Igor Laborie

Expert Java & Web,

<http://www.monkeypatch.io/>

🐦 @ilaborie <https://twitter.com/ilaborie>

✉ igor@monkeypatch.io <mailto:igor@monkeypatch.io>

> When designing computer systems, one is often faced with a choice between using a more or less powerful language for publishing information, for expressing constraints, or for solving some problem. This finding explores tradeoffs relating the choice of language to reusability of information. The "Rule of Least Power" suggests **choosing the least**

1. Texte

2. HTML (sémantique) & CSS (layout, style, animations simples)

3. SVG (formes et animations complexes)

4. JavaScripts

⚠️*... mais il y a toujours de bonnes raisons pour ne*

Le CSS c'est vaste

- Selectors
- Box model
- Float
- Media Query
- Transitions

- Responsive Design
- Media
- Variables
- Colors
- Shapes

Utiliser un pré-processeur ?

Bordure des boutons

```css
button {
  background: lightblue;
  border: medium solid purple;
}
button.danger { /*
   background: salmon;
   color: rebeccapurple; */
}
```

Alors utilise-t-on un pré-processeurs ?

Oui, mais privilégiez:

- le CSS

- les post-processeurs

- ➥ `currentColor` <https://css-tricks.com/currentcolor/>

- ➥ `background-origin`

  <https://developer.mozilla.org/fr/docs/Web/CSS/background-origin>

- ➥ CSS Variables (aka Custom Properties)

  <https://www.w3.org/TR/css-variables/>

Unités

Une histoire d'unités CSS

Les unités de longueur

px, cm, pt, ...	longueurs absolues (mesure physique)
em, rem	fonction de la `font-size`
ex, ch	hauteur d'un x, largeur d'un 0
vh, vw	(100vh, 100vw) = (hauteur, largeur) du *viewport*

Holy Grail avec calc

Header

Menu | Lorem ipsum dolor sit amet, consectetur adipisicing elit. Consequatur corporis dicta libero placeat quas? Dicta | Side

- ➼ Unités <https://developer.mozilla.org/fr/docs/Web/CSS/length) et [Truc et astuces](https://www.w3.org/Style/Examples/007/units.fr.html>

Flexbox et Grid

DevFest
Toulouse 2017

Header

Menu | Lorem ipsum dolor sit amet, consectetur adipisicing elit. Consequatur corporis dicta libero placeat quas? Dicta | Side

Header

Menu | Lorem ipsum dolor sit amet, consectetur adipisicing elit. Consequatur corporis dicta libero placeat quas? Dicta | Side

Bilan Flexbox & Grid

Flexbox

- ➥ Flexbox, et le CSS redevient fun ! (Hubert SABLONNIÈRE) <https://www.youtube.com/watch?v=5F_ngjHDcJQ>

- ➥ Solved by Flexbox <https://philipwalton.github.io/solved-by-flexbox/>

- ➥ Flexbox Froggy <https://flexboxfroggy.com/>

Grid

- ➥ Grid by examples <https://gridbyexample.com/>

- ➥ CSS Grid Changes Everything (About Web Layouts) by Morten Rand-Hendriksen <https://www.youtube.com/watch?v=txZq7Laz7_4>

- ➥ Grid Garden <http://cssgridgarden.com/>

Pseudo éléments

Le dinner d'un philosophe

```css
.table {
  color: gray;
  font-size: 5em;
  /*content: '';*/
}
```

Triangle avec des bordures

```css
div.top, div.right, div.bottom, div.left {
  border: 1em solid transparent;
  display: inline-block;
  box-shadow: 0 0 0 .1em transparent;
}

div.top { border-top-color: transparent; }
div.right { border-right-color: transparent; }
div.bottom { border-bottom-color: transparent; }
div.left { border-left-color: transparent; }
```

Info-bulle

```css
.popover {
  position: relative;
  background: teal;
}

/*.popover::before {
  position: absolute;
  z-index: 0;
  content: '';
  top: 0em; left: 0em;
  border: 1em solid red;
  border-top-color: red;
}*/
```

Bilan pseudo éléments

- ➼ The :before and :after pseudo-elements

  <https://www.w3.org/TR/CSS22/generate.html#before-after-content>

- mais aussi `::first-letter`, `::first-line`, `::selection`, `::backdrop`

- ➼ An Ultimate Guide To CSS Pseudo-Classes And Pseudo-Elements <https://www.smashingmagazine.com/2016/05/an-ultimate-guide-to-css-pseudo-classes-and-pseudo-elements>

  ⚠️ `::before` et `::after` ne marchent pas sur `input`, `img`, `iframe` (pas encore spécifié)

- Table et assiette de ➼ CSS Diner <https://flukeout.github.io/>

- ➼ Dîner des philosophes

Animations

Texte de chargement

```css
.loader {
  display : inline-block;
  white-space : normal;
  height: 1em;
  line-height : 1.5;
  overflow: auto;
  box-shadow  : 0 0 0 .05em red;
}


.loader::before {
  display : inline-table;
  /*content: '0\a 1\a 2\a 3\a 4\a 5\a 6\a 7\a 8\a 9';*/
  /*content: '⠈\a ⠉\a ⠋\a ⠊\a ⠂\a ⠄\a ⠠\a ⠤\a ⠆\a ⠒';*/
  /*animation: spin 5s infinite;*/
}


@keyframes spin {
  to { transform : translateY(-15em); }
}
```

Dessiner

```css
svg path {
  stroke: purple;
  stroke-width: .1em;
  fill: none;
  /*stroke-dasharray: 0;*/
  /*stroke-dashoffset: 0;*/
  /*animation: draw 4s linear;*/
}

@keyframes draw {
  to { stroke-dashoffset: 0; }
}
```

Bilan animations

- ➤ Utiliser les animations CSS

  <https://developer.mozilla.org/fr/docs/Web/CSS/Animations_CSS/Utiliser_les_animations_CSS>

- ➤ Text Spinner <http://tawian.io/text-spinners/>

- ➤ CSS only loader <https://www.pexels.com/blog/css-only-loaders/>

- ➤ Animate.css <https://daneden.github.io/animate.css/>

- ➤ How SVG Line Animation Works <https://css-tricks.com/svg-line-animation-works/>

- ➤ `<progress>`

Pseudo classes
d'état

Usage des info-bulles

Input Text

mandatory field

Mandatory

Pseudo états

- :hover

- :focus

- :visited

- :checked

- :valid

- :invalid

- :empty

- :target

```css
.like input[type=checkbox] + label {
    box-shadow: 0 0 0 1px red;
}
.like input[type=checkbox] + label::before {
    content: '';
}
.like input[type=checkbox] + label::before {
    content: '';
}
.like fieldset input[type=checkbox] { opacity: 1; }
```

```
.switch + label {
  display: block;
  position: relative;
  padding: .1em;
  width: 2em;
  height: 1em;
  background-color: #ccc;
  border-radius: 1em;
  border: medium solid #444;
  transition: 0.4s;
}

.switch:checked + label {
  background-color: green;
}
```

```
.switch + label::before {
  display: block;
  position: absolute;
  content: '';
  top: 0.1em;
  left: 0.1em;
  height: 1em;
  width: 1em;
  background-color: #fff;
  border-radius: 50%;
  transition: all 0.25s;;
}

.switch:checked + label::before {
  transform: translateX(1em);
}
```

Panel

```css
.panel input[type=checkbox] {
    /* hide me */
}
```

Démo des onglets

HTML

```
details {
  border: medium solid currentcolor;
  border-radius: .25em;
  width: 100%;
}

details summary {
  background: #888; color: #eee;
}
```

Dialog

```css
.editable dialog {
  border: medium solid rgba(0, 0, 0, 0.3);
  border-radius: .125em;
  padding: .125rem;
  box-shadow: .25em .25em .125em rgba(0, 0, 0, 0.42);
}

/* .editable dialog::backdrop {
  position: fixed;
  top: 0; right: 0; bottom: 0; left: 0;
  background-color: rgba(0, 0, 0, 0.8);
} */
```

Polyfill

- ➥ Collapsible Panel Polyfill <https://github.com/chemerisuk/better-

details-polyfill/>

Compatibilité des navigateurs

➼ **caniuse** <http://caniuse.com>

➼ **The CSS3 / CSS4 Test** <http://css3test.com>

IE 7+, Firefox, Chrome   Pseudo classes (CSS3 selectors *93*)

IE 8+, Firefox, Chrome   `::before, ::after` *98*

IE 9+, Firefox, Chrome   `currentColor` *98*

`background-origin` *98*   `box-shadow` *98*

`calc` *97*   `vh, vw, ...` *97*

IE 10+, Firefox, Chrome   `flexbox` *98*

Conclusion

1. Utilisez du CSS pour simpifier le code

2. Utilisez intelligemment les pre/post-processeurs

3. HTML, SVG are Awesome !

1. Revue de code

2. DRY

3. Clean Code

4. Single Responsibility Principle

- ⤵ les slides <>
- ⤵ le code <>

Pour apprendre

- `(Ctrl|Cmd) + Shift + i`

- ↣ CSS Secret by Lea Verou

  <https://www.amazon.fr/CSS-Secrets-Lea-Verou/dp/1449372635>

- ↣ CSS sur MDN <https://developer.mozilla.org/fr/docs/Web/CSS>

- ↣ CodePen <https://codepen.io/> , ↣ JSFiddle <https://jsfiddle.net/> ,

  ↣ Dabblet <http://dabblet.com/> ,...

- ↣ CSS Tricks <>

- ↣ Shop Talk Show <>

CSS
IS
AWESOME!

DevFest
Toulouse 2017