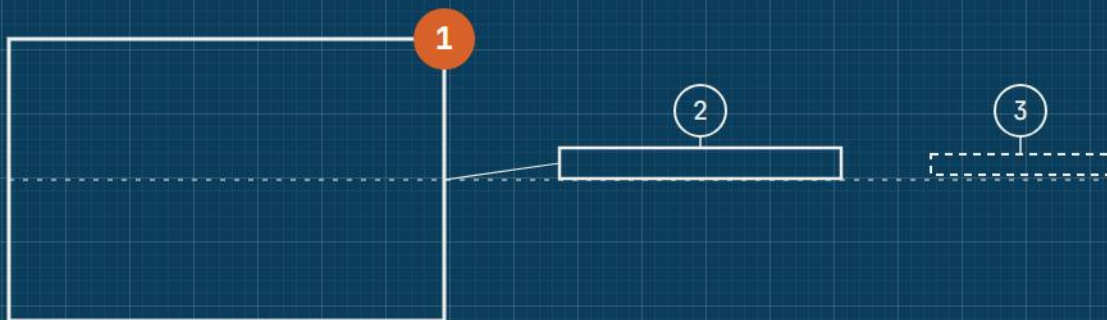


Bloc Laravel · chapitre 13

Tailwind CSS

Une interface propre et moderne, sans écrire de feuille de style : les pages de la Ludothèque des Tilleuls.



Une feuille de style écrite à la main décrit une classe inventée

Le catalogue affiche chaque jeu dans une carte : un cadre blanc, un titre, une ligne grise.

feuille de style

```
1 | .game-card {
2 |     padding: 1.5rem;
3 |     background-color: white;
4 |     border: 1px solid #e4e4e7;
5 |     border-radius: 0.5rem;
6 |     box-shadow: 0 1px 2px 0 rgb(0 0 0 /
   |     0.05);
7 | }
8 |
9 | .game-card h2 { ... }
10| .game-card p { ... }
```

vue

```
1 | <article class="game-card">
2 |     <h2>Les Comptoirs de
   |     Safran</h2>
3 |     <p>2 à 4 joueurs · 90 min</p>
4 | </article>
```

Le nom game-card relie les deux fichiers : pour lire le style de la carte, vous ouvrez la feuille de style.

Avec Tailwind, chaque balise reçoit des classes déjà prêtes

vue

```
1 <article class="rounded-lg border border-zinc-200 bg-white ① p-6 shadow-xs">
2   <h2 class="② text-lg font-semibold">Les Comptoirs de Safran</h2>
3   <p class="③ mt-2 text-zinc-600">2 à 4 joueurs • 90 min</p>
4 </article>
```

- ① La marge intérieure de la carte.
- ② La taille du texte du titre.
- ③ La marge au-dessus du paragraphe.

Chaque classe applique une seule propriété CSS.
Une classe de ce genre s'appelle une **classe utilitaire**.

navigateur

Les Comptoirs de Safran

2 à 4 joueurs • 90 min

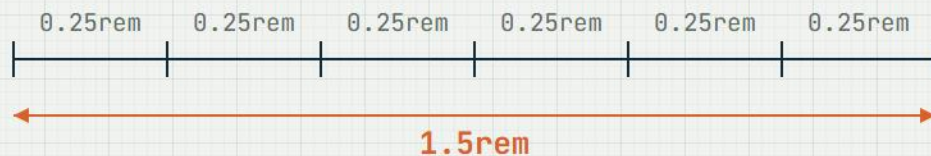
Tailwind écrit une règle CSS pour chaque classe trouvée dans vos vues

Tailwind lit vos vues, y trouve ces noms, et écrit pour chacun une règle dans le fichier CSS qu'il produit.

CSS produit · extrait

```
.p-6{padding:calc(var(--spacing) * 6)}  
.mt-2{margin-top:calc(var(--spacing) * 2)}  
.text-lg{font-size:var(--text-lg);line-height:var(--tw-leading,var(--text-lg--line-height))}
```

La variable `--spacing` vaut `0.25rem`. `p-6` donne donc $6 \times 0,25 \text{ rem}$, soit le `1.5rem` de la feuille de style. Vous choisissez dans une échelle au lieu d'écrire des nombres.

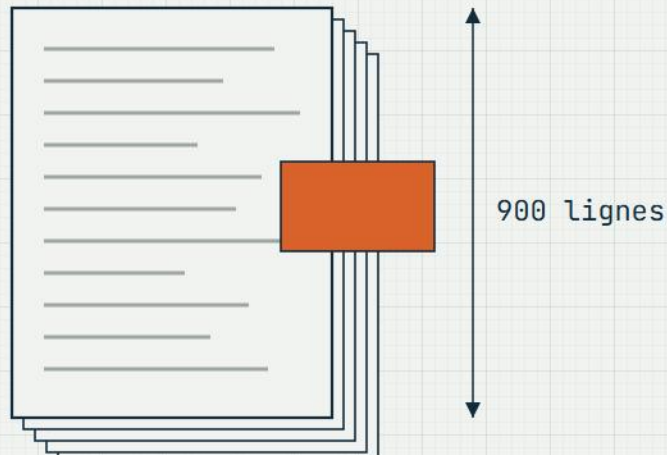


Une ludothèque veut retirer deux règles de sa feuille de style de 900 lignes

En binôme, cinq minutes. La page des prêts a été réécrite le mois dernier.

feuille de style · extrait

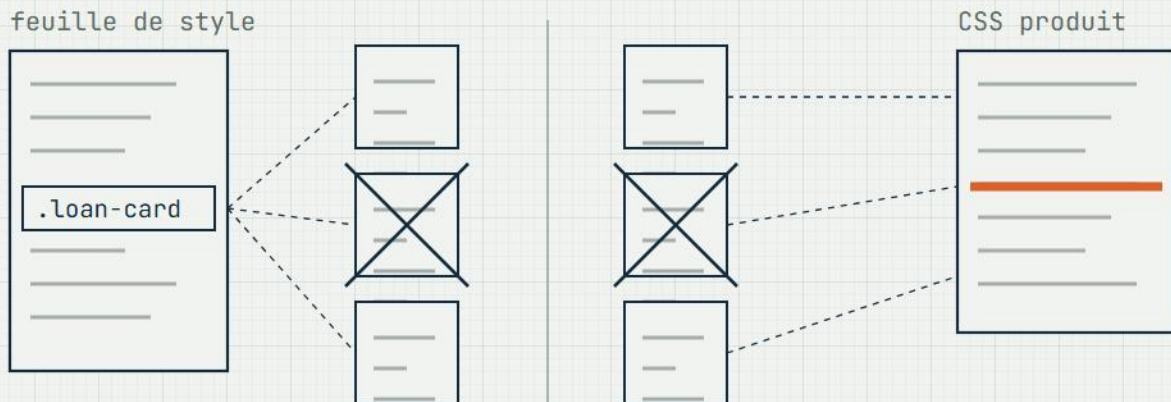
```
1 | .loan-card {  
2 |     padding: 1rem;  
3 |     border-left: 4px solid #b91c1c;  
4 | }  
5 |  
6 | .loan-card .late {  
7 |     color: #b91c1c;  
8 |     font-weight: 700;  
9 | }
```



Pouvez-vous supprimer ces deux règles ? Comment en êtes-vous sûrs ?

Avec Tailwind, les classes d'une vue disparaissent avec elle

La feuille de style ne dit pas quelles vues utilisent `.loan-card` : dans le doute, personne ne supprime la règle. Avec Tailwind, le style est écrit dans la vue. Quand la vue disparaît, ses classes disparaissent du prochain fichier CSS produit.



Cette façon de travailler s'appelle **utility-first** : vous partez des classes utilitaires, et n'écrivez du CSS que pour le reste.

resources/css/app.css charge Tailwind en une ligne

Tailwind est déjà installé dans tout projet créé avec `laravel new` : les paquets `tailwindcss` et `@tailwindcss/vite`, et leur extension dans `vite.config.js`.

resources/css/app.css

```
① 1 @import 'tailwindcss';  
2  
3 @source '../vendor/laravel/framework/src/Illuminate/Pagination/resources/views/*.blade.php';  
② 4 @source '../storage/framework/views/*.php';  
5  
③ 6 @theme {  
7     --font-sans: 'Instrument Sans', ui-sans-serif, system-ui, sans-serif, 'Apple Color Emoji', 'Segoe UI  
8         Emoji',  
9         'Segoe UI Symbol', 'Noto Color Emoji';  
7 }  
8  
9 }
```

- ① Charge Tailwind : remise à zéro, échelle, classes utilitaires.
- ② Ajoute des dossiers que Git ignore.
- ③ Des **variables de thème**, qui créent des classes.

Tailwind lit vos fichiers comme du texte : vos vues et votre JavaScript sont lus sans aucune ligne à ajouter.

@vite écrit les balises qui chargent le CSS et le JavaScript

Le navigateur ne charge pas app.css tel quel : il lui faut le fichier que Vite produit à partir de lui.

resources/views/components/layouts/app.blade.php

```
1 <head>
2     <meta charset="UTF-8">
3     <meta name="viewport" content="width=device-width, initial-scale=1">
4     <title>{{ $title }} · Ludothèque des Tilleuls</title>
5     @fonts
6     @vite(['resources/css/app.css', 'resources/js/app.js'])
7 </head>
```

- ① Le téléphone affiche la page à sa vraie largeur.
- ② Charge la police Instrument Sans.
- ③ Les deux sources de vite.config.js, une balise chacune.

Le code source montre un <link> et un <script> vers public/build/

Ouvrez le catalogue, puis affichez son code source.

navigateur · code source du catalogue

```
<title>Catalogue · Ludothèque des Tilleuls</title>
...
<link rel="preload" as="style"
href="http://tailwind-26.test/build/assets/app-D32E-bvF.css" /><link rel="modulepreload"
as="script" href="http://tailwind-26.test/build/assets/app-C5TGVq_p.js" /><link
rel="stylesheet" href="http://tailwind-26.test/build/assets/app-D32E-bvF.css" /><script
type="module" src="http://tailwind-26.test/build/assets/app-C5TGVq_p.js"></script></head>
```

La suite D32E-bvF change à chaque fois que le contenu du fichier change. Le navigateur ne peut donc pas garder en mémoire une ancienne version.

npm run build écrit les fichiers une fois, puis s'arrête

terminal

```
$ npm run build
```

```
...
```

```
public/build/manifest.json
```

```
1.47 kB | gzip: 0.33 kB
```

```
...
```

```
public/build/assets/app-D32E-bvF.css
```

```
29.46 kB | gzip: 6.28 kB
```

```
public/build/assets/app-C5TGVq_p.js
```

```
0.24 kB | gzip: 0.17 kB
```

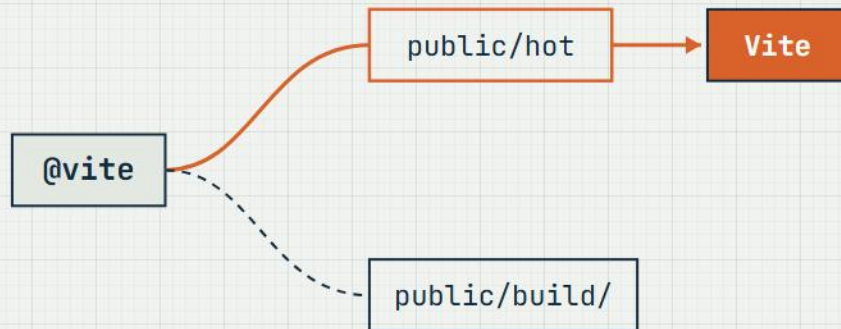
```
✓ built in 801ms
```

Le fichier CSS ne contient que les classes trouvées au moment du build. Une classe écrite ensuite dans une vue, comme `bg-red-50`, n'a aucun effet : aucune règle ne lui correspond dans `public/build/`.

npm run dev ajoute la règle de chaque nouvelle classe pendant le travail

terminal

```
$ npm run dev
...
VITE v8.3.2 ready in 319 ms
...
LARAVEL v13.34.0 plugin v3.2.0
```



La commande ne rend pas la main : Vite surveille vos fichiers et recharge la page. Il écrit son adresse dans public/hot, et @vite désigne alors Vite au lieu de public/build/. **Pendant que vous travaillez, laissez tourner npm run dev dans un terminal.**

Sans `public/build/`, la page répond « Vite manifest not found »

Dans un projet cloné avec Git, le dossier `public/build/` n'existe pas : il est exclu de Git. Une page qui utilise `@vite` renvoie une erreur 500.

```
navigateur · erreur 500
```

```
Vite manifest not found at: .../public/build/manifest.json
```

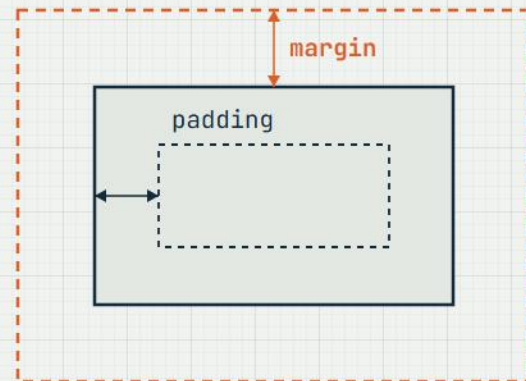
NON CONFORME

Le **manifeste** associe chaque source à son fichier produit, comme `app.css` à `app-D32E-bvF.css`. Lancez `npm install`, puis `npm run build` ou `npm run dev`.

mx-auto max-w-6xl centre le contenu dans un bloc de largeur limitée

vue · le contenu d'une page

```
1 | <main class="① mx-auto max-w-6xl px-4 py-8">
2 |   <h1 class="text-3xl font-semibold
   |   tracking-tight">Catalogue</h1>
3 |   <div class="mt-6 ② space-y-4">
4 |     ...
5 |   </div>
6 | </main>
```



- ① Largeur limitée à 72 rem, reste partagé des deux côtés.
- ② Écarte chaque enfant de son voisin du dessus.

La **marge intérieure**, padding, sépare le bord du contenu. La **marge extérieure**, margin, sépare l'élément de ses voisins. Le nombre d'une classe d'espacement est un multiple de 0.25rem, soit 4 pixels.

Une classe de couleur nomme ce qui est coloré, une teinte et une nuance

vue · alerte de la page des prêts

```
1 <p class="rounded-md bg-red-50 px-4 py-3 text-sm text-red-800">  
2     Deux prêts sont en retard.  
3 </p>
```



`bg-red-50` colore le fond, `text-red-800` le texte. Chaque teinte existe en onze nuances, de 50, presque blanc, à 950, presque noir. La ludothèque utilise `zinc`, un gris neutre, pour le texte, les fonds et les bordures.

Sur fond blanc, text-zinc-400 n'atteint que 2,6 pour 1

Un texte courant demande un contraste d'au moins 4,5 pour 1 avec son fond.

Classe	Contraste sur fond blanc	Exemple
text-zinc-400	2,6 pour 1	2 à 4 joueurs · 90 min
text-zinc-500	4,8 pour 1	2 à 4 joueurs · 90 min
text-zinc-600	7,7 pour 1	2 à 4 joueurs · 90 min

text-zinc-400 paraît lisible sur un écran neuf, mais plus sur un vieil écran ni en plein soleil.

--color-brand-600 dans @theme crée bg-brand-600 et ses voisines

La ludothèque a sa couleur, un bleu canard, qui ne fait pas partie des teintes de Tailwind. Elle la déclare en onze nuances.

resources/css/app.css

```
1 | @theme {  
2 |     ...  
3 |     --color-brand-50: oklch(0.98 0.016 195);  
4 |     ...  
5 |     --color-brand-600: oklch(0.54 0.105 198);  
6 |     ...  
7 |     --color-brand-950: oklch(0.25 0.045 204);  
8 | }
```

CSS produit · extrait

```
.bg-brand-600{background-color:var(--color-brand-600)}
```

Les vues écrivent brand, le rôle de la couleur. Si la ludothèque change de couleur, vous modifiez ces onze lignes, et tout le site suit.

flex justify-between pousse le jeu à gauche et le statut à droite

vue · une ligne de prêt

```
1 <div class="① flex items-center justify-between gap-4 border-b border-zinc-200 py-3">
2   <div>
3     <p class="font-medium">Marée d'étoiles</p>
4     <p class="text-sm text-zinc-600">Emprunté par Amina Benali, à rendre le 12 octobre</p>
5   </div>
6   <span class="② shrink-0 text-sm font-medium text-red-700">En retard</span>
7 </div>
```

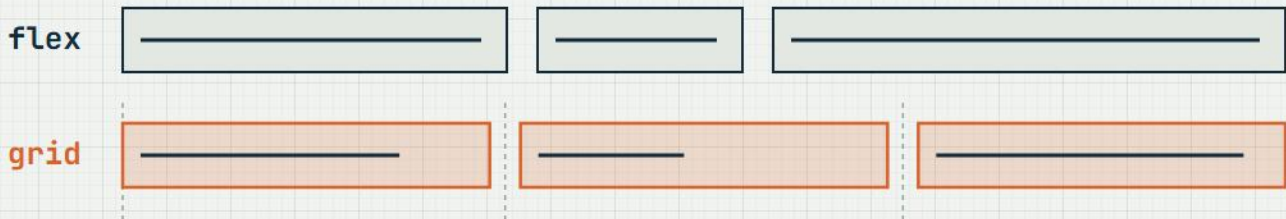
- ① Deux enfants côte à côte, aux deux bords, centrés en hauteur.
- ② Le statut ne rétrécit jamais : la phrase passe à la ligne.



Avec Grid, la largeur d'une carte dépend de la colonne, pas de son titre

vue · le catalogue

```
1 <ul class="grid grid-cols-3 gap-6">
2   <li><article class="rounded-lg border border-zinc-200 bg-white p-6 shadow-xs">...</article></li>
3   <li><article class="rounded-lg border border-zinc-200 bg-white p-6 shadow-xs">...</article></li>
4   <li><article class="rounded-lg border border-zinc-200 bg-white p-6 shadow-xs">...</article></li>
5 </ul>
```



Une grille découpe la largeur en colonnes, et range les enfants de gauche à droite, puis ligne par ligne. Avec Flex, la largeur d'un enfant dépend de son contenu.

sm:grid-cols-2 s'applique à partir de 640 pixels, pas sur un petit écran

```
resources/views/games/index.blade.php
```

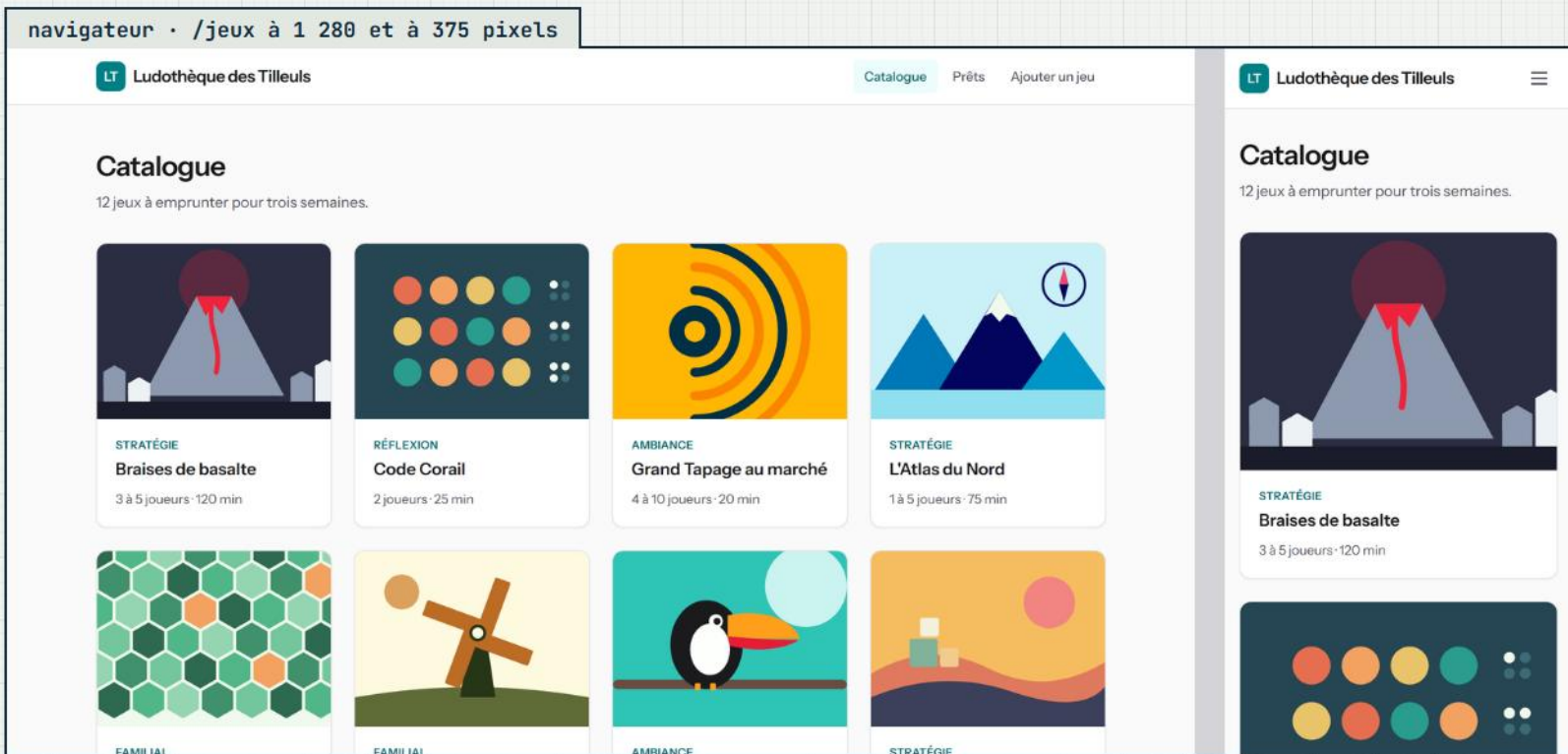
```
1 | <ul class="grid gap-6 sm:grid-cols-2 lg:grid-cols-3 xl:grid-cols-4">
```

Préfixe	À partir de	Écran typique
aucun	0 pixel	téléphone
sm:	640 pixels (40 rem)	grand téléphone couché
md:	768 pixels (48 rem)	tablette
lg:	1 024 pixels (64 rem)	ordinateur portable
xl:	1 280 pixels (80 rem)	écran de bureau
2xl:	1 536 pixels (96 rem)	grand écran

grid seul range les cartes sur une colonne, à toutes les largeurs. Chaque préfixe ajoute ce qui change à partir de sa largeur.

Mobile first : la page du téléphone s'écrit d'abord, sans préfixe

Vous écrivez d'abord la page du téléphone, puis vous ajoutez ce qui change sur un écran plus large. Cette façon de faire s'appelle **mobile first**.



hover:, focus-visible: et disabled: ne s'appliquent que dans leur état

vue · un bouton

```
1 | <button type="submit"  
2 |     class="rounded-lg bg-brand-600 px-4 py-2 text-sm font-semibold text-white  
   |     ① hover:bg-brand-700 ② focus-visible:outline-2 focus-visible:outline-offset-2  
   |     focus-visible:outline-brand-600 disabled:cursor-not-allowed ③ disabled:opacity-50">  
3 |     Ajouter le jeu  
4 | </button>
```

- ① Au survol de la souris.
- ② Au focus reçu par le clavier.
- ③ Quand le bouton porte l'attribut disabled.

Le **focus** est l'état de l'élément qui reçoit les touches du clavier. La touche Tab le fait passer d'un lien ou d'un bouton au suivant.

Ne retirez jamais le contour de focus sans le remplacer

Beaucoup d'exemples écrivent `outline-none` sur les boutons et les champs. Une personne qui navigue au clavier ne sait alors plus où elle se trouve dans la page. Si vous changez ce contour, écrivez-en un autre avec `focus-visible:`, puis parcourez la page avec la touche Tab.

~~outline-none~~

NON CONFORME

navigateur • /composants, focus au clavier

Enregistrer

Annuler

Supprimer

Indisponible

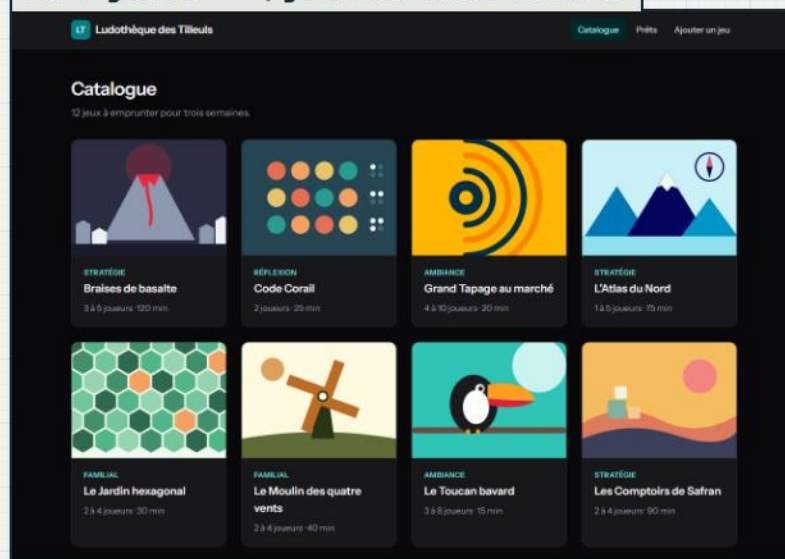
dark : applique une classe quand le système est réglé en mode sombre

resources/views/components/layouts/app.blade.php

```
1 | <body class="flex min-h-screen flex-col bg-zinc-50 font-sans text-zinc-900 antialiased  
   |   dark:bg-zinc-950 dark:text-zinc-100">
```

En mode sombre, le fond et le texte s'inversent. Chaque classe de couleur demande sa version sombre : ajoutez les classes `dark:` partout, ou nulle part.

navigateur · /jeux en mode sombre



Le composant card écrit une seule fois les classes de toutes les cartes

Recopier ces listes dans chaque vue obligerait à ouvrir toutes les vues au moindre changement de style.

resources/views/components/card.blade.php

```
1 <article {{ ① $attributes->class([
2     'overflow-hidden rounded-xl border
3     border-zinc-200 bg-white shadow-xs',
4     'dark:border-zinc-800 dark:bg-zinc-900',
5 ]) }}>
6     @isset($image)
7         {{ $image }}
8     @endisset
9     <div class="p-5">
10         {{ $slot }}
11     </div>
12 </article>
```

- ① Les classes du tableau, puis celles de la balise <x-card>.
- ② Le slot image, seulement si la page en donne un.
- ③ Écarte le texte des bords, mais pas l'image.

Les classes écrites dans <x-card> s'ajoutent à celles du composant

resources/views/games/index.blade.php

```
1 | @foreach ($games as $game)
2 |     <li>
3 |         <x-card class="h-full transition-shadow
4 |             hover:shadow-md">
5 |             <x-slot:image>
6 |                 
9 |             </x-slot:image>
10 |             ...
11 |         </x-card>
12 |     </li>
13 | @endforeach
```

navigateur • /jeux

Catalogue

12 jeux à emprunter pour trois semaines.



STRATÉGIE
Braises de basalte
3 à 5 joueurs · 120 min



RÉFLEXION
Code Corail
2 joueurs · 25 min



AMBIANCE
Grand Tapage au marché
4 à 10 joueurs · 20 min



STRATÉGIE
L'Atlas du Nord
1 à 5 joueurs · 75 min

h-full donne la même hauteur à toutes les cartes d'une ligne. **aspect-4/3** et **object-cover** recadrent l'image sans la déformer.

@class n'écrit une clé que si sa condition est vraie

resources/views/components/badge.blade.php

```
1 | @props(['status'])
2 |
3 | <span @class([
4 |     'inline-flex items-center rounded-full px-2.5 py-0.5 text-xs font-medium whitespace-nowrap ring-1
      ring-inset',
5 |     'bg-sky-50 text-sky-700 ring-sky-600/20 dark:bg-sky-400/10 dark:text-sky-300 dark:ring-sky-400/30' =>
      $status === 'ongoing',
6 |     'bg-red-50 text-red-700 ring-red-600/20 dark:bg-red-400/10 dark:text-red-300 dark:ring-red-400/30' =>
      $status === 'overdue',
7 |     ...
8 | ])>
9 |     {{ $slot }}
10 | </span>
```

Une valeur sans clé est toujours écrite. Une clé n'est écrite que si sa valeur vaut true. Le modèle Loan calcule le statut à partir des dates du prêt.

navigateur · /composants

En cours

En retard

Rendu

bg-{{ \$color }}-100 ne produit aucune règle CSS

resources/views/components/tag.blade.php

```
1 | @props(['color'])
2 |
3 | <span class="rounded-md bg-{{ $color }}-100 px-2 py-1 text-xs text-{{ $color }}-800">
4 |     {{ $slot }}
5 | </span>
```

terminal

```
$ npm run build
...
$ grep -c "bg-lime-100" public/build/assets/app-*.css
0
```

NON CONFORME

Avec `color="lime"`, le HTML contient bien `bg-lime-100`, mais l'étiquette n'a pas de fond vert. Tailwind lit `tag.blade.php` comme du texte, sans exécuter Blade : il y trouve `bg-{{ $color }}-100`.

Écrivez toujours les noms de classes en entier dans vos fichiers

resources/views/components/tag.blade.php

```
1 | @props(['color'])
2 |
3 | <span @class([
4 |     'rounded-md px-2 py-1 text-xs',
5 |     'bg-lime-100 text-lime-800' => $color ===
6 |     'lime',
7 |     'bg-sky-100 text-sky-800' => $color ===
8 |     'sky',
9 | ])>
10 |     {{ $slot }}
11 | </span>
```

terminal

```
$ grep -c "bg-lime-100"
public/build/assets/app
1
```

VÉRIFIÉ

La règle vaut aussi pour PHP et JavaScript : 'bg-' . \$color . '-100' dans un contrôleur ne produit pas de règle CSS.

Le composant button choisit ses classes selon sa prop variant

Le bouton existe en trois **variantes**, trois aspects choisis par une prop : principal, secondaire et danger.

resources/views/components/button.blade.php

```
1 | @props(['variant' => 'primary'])
2 |
3 | <button {{ $attributes① ->merge(['type' => 'button'])② ->class([
4 |     'inline-flex items-center justify-center gap-2 rounded-lg px-4 py-2 text-sm font-semibold shadow-xs
      transition-colors',
5 |     ...
6 |     'bg-brand-600 text-white hover:bg-brand-700 focus-visible:outline-brand-600 ③ disabled:hover:bg-brand-600' =>
      $variant === 'primary',
7 |     ...
8 | ]) }}>
9 |     {{ $slot }}
10 | </button>
```

- ① Le type button par défaut, sinon il envoie le formulaire.
- ② Le tableau de @class, plus les classes de la balise.
- ③ Désactivé, il ne change pas au survol.

Chaque `<x-button>` garde ses propres attributs

vue

```
1 <x-button>Emprunter</x-button>
2 <x-button variant="danger" type="submit">Supprimer le jeu</x-button>
3 <x-button variant="secondary" class="w-full" disabled>Indisponible</x-button>
```

Le second bouton garde `type="submit"`, écrit dans la balise. Le troisième reçoit `w-full` et l'attribut `disabled`. Chaque bouton principal utilise `brand-600` : si la ludothèque change de couleur, aucune vue ne change.

navigateur

Emprunter

Supprimer le jeu

Indisponible

Sous 768 pixels, le menu se cache derrière un bouton

resources/views/components/layouts/app.blade.php · extrait

```
1 | <button
2 |     type="button"
3 |     id="menu-button"
4 |     class="group rounded-md p-2 text-zinc-600 hover:bg-zinc-100 hover:text-zinc-900
      focus-visible:outline-2 focus-visible:outline-brand-600 ① md:hidden dark:text-zinc-300
      dark:hover:bg-zinc-800 dark:hover:text-white"
5 |     aria-controls="main-menu"
6 |     aria-expanded="false"
7 | >
8 | ...
9 | <nav id="main-menu" ② class="hidden w-full md:block md:w-auto" aria-label="Navigation principale">
```

- ① Le bouton disparaît à partir de 768 pixels.
- ② Le menu, caché sur un téléphone, affiché ensuite.

aria-expanded dit au lecteur d'écran et au dessin si le menu est ouvert

Un **lecteur d'écran** lit la page à voix haute aux personnes aveugles ou malvoyantes. `sr-only` cache le mot « Menu » à l'écran, mais le lecteur l'annonce.

`group-aria-expanded:hidden` cache les trois traits quand le bouton porte `aria-expanded="true"`, et la croix s'affiche.

navigateur · 1 280 pixels

LT Ludothèque des Tilleuls

Catalogue Prêts Ajouter un jeu

navigateur · 375 pixels, menu ouvert



Ludothèque des Tilleuls



Catalogue

Prêts

Ajouter un jeu

Prêts

15 prêts, du plus récent au plus ancien.

Jeu

Membre

[Le Toucan bavard](#)

Yasmine El Idrissi

resources/js/app.js ouvre et ferme le menu à chaque clic

resources/js/app.js

```
1 // Le menu mobile : le bouton burger affiche ou cache la liste des liens.
2 const menuButton = ① document.querySelector('#menu-button');
3 const menu = document.querySelector('#main-menu');
4
5 if (menuButton && menu) {
6     menuButton.addEventListener('click', () => {
7         const isOpen = menuButton.getAttribute('aria-expanded') === 'true';
8
9         menuButton.setAttribute('aria-expanded', String(!isOpen));
10        menu.classList.③toggle('hidden');
11    });
12 }
```

① L'élément dont l'id vaut menu-button, ou null.

② Écrit l'état inverse.

③ Retire hidden s'il y est, l'ajoute sinon.

Tailwind lit aussi app.js : une classe que seul votre JavaScript ajoute doit y être écrite en entier, comme 'hidden'.

Un exemple écrit pour Tailwind 3 se vérifie avant d'être copié

La documentation, tailwindcss.com/docs, ouvre sa recherche avec Ctrl+K : tapez la propriété CSS en anglais. L'extension **Tailwind CSS IntelliSense** complète les noms de classes dans VS Code.

Tailwind 3	Tailwind 4	L'ancien nom en Tailwind 4
un fichier <code>tailwind.config.js</code>	<code>@theme</code> dans <code>app.css</code>	le fichier n'est plus lu
<code>@tailwind base;</code> et deux lignes voisines	<code>@import 'tailwindcss';</code>	ne charge plus Tailwind en entier
<code>bg-opacity-50</code>	<code>bg-black/50</code>	aucun effet
<code>shadow-sm</code>	<code>shadow-xs</code>	une ombre plus marquée
<code>ring</code>	<code>ring-3</code>	un contour de 1 pixel
<code>border</code>	<code>border border-zinc-200</code>	une bordure de la couleur du texte

Au chapitre 14, vous stylez la ludothèque dans tailwind-26

- 1 Une classe utilitaire applique une seule propriété CSS.
- 2 `npm run dev` tourne pendant le travail.
- 3 Sans préfixe pour le téléphone, puis `md:` ou `lg:`.
- 4 Les listes de classes vont dans un composant Blade.
- 5 Un nom de classe s'écrit toujours en entier.

```
npm install, puis npm run dev
```

