

**CSS
IS
AWESOME!**

Igor Laborie

Expert Java & Web, < **Monkey Patch** /> <<http://www.monkeypatch.io/>>

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

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

 *Je ne suis pas un designer*

“ 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 powerful language suitable for a given purpose.

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 pas suivre ces règles

- Selectors
- Box model
- Float
- Media Query
- Transitions
- Gradients
- Responsive Design
- Media
- Variables
- Colors
- Shapes
- ...

- I. Utiliser un pré-processeur ?
- II. Unités
- III. Flexbox et Grid
- IV. Pseudo éléments
- V. Animations
- VI. Pseudo classes d'état
- VII. HTML
- VIII. Compatibilité des navigateurs
- IX. Conclusion

Utiliser un pré-processeur ?

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


Alors utilise-t-on un pré-processeurs ?

#8

Oui, mais privilégiez:

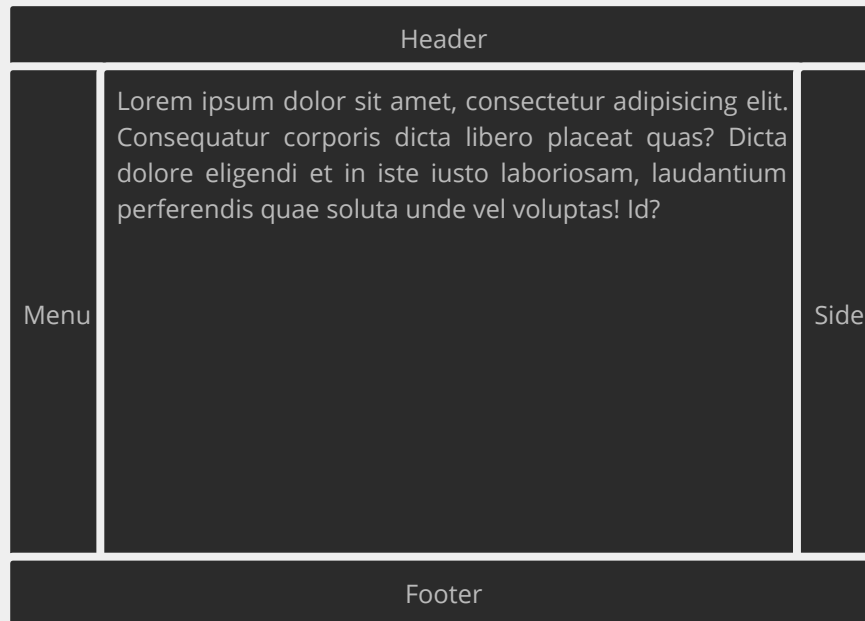
- le CSS
- les post-processeurs
- ➡ `currentColor` <<https://css-tricks.com/currentcolor/>>
- ➡ `background-origin`
<<https://developer.mozilla.org/fr/docs/Web/CSS/background-origin>>
- ➡ CSS Variables (aka Custom Properties)
<<https://www.w3.org/TR/css-variables/>>
- ➡ CSS Color Module Level 4 <<https://www.w3.org/TR/css-color-4/>>

Unités



➔ CommitStrip <<http://www.commitstrip.com/fr/2016/10/10/a-story-about-css-units/>>

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 <i>viewport</i>
vmin, vmax	$\min(1vh, 1vw)$, $\max(1vh, 1vw)$

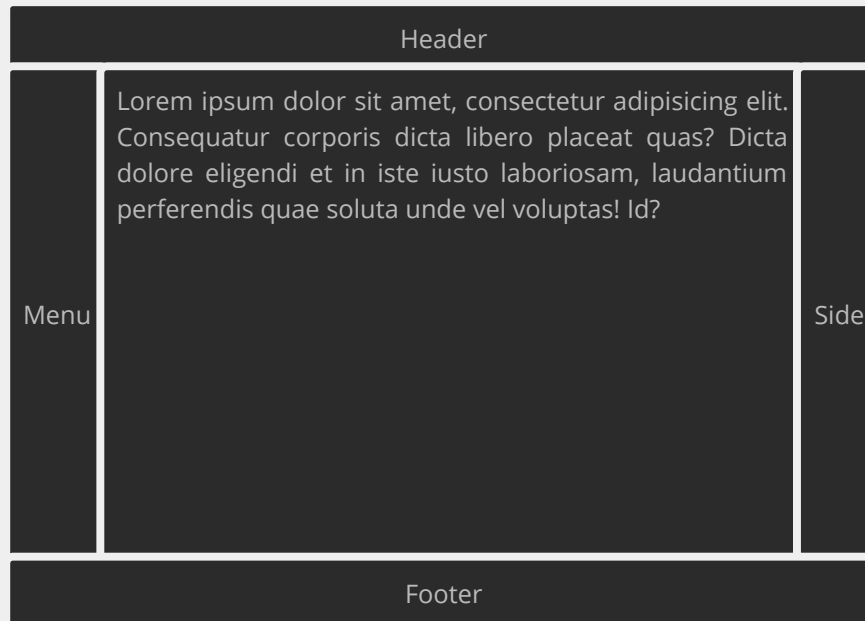


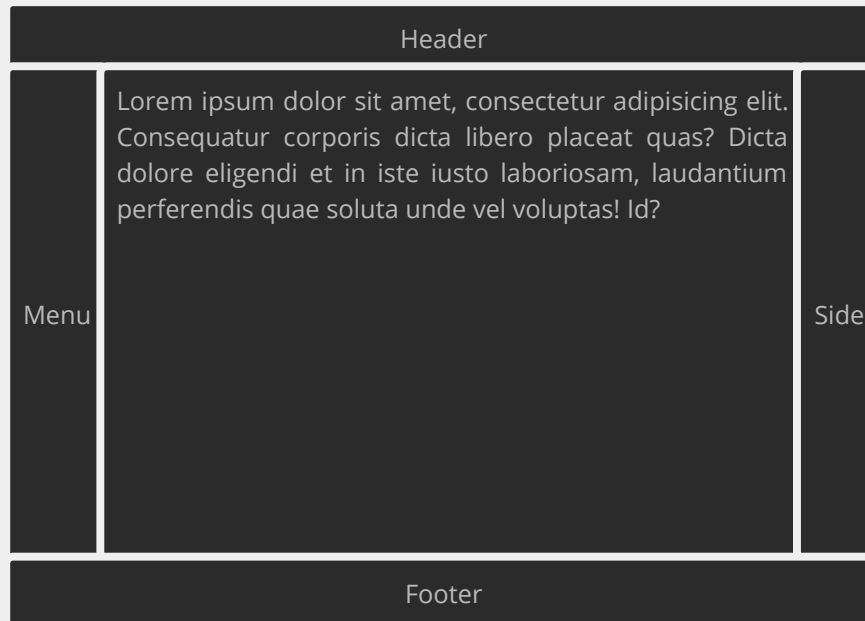
- ➡ Unités <<https://developer.mozilla.org/fr/docs/Web/CSS/length>>
- ➡ Truc et astuces <<https://www.w3.org/Style/Examples/007/units.fr.html>>
- ➡ calc <<https://developer.mozilla.org/fr/docs/Web/CSS/calc>>

Flexbox et Grid

Holy Grail avec flexbox

#15





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/>>
- ➡ Here's a Super Quick Way to Try out CSS Grid <<http://jensimmons.com/post/aug-15-2017/heres-super-quick-way-try-out-css-grid>>

Pseudo éléments

Le dinner d'un philosophe

#19

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

Triangle avec des bordures

#20

```
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; }
```

```
.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;  
}*/
```

- ➡ The :before and :after pseudo-elements
[<https://www.w3.org/TR/CSS22/generate.html#before-after-content>](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>](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/>](https://flukeout.github.io/)
- ➡ Dîner des philosophes
[<https://fr.wikipedia.org/wiki/D%C3%AEner_des_philosophes>](https://fr.wikipedia.org/wiki/D%C3%AEner_des_philosophes)

Animations


```
.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 :\ a :\ a';*/
  /*animation: spin 5s infinite;*/
}

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

```
.editable 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; }  
}
```

- ➡ 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>`

<https://developer.mozilla.org/fr/docs/Web/HTML/Element/Progress>

Pseudo classes d'état

Input Text

Mandatory

➡ hover me <#usage_des_info_bulles>

- :hover
- :focus
- :visited
- :checked
- :valid
- :invalid
- :empty
- :target
- ...

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

Principe pour les onglets

#33



Démo des onglets

#34

7ETML



```
details {  
  border: medium solid currentcolor;  
  border-radius: .25em;  
  width: 100%;  
}  
  
details summary {  
  background: #888; color: #eee;  
}
```

```
.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);  
} */
```

- ➡ Collapsible Panel Polyfill <<https://github.com/chemerisuk/better-details-polyfill/>>
- ➡ Dialog Polyfill <<https://github.com/GoogleChrome/dialog-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

Animations 98

Conclusion

1. Utilisez du CSS pour simplifier le code
2. Utilisez intelligemment les pre/post-processeurs
3. HTML, SVG are Awesome !
4. JavaScript, TypeScript could be Awesome !

1. Revue de code
2. DRY
3. Clean Code
4. Single Responsibility Principle
5. ...

- ➡ les slides <>
- ➡ le code <>
- ➡ Making Of <>

- (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 Flags <>

CSS
is
Awesome!

