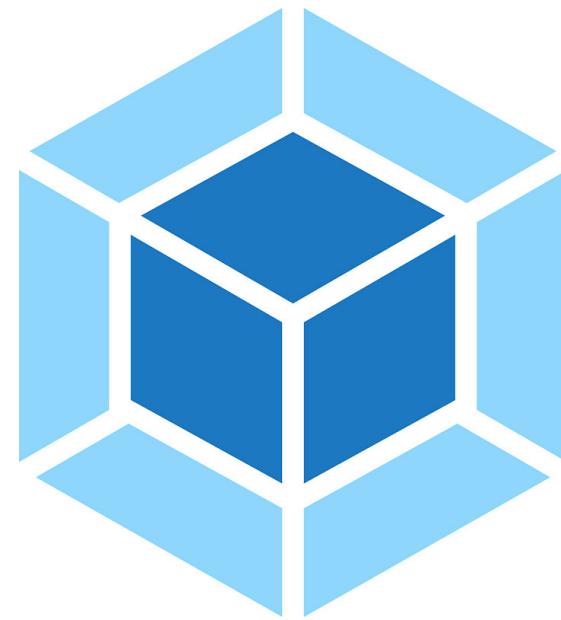
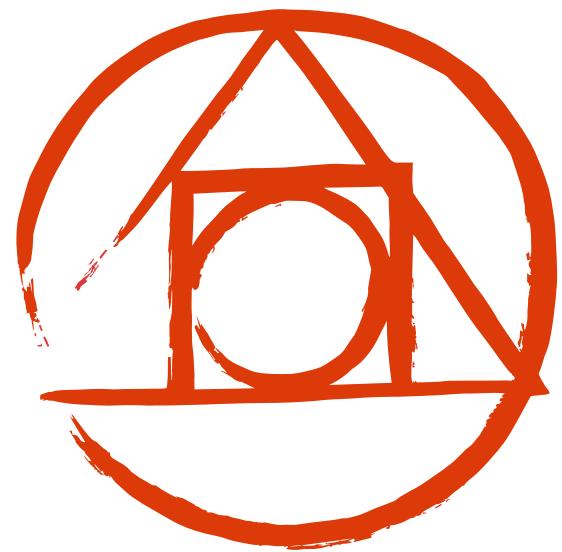
```
.title { color: green; }
```

```
/* BadgeStep.module.css */
```

```
.title { color: green; }
```

```
/* BlogPosts.module.css */
```

```
.title { color: green; }
```



```
/* Accordion.css */
.Accordion--title--axYow2l { color: green; }

/* BadgeStep.css */
.BadgeStep--title--9scas2 { color: green; }

/* BlogPosts.css */
.BlogPosts--title--Fsvloa { color: green; }
```

html

```
<h1 css-module="title">postcss-modules</h1>
```

js

```
import styles from "./component.module.css";
element.innerHTML =
`<div class="${style.title}">postcss-modules</div>`;
```

pug

```
h1(class=css.title) postcss-modules
```

h1.src-components-PageTitle----PageTitle-module---title---EAYj9.src-components-FadeOnScroll---fade-module---on0---1ERf8.src-components-FadeOnScroll---fade-module---top---2sz2---1120 x 56

Una realtà solida e affidabile

Analisi pagina Console Debugger { } Editor stili Prestazioni Memoria Rete Archiviazione

●

Cerca in HTML

```
section class="src-layouts----container-module---main---3ifBy">
<div>
  <!--react-empty: 697-->
<div class="src-components-FadeOnScroll----fade-module---isIntersecting---16byK">
  <section class="src-components-PageTitle----PageTitle-module---main---sIqcI">
    <h1 class="src-components-PageTitle----PageTitle-module---title---EAYj9...Rf8 src-components-FadeOnScroll----fade-module---top---2sz2---">Una realtà solida e affidabile</h1>
  </section>
</div>
<section class="src-pages-style----chi-siamo-module---section---3uBn2">...</section>
<section class="src-pages-style----chi-siamo-module---sectionTeam---3oZ0b">...</section>
<section class="src-pages-style----chi-siamo-module---section---3uBn2">...</section>
<section class="src-pages-style----chi-siamo-module---carousel---29alw">...</section>
<div>
  <div class="src-components-ClickToCall----ClickToCall-module---opener---3lt6S"></div>
  <div id="thisIsclickToCall0000000000" class="src-components-Modal----Modal-module---wrunner---1KxRc" aria-...
  > div.src-components-FadeOnScroll----fade-... > section.src-components-PageTitle----Page... > h1.src-components-PageTitle----PageTitle... >
```

- No global scope
- Niente conflitti
- Dipendenze esplicite

Local styling with CSS Modules - Hugo Giraudel @ From the Fron...



E I WEB COMPONENTS?

Aspettiamo Max tra qualche decina di
minuti 😊

Problema:

CONDIVIDERE

Tipo tra Sass e JS

```
$breakpoints: (  
    small: 640,  
    medium: 720,  
    large: 1024  
) ;
```

COME ACCEDERE A QUESTE PROPERTY NEL DOM?

- Non renderizzare intere porzioni di codice
- Utilizzare element query
- Far partire o meno chiamate ajax

```
$breakpoints: (  
    small: 640,  
    medium: 720,  
    large: 1024  
) ;
```

Come condivideri questo in JS?

```
export default {  
    small: 640,  
    medium: 720,  
    large: 1024  
}
```

```
import breakpoints from "./breakpoints.js"
```

E probabilmente il nostro compilatore Sass viene lanciato da JS (Gulp, Webpack, whatelse)

data

- Type: `String`
- Default: `null`

Special: `file` or `data` must be specified

A string to pass to [LibSass](#) to compile. It is recd

```
import breakpoints from "./design/breakpoints"
/* ... */
loader: "sass-loader",
options: {
  data: `$breakpoints: ${breakpoints}`
}
```

js-to-scss

```
npm install js-to-scss -D
```

This is a simple module that will flatten down a javascript object into a string. You can inject that string inside your Sass engine, via the [data option](#)

Example:

```
import flattenObjSass from "js-to-scss";

const color= {
  black: '#000000',
  white: '#ffffff'
}
flattenObjSass(color);
```

will return:

```
$black: #000000;
$white: #ffffff;
```

```
import breakpoints from "./design/breakpoints"
import flattenObjSass from "js-to-scss"
/*...*/
loader: "sass-loader",
options: {
  data: `$breakpoints:
${flattenObjSass(breakpoints)}`
}
```

functions (>= v3.0.0) - *experimental*

This is an experimental LibSass feature. Use with caution.

`functions` is an `Object` that holds a collection of custom functions that may be invoked by the sass files being compiled. They may take zero or more input parameters and must return a value either synchronously (`return ...;`) or asynchronously (`done();`). Those parameters will be instances of one of the constructors contained in the `require('node-sass').types` hash. The return value must be of one of these types as well. See the list of available types below:

`types.Number(value [, unit = ""])`

- `getValue()` / `setValue(value)` : gets / sets the numerical portion of the number
- `getUnit()` / `setUnit(unit)` : gets / sets the unit portion of the number

`types.String(value)`

- `getValue()` / `setValue(value)` : gets / sets the enclosed string

`types.Color(r, g, b [, a = 1.0])` or `types.Color(argb)`

- `getR()` / `setR(value)` : red component (integer from `0` to `255`)
- `getG()` / `setG(value)` : green component (integer from `0` to `255`)
- `getB()` / `setB(value)` : blue component (integer from `0` to `255`)
- `getA()` / `setA(value)` : alpha component (number from `0` to `1.0`)

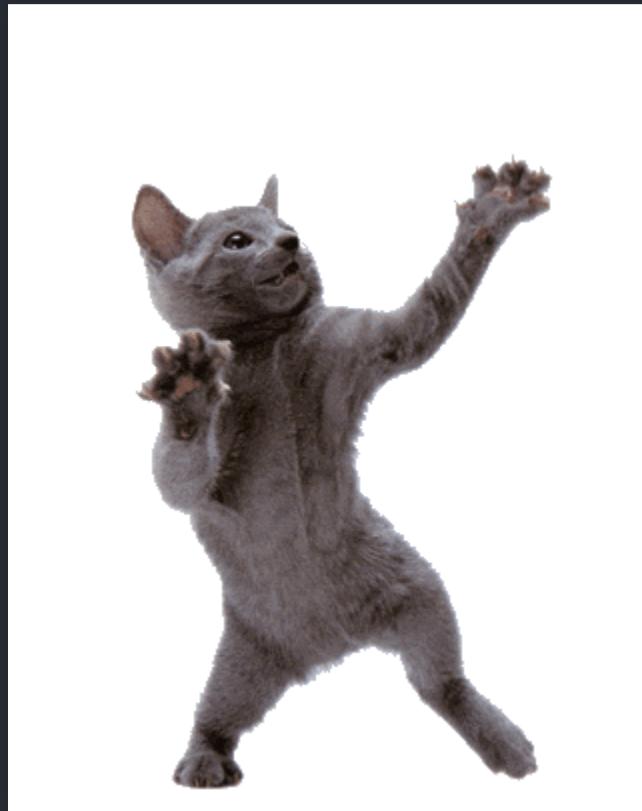
Ad esempio, se volessi condividere quanto accade in un foglio Sass con un altro foglio Sass?

```
'env-map($env-sass-map-name, $map: () )':  
function (envSassMapName, map) {  
if (!env.sass[envSassMapName.getValue()]) {  
  env.sass[envSassMapName.getValue()] =  
    new nodeSass.types.Map(0);  
}  
if (map.getLength()) {  
  env.sass[envSassMapName.getValue()] = map;  
}
```

E CON ALTRI PROCESSORI CSS?

macropodhq/postcss-constants

OK, UN GATTINO PER RIAVERE L'ATTENZIONE:



Problema:

CONDIVIDERE A RUNTIME

Aggiornare il CSS tramite JS

```
Element.style.left = "42px";
```

ANCORA PIÙ FACILE GRAZIE ALLE CUSTOM PROPERTIES

aka CSS variables

```
Element.style.setProperty(  
  `--custom`, '42px'  
)
```

HTML CSS JS Result

EDIT ON

HOVER ME

Standalone parallax scrolling with CSS variables

basicScroll allows you to change CSS variables depending on the scroll position. Use the variables directly in your CSS to animate whatever you want.

[See on GitHub](#)

[Live Demo](#)



Problema:

CODICE RIPETUTO

```
.square {  
    height: 10px;  
    width: 10px;  
}
```

```
// Sass
.square {
  @include square(10px);
}
```

```
// PostCSS
.square {
  @mixin square 10px;
}
.square2 {
  size: 10px;
}
```

```
module.exports = (mixinNode, value) => {
  return {
    width: value,
    height: value,
  }
};
```

Una domanda sorge spontanea

PERCHÉ POSTCSS?

e non uno degli altri?

TEST

```
import size from '../src/mixins/size.js';

test('Should return CSS rules', () => {
  expect(size(null, '64px')).toEqual({
    width: '64px',
    height: '64px',
  })
}) ;
```

Sandrina Pereira: How can Javascript improve your CSS mixins | ...



ANCHE SASS È TESTABILE: TRUE

```
@include test-module('Zip [function]')
@include test('Zips multiple lists into one')
// Assert the expected results
@include assert-equal(
    zip(a b c, 1 2 3),
    (a 1, b 2, c 3));
}
```

Problema:

COMPONENTIZZARE

Accattatil

```
<a href="#" class="btn btn-primary">
```

Accattatil

```
</a>
```



<Button primary>

Accattatil

</Button>

```
const Button = styled.button`  
background: ${props =>  
  props.primary  
  ? 'palevioletred'  
  : 'white'  
};  
color: ${props => props.primary ? 'white' : 'palev:  
font-size: 1em;  
margin: 1em;  
padding: 0.25em 1em;  
`
```

Una domanda sorge spontanea

PERCHÉ NON LE CLASSI?

meno spontanea di prima

- Eliminano il global space
- Le dipendenze sono esplicite
- Condividere costanti è più facile

MA PER ME IL SELLING POINT È...

suspense

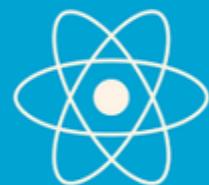
attesa

momento verità



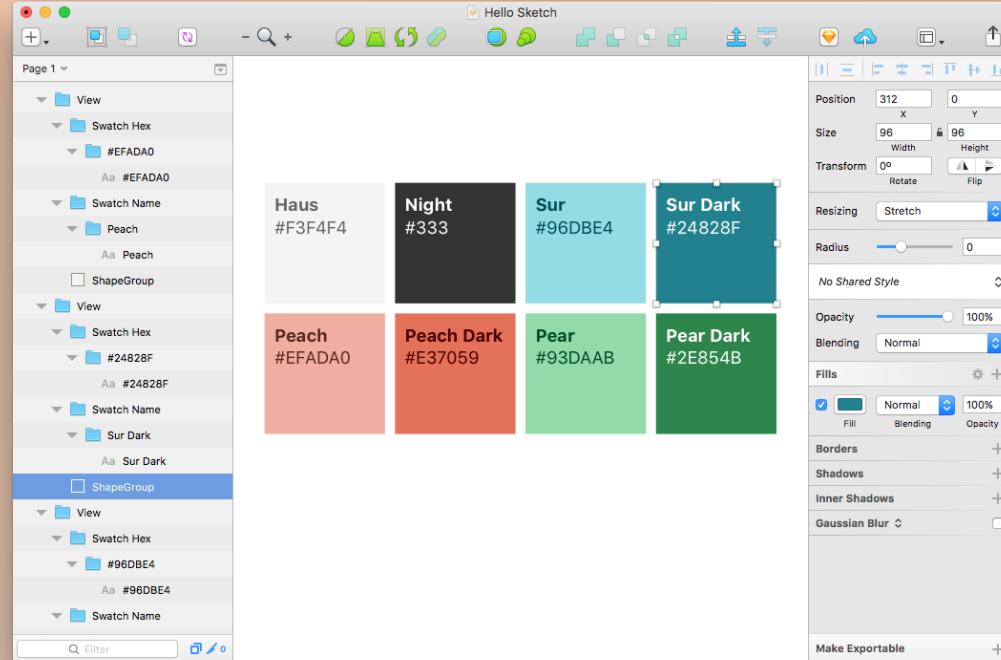
L'ASTRAZIONE

???



React Native

```
my-command.js — ~/code/react-sketchapp/hello-sketch
28     .... {name} ~
29     <Text>
30     <Text name="Swatch Hex" style={{ color: textColor(hex) }}>
31     .... {hex}
32     </Text>
33     </View>
34   );
35   ...
36 const Document = ({ colors }) => (
37   <View style={styles.wrapper}>
38     {Object.keys(colors).map(color => <Swatch name={color} hex={colors[color]} />)}
39   </View>
40 );
41 ...
42 const colorList = {
43   Haus: '#F3F4F4',
44   Night: '#333',
45   Sur: '#96DBE4',
46   'Sur Dark': '#24828F',
47   Peach: '#EFADA0',
48   'Peach Dark': '#E37059',
49   Pear: '#93DAAB',
50   'Pear Dark': '#2E854B',
51 };
52 ...
53 export default context =>
54   render(<Document colors={colorList} />, context);
55
```



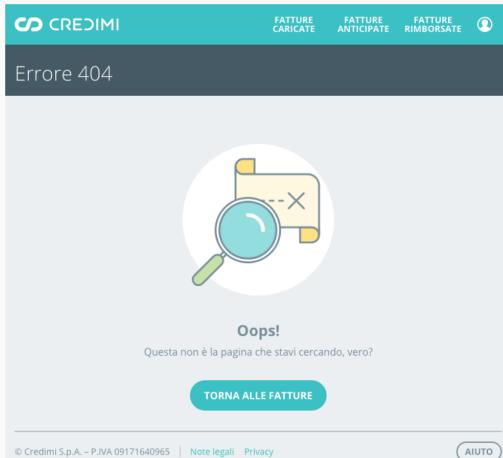
Problema:

REGRESSION TEST

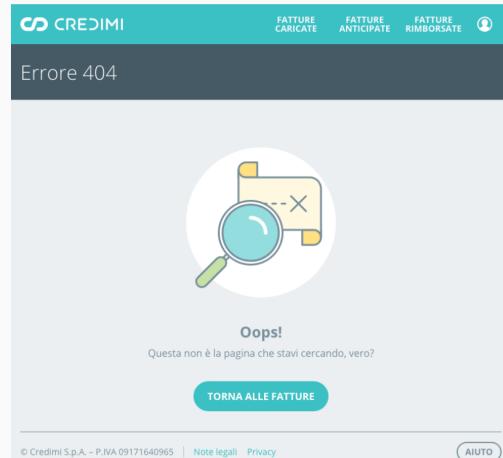
cos'è?

filename: OMClient_404_0_document_2_medium.png

REFERENCE



TEST



SHOW DIFFS



DIFF



BACKSTOPJS

```
{  
  "id": "OM Client",  
  "viewports": [  
    {"label": "main", "width": 1366, "height": 768}  
  ]  
  {  
    "label": "archive",  
    "url": "https://localhost:9000/archive",  
    "delay": 2000,  
    "hideSelectors": ["#loading-bar-new"],  
    "onBeforeScript": "helpers/localStorageOM.js"  
  }  
}
```

```
module.exports = (chromy, scenario) => {
  chromy
    .wait('.ic-app')
    .click('.introjs-skipbutton');
};
```

credimi.com Storybook

CREDIMI UI KIT

Filter

CustomerStory
Features
Hero
HowItWorks
Manifesto
PageTitle
PartnersGrid
Person
PressOverview
PressQuotes
Quote
Requirements list
TestimonialOverview
TextColumn
Threesome
TotalFinanced
Video

Button
Card
Carousel
Colors
Cookie Bar
Fade on scroll

PRIMARY

PRIMARY

PRIMARY

PRIMARY

INVERTED

INVERTED

INVERTED

INVERTED

README KNOBS ACTION LOGGER

Simply Buttons

Click a button to copy style

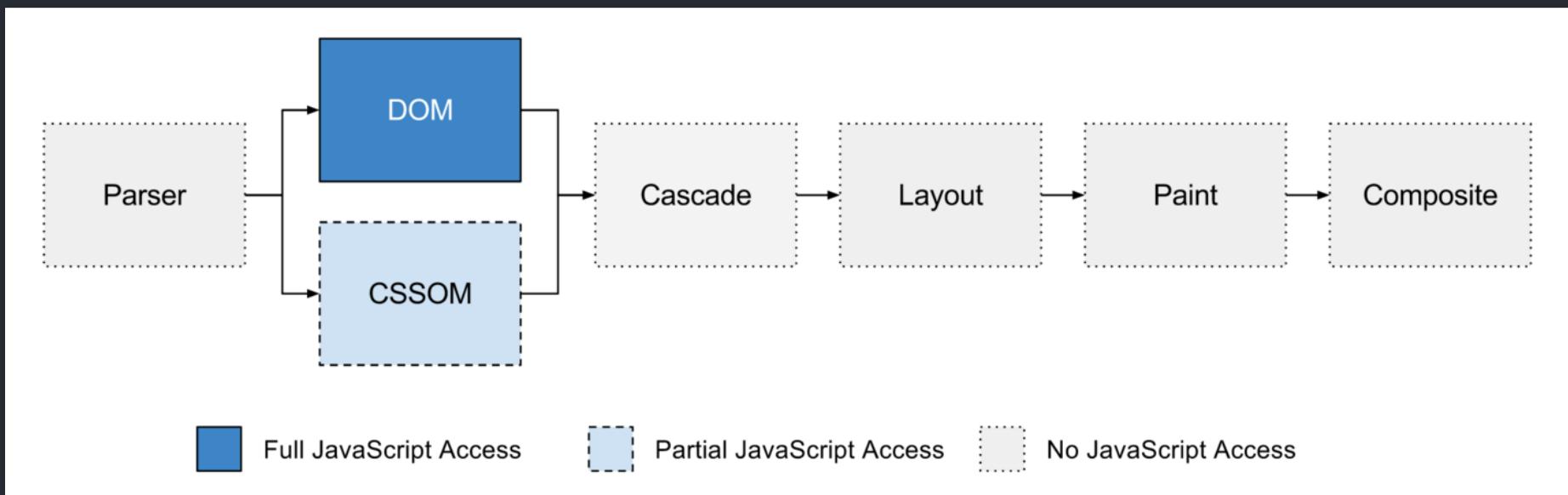
Buttons are not managed via `<button>` tag, because those days everything could look like a button.
So? So we use CSS and with CSS we mean SASS. How? Include the component `Button.sass` (don't

Parte 3

IL FUTURO

WEB COMPONENTS

PROJECT HOUDINI



CSS PARSER API

@apply o @extends a run time!

CSS LAYOUT API

A screenshot of a web browser window displaying a grid of 8 cards, each containing a numbered list of buzzwords. The browser's developer tools are open, showing the DOM structure and CSS styles for the cards.

The cards contain the following lists:

1. Drinking vinegar seltan intelligentia, cray knausgaard aesthetic trade deep v vice. Fam credo coding listicle whatever semiotics yr kitsch leggings kale chips tilde selvege meggings crucifix disrupt portland.
2. Pabst bitters knausgaard, art party jianbing urchin aesthetic. Sartorial affogato four dollar toast iPhone, iPhone offal pabst cloud bread air plant shoreditch.
3. Shaman pabst knausgaard raw denim next level, hella artisanal dreamcatcher whatever lomo. Slow-carb 8-bit biodiesel celiac. YOLO leggings mixtape activated charcoal lo-fi.
4. Umami shabby chic turmeric, portland cloud bread gastropub sartorial meh authentic brooklyn enamel pin disrupt DIY edison bulb. Narwhal beard seltan helvetica keytar. Sartorial plaid milksh disrupt waistcoat shabby chic. Hammock skateboard butcher pop-up kitsch before they sold out.
5. Iceland art party subway tile, freegan thundercats gentrify pok pok offal pug. Before they sold out skateboard hella paleo. Typewriter roof pinstripe listicle tbh he.
6. Austin fixie XOXO franzen tattooed subway tile. Shoreditch street art typewriter dreamcatcher. Vegan mixtape bicycle rights tumblr, +1 ethical occupy letterpress sriracha polaroid master cleanse.
7. Shag bag mumblecore listicle Sriracha media semiotics.
8. Pour-over adaptogen selfies, scenester shabby.

The developer tools sidebar shows the following CSS code:

```
<!DOCTYPE html>
<html>
  <head></head>
  <body>
    <div class="test"></div> == $0
    <script></script>
  </body>
</html>
```

```
element.style {
}
.test {
  display: layout(test);
  background: #ccc;
  border-radius: 3px;
  font-family: sans-serif;
}
.test {
  color: white;
  margin: 20px 0;
  width: 520px;
  background-color: red;
  padding: 20px;
  columns: auto;
}
div {
  display: block;
}
```

CSS PAINT API

Per farla breve una sorta di Canvas

```
.bubble {  
    background: paint('circle');  
}
```

A riguardo

Recap

QUINDI?

**USIAMO DA ANNI TOOL PER
MIGLIORARE IL NOSTRO CSS**

JS È GIÀ NELLA NOSTRA BUILD CHAIN, USIAMOLO A NOSTRO VANTAGGIO

Per risolvere alcuni problemi che possiamo avere con CSS

**DAI RAGA, TUTTE STE
POSSIBILITÀ SONO UNA FIGATA**



GRAZIE



DAVIDE DI PUMPO

- Technology - Credimi
- Co-organizer Milano Frontend
- **MakhBeth** on: Twitter, Github, Internet
- I like Cats, Drink, Comics and Videogames...