

TAILWIND

Manuel MARIE
DWWM 2025



```
index.html">Home</a></li>
"home-events.html">Home Events</a></li>
"multi-col-menu.html">Multiple Column
"has-children"> <a href="#" class="curr
li>a href="tall-button-header.html">Ta
li>a href="image-logo.html">Image Log
li class="active"><a href="tall-logo.h
"has-children"> <a href="#">Carouse
li>a href="variable-width-slider.h
li>a href="testimonial-slider.html
li>a href="featured-work-slider.h
li>a href="equal-column-slider.h
li>a href="video-slider.html">Video
li>a href="mini-bootstrap-carous
```

SOMMAIRE

- 1 Définition et installation
- 2 Texte et couleurs
- 3 Box-sizing
- 4 Background
- 5 Mise en page (flexbox / grid)
- 6 Responsive design
- 7 Customisation

1. Définition et installation

1
Définition et installation

2
Texte et couleurs

3
Box-sizing

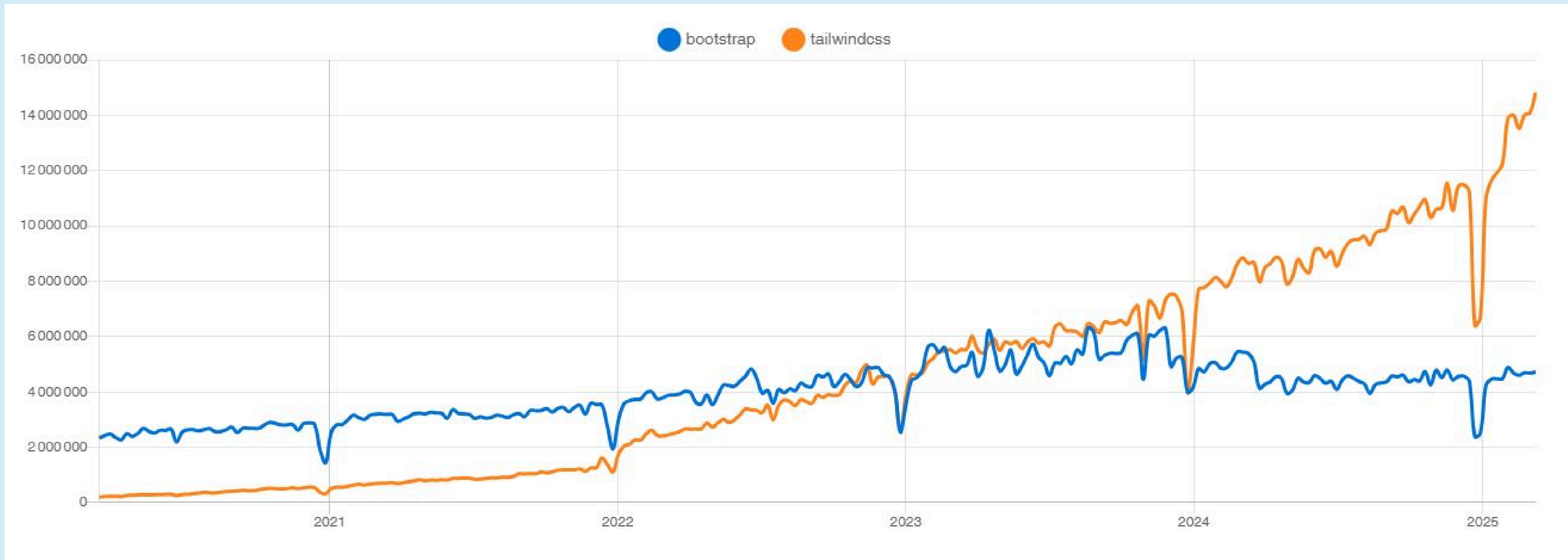
4
Background

5
Mise en page (flexbox / grid)

6
Responsive design

7
Customisation

UTILISATION DE FRAMEWORK CSS



Statistique de npm trends

DÉFINITION FRAMEWORK

Framework = cadre d'applications

Ensemble de fichiers dont les règles de styles sont déjà écrites.

Permet une accélération du processus de création de conception web.

TAILWIND CSS

Framework CSS *utility-first*. Pas de composant.

- Pas d'aller-retour entre le HTML et le CSS.
- Créer ses propres composants
- Développer la personnalisation pour concevoir un site web
- Système de purge CSS
- Fichier très léger, favorise la vitesse de chargement du site sweb

PRÉ-REQUIS

Pour utiliser Tailwind CSS, il faut :

- Visual Studio Code, avec comme extensions :
 - Live Server (rafraîchissement automatique)
 - Tailwind CSS IntelliSense (pour l'autocomplétion)

PRÉ-INSTALLATION

Pour installer Tailwind CSS, nous allons utiliser Node.js.

Une fois Node.js installé, nous pourrons utiliser *npm* (*Node Package Manager*), afin d'installer des dépendances comme des librairies ou des outils.

Installer Node.js (via fnm) et vérifiez la version de Node.js et de npm.

```
1 # Télécharger et installer fnm :  
2 winget install Schniz.fnm  
3  
4 # Télécharger et installer Node.js :  
5 fnm install 22  
6  
7 # Vérifier la version de Node.js :  
8 node -v # Doit afficher "v22.14.0".  
9  
10 # Vérifier la version de npm :  
11 npm -v # Doit afficher "10.9.2".
```

INSTALLATION

Ouvrez le terminal dans VSC (ctrl + maj + ù) et lancer ces commandes :

- **npm install tailwindcss @tailwindcss/cli**

Si problème d'autorisation :

- **Set-ExecutionPolicy Unrestricted -Scope CurrentUser -Force**

INSTALLATION

Créez votre fichier index.html, et vos fichiers css.

Puis dans votre css, ajoutez Tailwind :

```
@import "tailwindcss";
```

Enfin, lancez votre compilation et générer la feuille de style de base de Tailwind :

```
npx @tailwindcss/cli -i ./src/input.css -o ./src/output.css --watch
```

Appelez le CSS dans votre page html, et vérifiez le bon fonctionnement de Tailwind.

Space

Nous pouvons ajouter des espacements entre les éléments enfants :

space-{x|y}-{valeur}

Elle doit être ajoutée au niveau de l'élément parent.

- **x** permet une répartition horizontale
- **y** permet une répartition verticale
- **valeur** : nombre entre 0 et 96 (parmi les 34 valeurs de Tailwind)

Bordures

border-{côté}-{taille}

border seule permet d'ajouter une bordure tout autour de l'élément avec une épaisseur de 1px.

L'attribut **côté** est facultatif. Il prend les valeurs :

- **t**
- **b**
- **l**
- **r**
- **x**
- **y**

Attention ! L'unité pour la bordure est en px : 0, 2, 4 ou 8.

Principes de base

Bouton

Faire un bouton en CSS :

```
.btn {  
    background-color: #3b82f6;  
    color: #fff;  
    font-size: 18px;  
    padding: 8px 16px;  
    border: none;  
    border-radius: 8px;  
}
```

Faire un bouton avec Bootstrap :

```
<a href="#" class="btn btn-primary">Bouton</a>
```

Faire un bouton avec Tailwind :

```
<a href="#" class="bg-blue-500 text-white px-4 py-2 text-lg rounded-lg w-fit">Bouton</a>
```

TAILWIND PLAY

Si vous avez un problème pour l'installation ou le fonctionnement de Tailwind, tous les exercices et cas pratique sont réalisable en ligne, sur le Tailwind Play :

<https://play.tailwindcss.com/>

DOCUMENTATION

Toutes les classes, fonctionnalités et détails sont disponible sur :

- <https://tailwindcss.com/>

Un petit C T R L + K et vous avez une recherche disponible pour ce que vous souhaitez.

Attention, la plupart des classes sont en anglais

2. Texte et couleurs

- 1 Définition et installation
- 2 Texte et couleurs
- 3 Box-sizing
- 4 Background
- 5 Mise en page (flexbox / grid)
- 6 Responsive design
- 7 Customisation

Couleurs

Tailwind propose une large palette de couleurs :

- 22 couleurs de bases
- 10 nuances par couleur

Pour l'utiliser, il faut utiliser la syntaxe suivante :

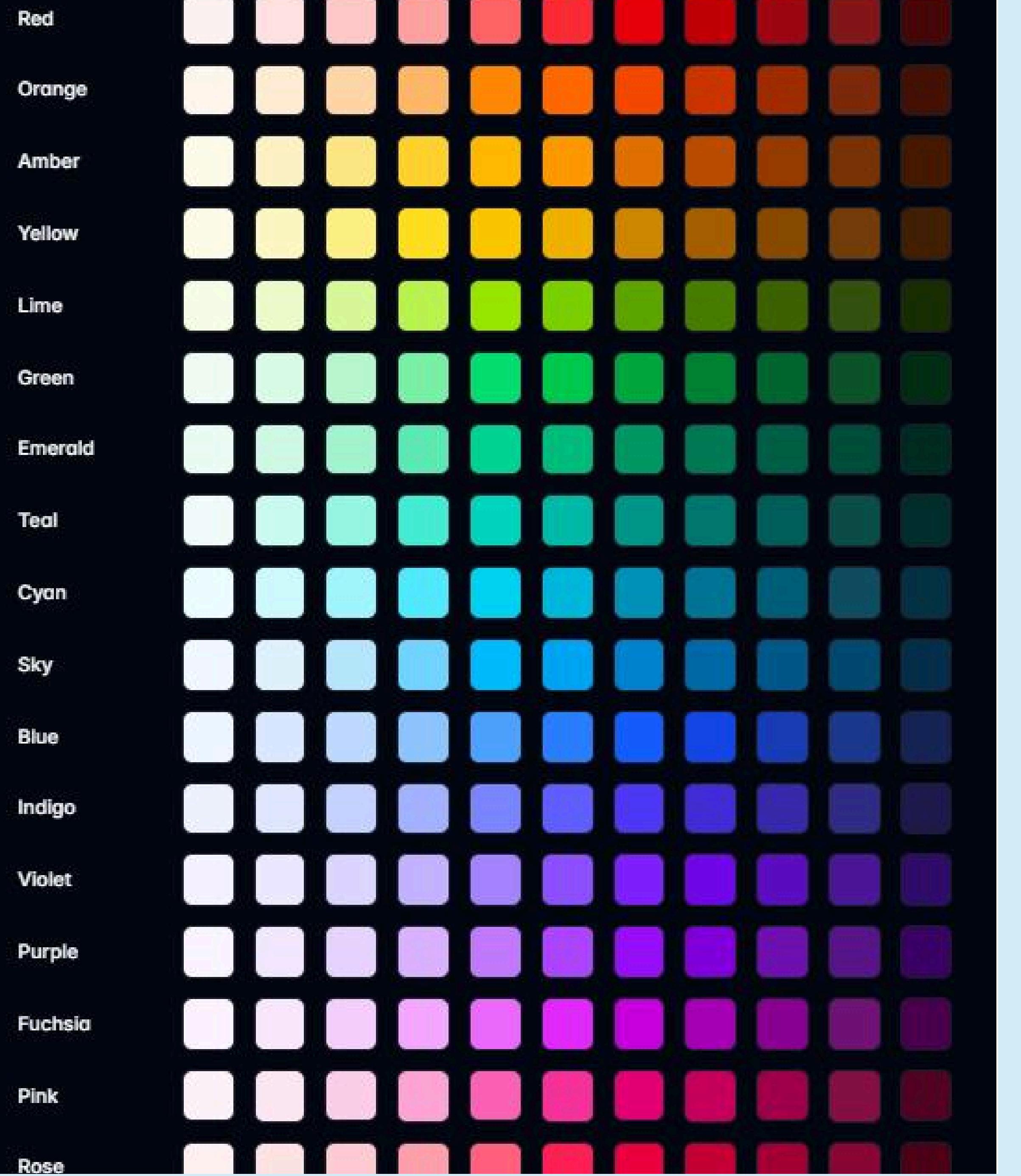
text-{couleur}-{nuance}

Nuances

Les nuances prennent les valeurs suivantes :

- **50**
- **100**
- **200**
- **300**
- **400**
- **500**
- **600**
- **700**
- **800**
- **900**

50 100 200 300 400 500 600 700 800 900 950



Green

Emerald

Teal

Cyan

Sky

Blue

Indigo

Violet

Purple

Fuchsia

Pink

Rose

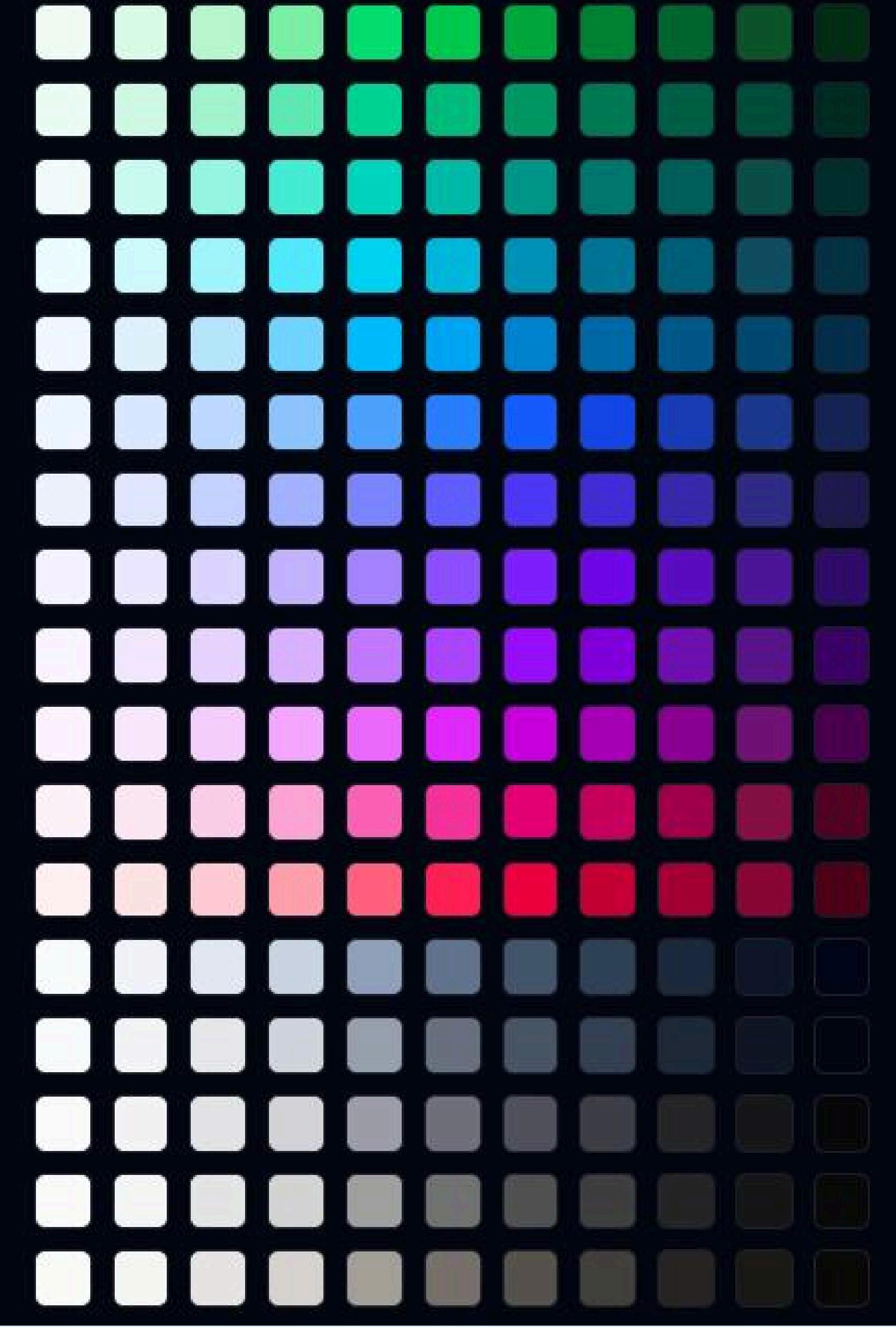
Slate

Gray

Zinc

Neutral

Stone



Texte

Pour le texte, il existe des classes spécifiques pour les couleurs :

- **text-current** : prend la couleur héritée de parent
- **text-transparent**
- **text-black**
- **text-white**

L'ajout de couleur arbitraire reste possible :
text-[#123456]

Pour l'opacité, il suffit d'ajouter “ / “ :

- **text-blue-600/100** : opacité à 100%
- **text-blue-600/75** : opacité à 75%
- **text-blue-600/50** : opacité à 50%
- **text-blue-600/25** : opacité à 25%
- **text-blue-600/0** : opacité à 0

La combinaison est également possible :

text-blue-600/[0.6] : opacité à 60%

Ajout / opacité

Texte

Pour modifier la taille de la police “text-{taille}” avec :

- **xs**
- **sm**
- **base**
- **lg**
- **xl**
- **2 à 9xl**

Interligne

L’interlignage est automatique, mais on peut le modifier avec “leading-{valeur}” :

- **1-10** : ratio par rapport à la taille de la police
- value (en em, rem, px etc...)
- **none** : line-height à 1

Italic / Bold / soulignement

Pour l'*italique*, tout simplement : italic / not-italic

Pour le bold :

- font-thin (100)
- font-extralight (200)
- font-light (300)
- font-normal (400)
- font-medium (500)
- font-semi-bold (600)
- font-bold (700)
- font-extrabold (800)
- font-black (900)

Pour le soulignement : underline/overline/line-through

Alignment (texte)

Alignment horizontal :

- text-left (start)
- text-center
- text-right (left)
- text-justify

Alignment vertical :

- align-top
- align-middle
- align-bottom

Font

Tailwind dispose de 3 types de font :

- **font-sans** : Sans serif
- **font-serif** : Avec serif
- **font-mono** : Monospace

On peut utiliser d'autre font (comme des Google Fonts) avec :

- **font-[‘MaFont’]**

Classe

Toutes les propriétés et noms dans Tailwind correspondent à des classes, à mettre en attribut dans les balises concernées :

```
<h1 class="font-mono font-bold text-red-400 italic uppercase text-center">Hello</h1>
```

Cas pratique

Faire un titre en utilisant plusieurs types de font, de couleurs, et de style (italic, bold...).

Ajouter plusieurs paragraphes, en modifiant l'interligne, et en utilisant les mêmes couleurs, mais avec une nuance différente.

Le résultat doit être visuellement satisfaisant !

3. Box-Sizing

- 1 Définition et installation
- 2 Texte et couleurs
- 3 Box-sizing
- 4 Background
- 5 Mise en page (flexbox / grid)
- 6 Responsive design
- 7 Customisation

Largeur

La classe utilitaire pour la largeur est “w-{valeur}”.

Pour la hauteur, il suffit de remplacer le “w” par un “h”.

(1 rem = 4 px)

- **w-0** = width: 0;
- **w-px** = width: 1px;
- **w-auto** = width: auto;
- **w-screen** = width: 100vw;
- **w-fit** = width: fit-content;
- **w-0.5** = width: 0.125 rem; /* 2px */
- **w-1** = width: 0.25 rem; /* 4px */
- **w-2** = width: 0.5 rem; /* 8px */
- ...
- **w-72** = width: 20rem; /* 320px */
- **w-96** = width: 24 rem; /* 384px */

Largeur

Le pourcentage :

- **w-1/2** = width: 50%;
- **w-1/3** = width: 33.333%;
- **w-2/3** = width: 66.667%;
- **w-1/4** = width: 25%;
- **w-2/4** = width: 50%;
- **w-3/4** = width: 75%;
- ...
- **w-full** = width: 100%;

Minimum

Des classes permettent de définir la largeur minimale d'un élément :

- **min-w-0** = min-width: 0;
- **min-w-full** = min-width: 100%;
- **min-w-content** = min-width: fit-content;

Maximum

Pour le responsive, l'utilisation de classe pour la largeur maximale est parfaite :

- **max-w-0** = max-width: 0;
- **max-w-xs** = max-width: 20rem; /* 320px */
- **max-w-sm** = max-width: 24rem; /* 384px */
- **max-w-md** = max-width: 28rem; /* 448px */
- **max-w-lg** = max-width: 32rem; /* 512px */
- **max-w-xl** = max-width: 36rem; /* 576px */
- **max-w-2xl** = max-width: 42rem; /* 672px */
- ...
- **max-w-7xl** = max-width: 80rem; /* 1280px */
- **max-w-full** = max-width: 100%;
- **max-w-content** = max-width: fit-content;

Margin et padding

Devinez ?

Margin et padding

- “**m**” pour margin
- “**p**” pour padding

Excellent !

Margin et padding

Pour les côtés :

- **t : top**
- **b : bottom**
- **l : left**
- **r : right**
- **x : gauche + droite**
- **y : haut + bas**

Le côté est facultatif si l'on veut gérer les 4 directions simultanément.

Exemples

m-4 :

- margin-top : 1rem (ou 4px)
- margin-left : 1rem (ou 4px)
- margin-bottom : 1rem (ou 4px)
- margin-right : 1rem (ou 4px)

pt-6 :

- padding-top: 1.5rem (ou 6px)

mx-8 :

- margin-right : 2rem (ou 8px)
- margin-left : 2rem (ou 8px)

Si besoin, on peut utiliser des valeurs arbitraires :

pb-[5px] :

- padding-bottom : 5px;

Couleur

border-{couleur}, en utilisant les couleurs de Tailwind.

On peut utiliser plusieurs couleurs pour chacun des côtés d'une bordure, tout comme la taille.

Style

border-{valeur}:

- **solid**
- **dashed**
- **dotted**
- **double**
- **hidden**
- **none**

Cas pratique

Comment faire un espace pour un menu avec une liste de liens :

```
<nav>
  <ul>
    <li>Lien 1</li>
    <li>Lien 2</li>
    <li>Lien 3</li>
    <li>Lien 4</li>
  </ul>
</nav>
```

Comment faire un espace d'un paragraphe

```
<p class="max-w-[300px]">Un paragraphe sur au moins 2 lignes, car il a une largeur maximale définie à 300px</p>
```

Cas pratique

Comment faire une bordure :

- à gauche et à droite de 2px.
- tout autour de l'élément sauf en haut.
- de 10px d'épaisseur.
- de 4 couleurs différentes avec 4 tailles différents ?!

Comment faire une bordure avec 4 styles différents ?

Et le border-radius ?

Trouver les valeurs possibles pour arrondir tous les côtés, ou un/plusieurs côté(s) en particulier.

Vrai ou faux ?

L'attribut côté pour les arrondis peut-il prendre ces valeurs :

- **rounded-tl-md**
- **rounded-tr-2**
- **rounded-br-[10px]**
- **rounded-bs-lg**
- **rounded-full**

4. Background

- 1 Définition et installation
- 2 Texte et couleurs
- 3 Box-sizing
- 4 Background
- 5 Mise en page (flexbox / grid)
- 6 Responsive design
- 7 Customisation

Background

bg-[<value>] pour attribuer un arrière-plan.

- **bg-{color}** pour ajouter une couleur unie
- **bg-[#123456]** pour ajouter une couleur spécifique

Pour une image d'arrière-plan :

- **bg-[url(/img/libellule.jpg)]**

bg-gradient pour ajouter un dégradé

- **-to-t** : dégradé vertical vers le haut
- **-to-r** : dégradé vers la droite
- **-to-tr** : dégradé en diagonale vers le coin en haut à droite

from-{couleur} to-{couleur} pour indiquer les couleurs du dégradé :

- **bg-gradient-to-r from-orange-100 to-blue-500**

Background

Pour positionner l'arrière-plan (notamment pour les images) :

bg-{valeur} avec comme valeurs :

- **bottom**
- **center**
- **top**
- **left**
- **left-bottom**
- **left-top**
- **right**
- **right-bottom**
- **right-top**

Pour gérer l'adaptation de l'image :

- **bg-cover** : couvre la surface de l'élément, en rognant l'image
- **bg-contain** : affiche toute l'image, sans couvrir l'élément

Pour la répétition :

- **bg-no-repeat**
- **bg-repeat-x**
- **bg-repeat-y**

Cas pratique

Faites un beau fond en dégradé !

Essayez de mettre une image en arrière-plan, avec un image en local et une image sur internet.

5. Mise en page (flexbox / grid)

- 1 Définition et installation
- 2 Texte et couleurs
- 3 Box-sizing
- 4 Background
- 5 Mise en page (flexbox / grid)
- 6 Responsive design
- 7 Customisation

Conteneur

Comme dans Bootstrap, les éléments **container** de Tailwind CSS possèdent différents points de ruptures :

- Aucun = width: 100%;
- sm = max-width: 640px;
- md = max-width: 768px;
- lg = max-width: 1024px;
- xl = max-width: 1280px;
- 2xl = max-width: 1536px;

Certains frameworks centre automatiquement le contenu, pas Tailwind.

Il faut ajouter la propriété **mx-auto** au parent pour cela.

Colonnes

Il est possible de présenter le texte ou les images sous forme de colonnes.

Pour cela, nous utiliserons la classe **columns-{nombre}** ou encore **columns-{largeur}**

{nombre} :

- 1 à 12

{largeur} :

- **3xs**
- **2xs**
- **xs**
- **sm**
- **md**
- **lg**
- **xl**
- **2xl à 7xl**

Display

Avec Tailwind, les classes sont faciles à retenir, leur nom prend la valeur de la propriété *display* :

- **block**
- **inline-block**
- **inline**
- **flex**
- **grid**
- **!\\ hidden** = `display: none; !\\`

Visibility

Pour gérer la visibilité d'un élément, on utilise ces classes :

- **visible** = `visibility: visible;`
- **!\\ invisible** = `visibility: hidden; !\\`
- **collapse** = `visibility: collapse;`

FLEXBOX

Pour flex sur Tailwind, c'est très simple :

Flexbox

Les directions sont appelés par leur nom :

- **flex-row**
- **flex-col**
- **flex-row-reverse**
- **flex-col-reverse**

Wrap

L'évidence même :

- **flex-wrap**
- **flex nowrap**
- **flex-wrap-reverse**

Justify-content

Fastoche :

- **justify-start**
- **justify-end**
- **justify-center**
- **justify-between**
- **justify-around**
- **justify-evenly**

Align-items

Idem :

- **items-start**
- **items-end**
- **items-center**
- **items-stretch**

Order

Pour modifier l'ordre des éléments :

- **order-1 à 12**
- **order-first (-9999)**
- **order-last (9999)**
- **order-none (0)**

6. Responsive design

- 1 Définition et installation
- 2 Texte et couleurs
- 3 Box-sizing
- 4 Background
- 5 Mise en page (flexbox / grid)
- 6 Responsive design
- 7 Customisation

Breakpoint

Tailwind, tout comme Bootstrap, est un framework orienté mobile-first.

Concernant le responsive design, toutes les classes utilisées sans attribut de breakpoint s'appliqueront à partir du format mobile.

Pour chaque breakpoint, il faudra le spécifier de la façon suivante :

{breakpoint}:

Les breakpoints de Tailwind sont :

- **sm** (*40rem / 640px*)
- **md** (*48rem / 768px*)
- **lg** (*64rem / 1024px*)
- **xl** (*80rem / 1280px*)
- **2xl** (*96rem / 1536px*)

Exemples :

```
<h2 class="m-2 md:m-4 lg:m-6">  
| Un titre secondaire  
</h2>
```

En mobile, ce titre aura une margin de 2rem.

Au-delà de 768px, une marge de 4rem.

Au-delà de 1024px, une marge de 6rem.

Intervalle

En plus des breakpoints, Tailwind permet de créer des intervalles de point de rupture.

```
<h2 class="m-2 md:max-lg:m-4">  
  Un titre secondaire  
</h2>
```

Sur cet exemple, la modification de la marge pour le titre ne s'appliquera que pour les écrans dont la taille se situe entre 768px et 1024px.

Ces breakpoints et ces intervalles sont utilisables sur toutes les classes de Tailwind.

7. Customisation

- 1 Définition et installation
- 2 Texte et couleurs
- 3 Box-sizing
- 4 Background
- 5 Mise en page (flexbox / grid)
- 6 Responsive design
- 7 Customisation

Nouvelles classes

Nous pouvons créer des classes avec Tailwind, de façon à les réutiliser comme dans Bootstrap :

```
@theme {  
  --color-turquoise-400: #51e9e9;  
}
```

Les classes utilitaires comme **text-turquoise-400** ou **bg-turquoise-400** seront disponibles.

Variables

Nous pouvons modifier des variables dans Tailwind :

```
@theme {  
  --breakpoint-md: 22rem;  
}
```

Toutes les classes utilisant le breakpoint **md** se déclencheront désormais à partir de 22rem.

Composants

Nous pouvons créer des composants avec Tailwind, de façon à les réutiliser comme dans Bootstrap :

```
.btn-perso{  
|   @apply bg-blue-500 text-white px-4 py-2 text-lg rounded-lg w-fit;  
| }  
| }
```

Et créer des composants avec des variantes,
comme ceci.

@apply permet d'utiliser des classes utilitaires
déjà existantes.

```
.btn{  
|   @apply px-4 py-2 text-lg rounded-lg w-fit;  
| }  
  
.btn-blue {  
|   @apply bg-blue-500 text-white;  
| }  
| }
```

Composants

Les composants sont rassemblés dans un **@layer components**:

```
@layer components {
  .btn {
    |   @apply px-4 py-2 text-lg rounded-lg w-fit;
  }
  .btn-blue {
    |   @apply bg-blue-500 text-white;
  }
}
```