CSS IS AWESOME!

$ whoami

Igor Laborie

Expert Java & Web, <Monkey Patch/>

🦪 @ilaborie

✉️ igor@monkeypatch.io

⚠️ *Je ne suis pas un designer*

w3c The Rule of Least Power

> *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.*

Règles du jeu

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*

Le CSS c'est vaste

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

Plan

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

Utiliser un pré-processeur ?

Bordure des boutons


```
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`
- 🦊 `background-origin`
- *w3c* CSS Variables (aka Custom Properties)
- *w3c* CSS Color Module Level 4

Compatibilité

									
	26,6 %	18,6 %	15,8 %	11,5 %	4,7 %	4,4 %	1,3 %	1,0 %	0,5 %
CSS currentColor value	49 58 59 60	61	9.0-9.2 9.3 10.0-10.2 10.3	48 52 54 55	11	10 9.1 10.1	14	4.4 4.4.3-4.4.4	46
CSS3 Background-image options	49 58 59 60	61	9.0-9.2 9.3 10.0-10.2 10.3	48 52 54 55	11	10 9.1 10.1	14	4.4 4.4.3-4.4.4	46
CSS Variables (Custom Properties)	49 58 59 60	61	9.0-9.2 9.3 10.0-10.2 10.3	48 52 54 55	11	10 9.1 10.1	14	4.4 4.4.3-4.4.4	46

Unités

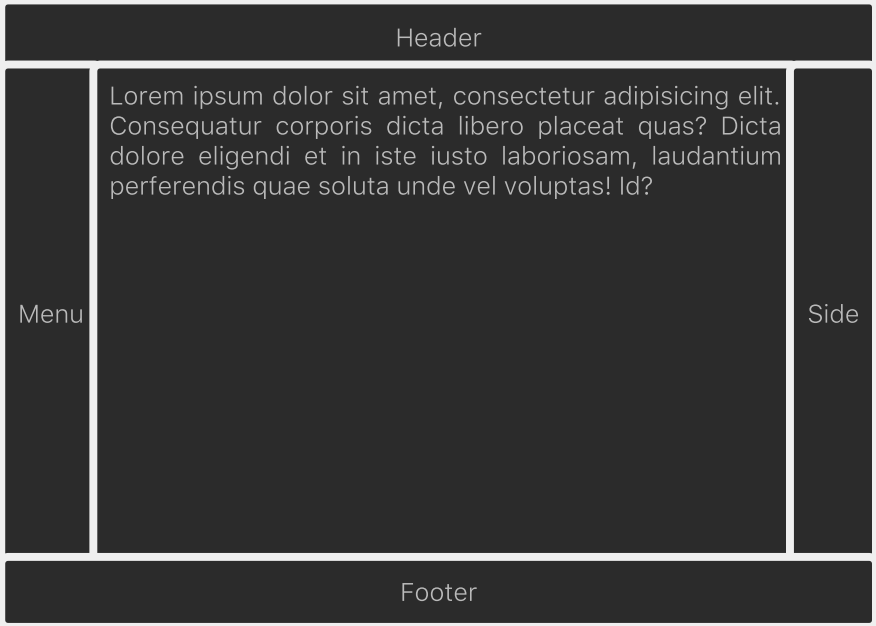
Une histoire d'unités CSS

🎨 CommitStrip

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*
vmin, vmax	min(1vh, 1vw), max(1vh, 1vw)

Holy Grail avec **calc**

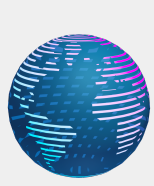



```
<body>
  <header>Header</header>
  <div>
    <nav>Menu</nav>
    <main>Content</main>
    <aside>Side</aside>
  </div>
  <footer>Footer</footer>
</body>
```

Bilan unités

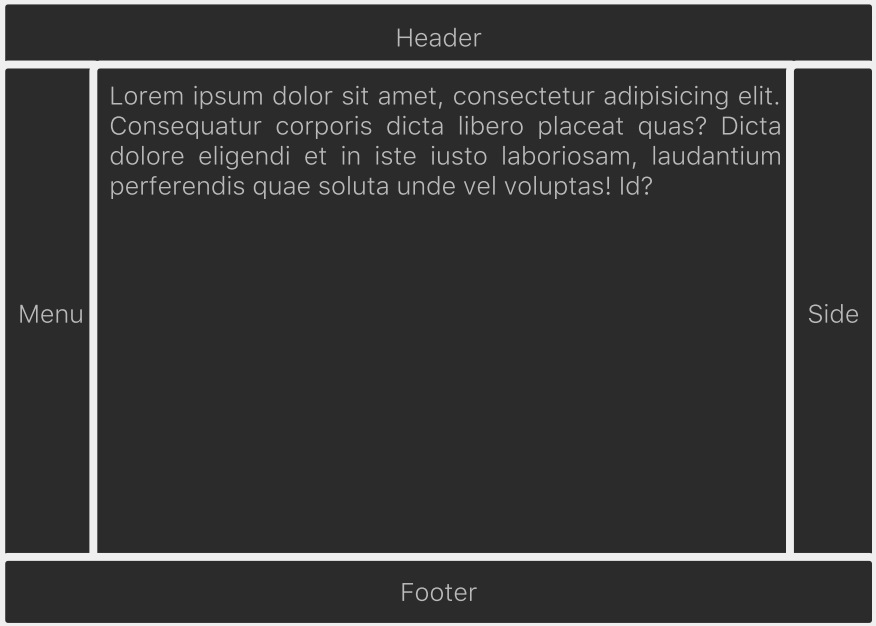
- 🦊 Unités
- *w3c* Truc et astuces
- 🦊 `calc`
- ✳ Fun with Viewport Units

Compatibilité

	 26,6 %	 18,6 %	 15,8 %	 11,5 %	 4,7 %	 4,4 %	 1,3 %	 1,0 %	 0,5 %
rem (root em) units	49 58 59 60	61	9.0-9.2 9.3 10.0-10.2 10.3	48 52 54 55	11	10 9.1 10.1	14	4.4 4.4.3-4.4.4	46
Viewport units: vw, vh, vmin, vmax	49 58 59 60	61	9.0-9.2 9.3 10.0-10.2 10.3	48 52 54 55	11	10 9.1 10.1	14	4.4 4.4.3-4.4.4	46
calc() as CSS unit value	49 58 59 60	61	9.0-9.2 9.3 10.0-10.2 10.3	48 52 54 55	11	10 9.1 10.1	14	4.4 4.4.3-4.4.4	46

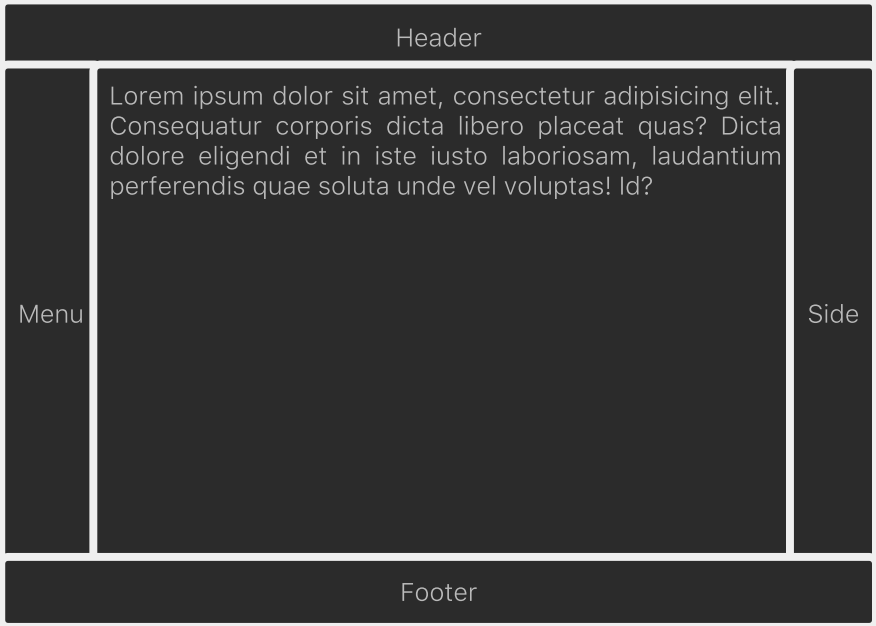
Flexbox et Grid

Holy Grail avec **flexbox**




```
<body>
  <header>Header</header>
  <div>
    <nav>Menu</nav>
    <main>Content</main>
    <aside>Side</aside>
  </div>
  <footer>Footer</footer>
</body>
```

Holy Grail avec **grid**




```
<body>
  <header>Header</header>
  <div>
    <nav>Menu</nav>
    <main>Content</main>
    <aside>Side</aside>
  </div>
  <footer>Footer</footer>
</body>
```

Bilan Flexbox & Grid

Flexbox

- 📹 Flexbox, et le CSS redevient fun ! (Hubert SABLONNIÈRE)
- 📐 Solved by Flexbox
- 🐸 Flexbox Froggy

Grid

- # Grid by examples
- 📹 CSS Grid Changes Everything (About Web Layouts) by Morten Rand-Hendriksen
- 🥕 Grid Garden

Compatibilité

	26,6 %	18,6 %	15,8 %	11,5 %	4,7 %	4,4 %	1,3 %	1,0 %	0,5 %
CSS Flexible Box Layout Module	49 58 59 60	61	9.0-9.2 9.3 10.0-10.2 10.3	48 52 54 55	11	10 9.1 10.1	14	4.4 4.4.3-4.4.4	46
CSS Grid Layout	49 58 59 60	61	9.0-9.2 9.3 10.0-10.2 10.3	48 52 54 55	11	10 9.1 10.1	14	4.4 4.4.3-4.4.4	46

Pseudo éléments

Le dinner d'un philosophe


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

Triangle avec des bordures


```
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


```
.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

- w3c The :before and :after pseudo-elements
- mais aussi `::first-letter`, `::first-line`, `::selection`, `::backdrop`
- 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
- Ⓦ Dîner des philosophes

Compatibilité

	 26,6 %	 18,6 %	 15,8 %	 11,5 %	 4,7 %	 4,4 %	 1,3 %	 1,0 %	 0,5 %
CSS Generated content for pseudo-elements	49 58 59 60	61	9.0-9.2 9.3 10.0-10.2 10.3	48 52 54 55	11	10 9.1 10.1	14	4.4 4.4.3-4.4.4	46
CSS3 2D Transforms	49 58 59 60	61	9.0-9.2 9.3 10.0-10.2 10.3	48 52 54 55	11	10 9.1 10.1	14	4.4 4.4.3-4.4.4	46

Animations

Texte de chargement


```
.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

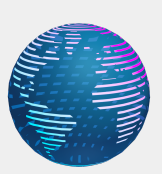

```
.editable svg path {
  stroke: purple;
  stroke-width: .1em;
  fill: none;
  /*stroke-dasharray: 0;*/
  /*stroke-dashoffset: 0;*/
  /*animation: draw 4s linear;*/
}

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

Bilan animations

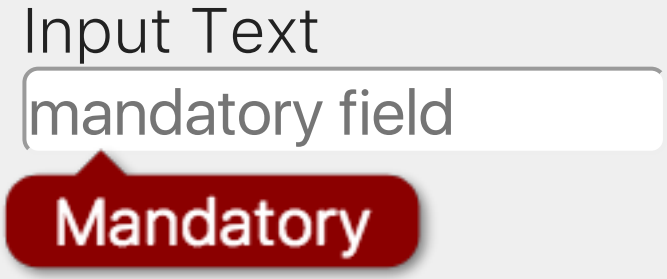
- 🦊 Utiliser les animations CSS
- ➳ Text Spinner
- ➳ CSS only loader
- 📐 Animate.css
- ✻ How SVG Line Animation Works
- 🦊 `<progress>`

Compatibilité

	 26,6 %	 18,6 %	 15,8 %	 11,5 %	 4,7 %	 4,4 %	 1,3 %	 1,0 %	 0,5 %
CSS Animation	49 58 59 60	61	9.0-9.2 9.3 10.0-10.2 10.3	48 52 54 55	11	10 9.1 10.1	14	4.4 4.4.3-4.4.4	46
SVG (basic support)	49 58 59 60	61	9.0-9.2 9.3 10.0-10.2 10.3	48 52 54 55	11	10 9.1 10.1	14	4.4 4.4.3-4.4.4	46

Pseudo classes d'état

Usage des info-bulles



➳ hover me

Pseudo états

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

Checkbox


```
.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


```
.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


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

Principe pour les onglets


```
<div class="tabs">
  <input type="radio" name="tab" id="home" checked>
  <input type="radio" name="tab" id="projects">
  <input type="radio" name="tab" id="about">
  <nav>
    <label for="home">Home</label>
    <label for="projects">Projects</label>
    <label for="about">About</label>
  </nav>
  <div data-for="home">Home page</div>
  <div data-for="projects">Projects page</div>
  <div data-for="about">About page</div>
</div>
```

Démo des onglets

Compatibilité

	26,6 %	18,6 %	15,8 %	11,5 %	4,7 %	4,4 %	1,3 %	1,0 %	0,5 %
CSS3 selectors	49 58 59 60	61	9.0-9.2 9.3 10.0-10.2 10.3	48 52 54 55	11	10 9.1 10.1	14	4.4 4.4.3-4.4.4	46
Form validation	49 58 59 60	61	9.0-9.2 9.3 10.0-10.2 10.3	48 52 54 55	11	10 9.1 10.1	14	4.4 4.4.3-4.4.4	46
CSS3 3D Transforms	49 58 59 60	61	9.0-9.2 9.3 10.0-10.2 10.3	48 52 54 55	11	10 9.1 10.1	14	4.4 4.4.3-4.4.4	46

HTML

Panel


```html
<details>
  <summary>Des détails</summary>
  <p>Plus d'infos
    à propos des détails.</p>
</details>
```

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

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

Dialog


```
.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
- 📐 Dialog Polyfill

Compatibilité

									
	26,6 %	18,6 %	15,8 %	11,5 %	4,7 %	4,4 %	1,3 %	1,0 %	0,5 %
Details & Summary elements	49 58 59 60	61	9.0-9.2 9.3 10.0-10.2 10.3	48 52 54 55	11	10 9.1 10.1	14	4.4 4.4.3-4.4.4	46
Dialog element	49 58 59 60	61	9.0-9.2 9.3 10.0-10.2 10.3	48 52 54 55	11	10 9.1 10.1	14	4.4 4.4.3-4.4.4	46
progress element	49 58 59 60	61	9.0-9.2 9.3 10.0-10.2 10.3	48 52 54 55	11	10 9.1 10.1	14	4.4 4.4.3-4.4.4	46

Conclusion

Bilan

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

👉 Traitez le CSS comme du code

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

Liens

- 📐 les slides en HTML
- 📐 les slides en PDF
- 📐 le code
- 🐵 Making Of

Pour apprendre

- `(Ctrl|⌘) + Shift + i`
- 

➳ CSS Secret by Lea Verou
- 🦊 CSS sur MDN
- ➳ CodePen , ➳ JSFiddle , ➳ Dabblet ,...
- ✻ CSS Tricks , 𝓢 Smashing Magazine
- 📐 CSS Flags