

# **Framework-less frontend development**

**by Matthias Hryniszak**



# **Tuesday January 9th, 2018**

## **The day the world changed**



# The Basics

**HTML/DOM** is the representation of concepts

**CSS** determines how they look like and what is their positioning

**JavaScript** makes them behave dynamically

# Let's talk about HTML

# Old-scool HTML 🤪

```
<body>
  <div class="container">
    <div class="header">
      Header
    </div>
    <div class="nav">
      Navigation
    </div>
    <div class="sidebar">
      Sidebar
    </div>
    <div class="main">
      <p>...Lorem ipsum here...</p>
    </div>
    <div class="footer">
      Footer
    </div>
  </div>
</body>
```

# Semantic HTML using HTML5

```
<body>
  <header>
    Header
  </header>
  <nav>
    Navigation
  </nav>
  <aside>
    Sidebar
  </aside>
  <article>
    <p>...Lorem ipsum here...</p>
  </article>
  <footer>
    Footer
  </footer>
</body>
```



# Let's talk about CSS layouts

# CSS Layouts pre-2018

- Unintuitive tricks
- Negative margins, floats, clear-fixes and more
- Obstructive constructs like containers for no obvious reasons
- Frameworks

# CSS Layouts as it exists Today

- Grid (<https://caniuse.com/#search=grid>)
- Flexbox (<https://caniuse.com/#search=flex>)
- Media queries (<https://caniuse.com/#feat=css-mediaqueries>)

## First: fix the browser 😊

```
* {  
  box-sizing: border-box;  
}
```

# Let's get you laid out on a grid

```
body {  
  display: grid;  
}
```

```
header { }  
nav     { }  
aside   { }  
article { }  
footer  { }
```

# Naming things

```
header { area: header; }  
nav    { area: navigation; }  
aside  { area: sidebar; }  
article { area: main; }  
footer { area: footer; }
```

# Defining how things are laid out - mobile first

```
body {  
  display: grid;  
  
  grid-template-areas:  
    "header"  
    "navigation"  
    "main"  
    "sidebar"  
    "footer"  
}
```

## Let's go tablet-size

```
@media (min-width: 500px) {  
  body {  
    grid-template-columns:  
      minmax(200px, 300px) minmax(350px, 500px);  
  
    grid-template-areas:  
      "header      header"  
      "navigation  navigation"  
      "sidebar     main"  
      "footer      footer"  
  }  
}
```



## Let's go big screen - fluid layout

```
@media (min-width: 800px) {  
  body {  
    grid-template-columns:  
      300px minmax(500px, 1fr);  
  
    grid-template-areas:  
      "header      header"  
      "navigation  navigation"  
      "sidebar     main"  
      "footer      footer"  
  }  
}
```

## Let's go big screen - fixed layout

```
@media (min-width: 800px) {  
  body {  
    grid-template-columns:  
      1fr 300px 500px 1fr;  
  
    grid-template-areas:  
      ". header      header      ."  
      ". navigation  navigation  ."  
      ". sidebar     main        ."  
      ". footer      footer      ."  
  }  
}
```

## Where did the 800px came from?

```
@media (min-width: 500px) {  
  body {  
    grid-template-columns:  
      minmax(200px, 300px) minmax(350px, 500px);  
  }  
}
```

$$300px + 500px = 800px$$

# Let's talk about JavaScript

# JavaScript - the bad parts 🙄

```
// https://www.destroyallsoftware.com/talks/wat
```

```
> [] + []
```

```
> [] + {}  
[object Object]
```

```
> {} + []  
0
```

```
> {} + {}  
NaN
```

# Asynchronous operations

<https://caniuse.com/#feat=async-functions>

```
function resolveAfterXSeconds(x) {  
  return new Promise(resolve => {  
    setTimeout(() => { resolve('resolved') }, x);  
  });  
}  
  
async function asyncCall() {  
  console.log('calling');  
  const result = await resolveAfterXSeconds(1);  
  console.log(result);  
}  
  
asyncCall();
```

# DOM manipulation

<https://caniuse.com/#feat=classList>

<https://caniuse.com/#feat=insert-adjacent>

<https://caniuse.com/#search=dataset>

[https://developer.mozilla.org/en-](https://developer.mozilla.org/en-US/docs/Web/API/Element/insertAdjacentElement)

[US/docs/Web/API/Element/insertAdjacentElement](https://developer.mozilla.org/en-US/docs/Web/API/Element/insertAdjacentElement)

[https://developer.mozilla.org/en-](https://developer.mozilla.org/en-US/docs/Web/API/HTMLElement/dataset)

[US/docs/Web/API/HTMLElement/dataset](https://developer.mozilla.org/en-US/docs/Web/API/HTMLElement/dataset)



# JavaScript without frameworks

```
for (let i = 0; i < 100; i++) {  
  const div = document.createElement('div')  
  
  const color = Math.round(Math.random() * 0xffffffff)  
    .toString(16).padStart(6, '0');  
  
  div.dataset.color = '#' + color;  
  div.style.backgroundColor = '#' + color  
  div.innerText = '#' + color  
  div.classList.add('item')  
  
  const rnd = Math.random();  
  if (rnd > 0.25 && rnd < 0.5) div.classList.add('one')  
  if (rnd > 0.5 && rnd < 0.75) div.classList.add('two')  
  if (rnd > 0.75) div.classList.add('three')  
  
  document.body.insertAdjacentElement('beforeend', div)  
}
```

# Maintainability

<https://caniuse.com/#feat=es6-class>

<https://caniuse.com/#feat=custom-elements>

<https://www.polymer-project.org/>

# Maintainability

```
class RandomColorBox extends HTMLElement {
  connectedCallback () {
    const color = this.getRandomColor()
    this.style.backgroundColor = '#' + color
    this.innerText = '#' + color
    this.classList.add(this.getRandomClass())
  }
  getRandomColor () {
    return Math.round(Math.random() * 0xffffffff)
      .toString(16).padStart(6, '0');
  }
  getRandomClass () {
    const rnd = Math.random();
    if (rnd > 0.25 && rnd < 0.5) return 'one'
    if (rnd > 0.5 && rnd < 0.75) return 'two'
    return 'three'
  }
}
```

# Readability

```
window
  .customElements
  .define('random-color-box', RandomColorBox)

for (let i = 0; i < 100; i++) {
  document.body.appendChild(
    document.createElement('random-color-box')
  )
}
```

# Readability

<https://caniuse.com/#feat=template-literals>

# Definig multiline template literals

```
const variable = 'this'
```

```
const square = x => x * x
```

```
const template = `Hello!
```

```
This is a multiline string that you can have  
placeholders such as ${variable}, ${1 + 2}  
or ${square(2)} in.`
```

# Modularity

<https://caniuse.com/#search=module>

<https://twitter.com/FirefoxNightly/status/951382754125545473>

# Modularity

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport"
        content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Multicolor supergrid</title>
  <link rel="stylesheet" href="./index4.css">
</head>
<body>
  <script type="module" src="./index4.js"></script>
</body>
</html>
```



# Modularity

```
import { RandomColorBox } from './random-color-box.js'

window
  .customElements
  .define('random-color-box', RandomColorBox)

for (let i = 0; i < 100; i++) {
  document.body.appendChild(
    document.createElement('random-color-box')
  )
}
```

# Modularity

```
/**
 * RandomColorBox component
 *
 * This component creates a box in a random color
 * suitable for display in a grid layout with one of 3
 * randomly added classes: 'one', 'two' and 'three'
 */
export class RandomColorBox extends HTMLElement {
  connectedCallback () { ... }
  getRandomColor () { ... }
  getRandomClass () { ... }
}
```

# Arrow functions

<https://caniuse.com/#feat=arrow-functions>

🔥 that this hell 🔥

```
function click(e) {  
  console.log(this)  
}  
  
click()  
  
$('body').on('click', click)
```

🔥 that this hell 🔥

```
function click(e) {  
  console.log(this)  
}  
  
click()  
  
$('body').on('click', click)
```

🔥 that this hell 🔥

```
var that = this;  
  
function click(e) {  
  console.log(that)  
}  
  
click()  
  
$('body').on('click', click)
```

## Fat arrow + closure

```
const click = e => console.log(this)

document.body.addEventListener('click', click)
click()
```

# Collections API

```
const a = [ { x: 1 }, { x: 2 }, { x: 3 }]  
  
const b = a.map(i => i.x)  
b.reduce((acc, x) => acc + x)  
b.reduce((acc, x) => acc * x)  
b.reduce((acc, x) => acc + ', ' + x) // b.join(', ')  
b.some(x => x % 2 == 0)  
b.every(x => x < 10)  
b.find(x => x % 2 == 0)  
b.findIndex(x => x % 2 == 0)  
b.filter(x => x % 2 == 0)  
b.forEach(console.log)
```



# DOM Traversal

<https://caniuse.com/#feat=queryselector>

<https://caniuse.com/#feat=element-closest>

# Searching for DOM elements

```
const div = document.querySelector('div')
const divs = document.querySelectorAll('div')

const input = div.querySelector('input[type=text]')
const inputs = div.querySelectorAll('input[type=text]')

const body = div.closest('body')
```

Please note: this is the real CSS3 selector used here, not some pseudo selector invented for the purpose of fixing ancient browser shortcomings!

# Final thoughts

**When in doubt see <http://caniuse.com>**

**Favor polyfills over frameworks**

**Use frameworks that work with the platform\***

# Questions?

# May the platform be with you!

Blog:

<https://padcom13.blogspot.com>

LinkedIn:

<https://linkedin.com/in/padcom>