

# Support de cours - Tailwind CSS v4

Framework Utility-First - Version 4 (2024)

Formation DWWM - La Plateforme\_

## 1. Histoire et philosophie de Tailwind

### 1.1 Création et évolution

Tailwind CSS a été créé en 2017 par Adam Wathan et son équipe (Steve Schoger, Jonathan Reinink). C'est un framework CSS qui a révolutionné la façon de créer des interfaces web.

Timeline :

- 2017 : Naissance de Tailwind CSS
- 2019 : Version 1.0 stable
- 2020 : Version 2 avec dark mode et purge améliorée
- 2021 : Version 3 avec JIT (Just-In-Time) compiler
- 2024 : Version 4 - Révolution avec configuration CSS natif

💡 Tailwind v4 change TOUT : configuration CSS au lieu de JavaScript, 10x plus rapide !

### 1.2 Philosophie Utility-First

Tailwind utilise une approche "utility-first" : au lieu de composants prédéfinis (comme Bootstrap), tu composes ton design avec des classes atomiques.

Exemple comparatif :

```
<!-- BOOTSTRAP : Composant prédéfini -->
<button class="btn btn-primary">
  Mon bouton
</button>

<!-- TAILWIND : Composition de classes utilitaires -->
<button class="bg-blue-500 text-white px-4 py-2 rounded hover:bg-blue-600">
  Mon bouton
</button>
```

Avec Tailwind, tu construis ton design classe par classe, comme des briques de Lego.

### 1.3 Pourquoi Tailwind ?

- ✅ **Rapidité** : Pas besoin d'écrire de CSS custom, tout dans le HTML
- ✅ **Flexibilité** : Contrôle total sur le design, pas limité par des composants

- ✓ **Performance** : Fichier CSS de 10-50 KB après purge (vs 3+ MB avec CDN)
- ✓ **Responsive** : Classes responsive intégrées avec approche mobile-first
- ✓ **Cohérence** : Design system avec couleurs et espacements prédéfinis
- ✓ **Maintenance** : Pas de CSS à maintenir, tout est dans les classes

## 1.4 Tableau comparatif

Critère	Bootstrap	Materialize	Tailwind
Approche	Composants	Composants	Utility-first
Design	Neutre	Material	Totalement libre
Classes	Sémantiques	Sémantiques	Atomiques
Taille	~200 KB	~150 KB	10-50 KB (purgé)
Flexibilité	Moyenne	Moyenne	Maximale
Courbe	Facile	Facile	Moyenne

## 2. Installation Tailwind v4

**NEW** NOUVEAU v4 : Configuration CSS natif au lieu de JavaScript !

### 2.1 Changement majeur v4

Tailwind v4 change radicalement la façon de configurer le framework :

AVANT (v3) :

- Fichier tailwind.config.js (JavaScript)
- Fichier input.css avec @tailwind directives
- 2 fichiers à gérer

MAINTENANT (v4) :

- Un seul fichier app.css (CSS)
- Configuration avec @theme { }
- @import au lieu de @tailwind

✓ **Plus simple, plus rapide, CSS natif !**

### 2.2 Méthode 1 : CDN (pour apprendre)

Le CDN est le plus rapide pour découvrir Tailwind. Tu inclus juste un script dans le head.

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Tailwind CSS v4</title>
```

```

    <!-- CDN Tailwind CSS -->
    <script src="https://cdn.tailwindcss.com"></script>
</head>
<body>

    <h1 class="text-3xl font-bold text-blue-600">
        Hello Tailwind v4!
    </h1>

</body>
</html>

```

⚠ **ATTENTION** : Le CDN fait ~3 MB ! OK pour apprendre, JAMAIS en production.

## 2.3 Méthode 2 : CLI Tailwind (recommandé production)

Pour un vrai projet, on utilise la CLI de Tailwind avec configuration CSS.

Installation :

```

# Installer Tailwind v4
npm install tailwindcss@next @tailwindcss/cli@next

```

Créer le fichier app.css (TOUT EN UN) :

```

/* app.css */
@import "tailwindcss";

/* Configuration personnalisée */
@theme {
  --color-brand: #FF6B6B;
  --color-primary: #3B82F6;
}

/* Styles de base */
@layer base {
  h1 {
    @apply text-4xl font-bold;
  }
}

/* Composants */
@layer components {
  .btn {
    @apply px-4 py-2 rounded;
  }
}

```

Build :

```
# Development (watch)
npx @tailwindcss/cli -i app.css -o output.css --watch

# Production (minify + purge automatique)
npx @tailwindcss/cli -i app.css -o output.css --minify
```

HTML :

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="output.css">
</head>
<body>
  <h1 class="text-4xl">Tailwind v4</h1>
</body>
</html>
```

✅ Avec --minify, le fichier CSS final fait 10-50 KB au lieu de 3 MB !

## 2.4 Méthode 3 : Avec Vite/React

Pour les projets React, Vue, Next.js :

```
# Créer projet Vite
npm create vite@latest mon-projet

# Installer Tailwind v4
npm install tailwindcss@next @tailwindcss/vite@next

# Dans vite.config.js
import tailwindcss from "@tailwindcss/vite";

export default {
  plugins: [tailwindcss()],
}

# Dans src/index.css
@import "tailwindcss";
```

💡 Vite s'occupe automatiquement du build et de la purge !

## 3. Configuration v4 avec @theme

**NEW** NOUVEAU : Tout se configure dans le CSS avec @theme { }

### 3.1 Couleurs personnalisées

```

/* app.css */
@import "tailwindcss";

@theme {
  --color-brand: #FF6B6B;
  --color-primary: #3B82F6;
  --color-secondary: #8B5CF6;
}

<!-- HTML -->
<div class="bg-brand text-white">Ma couleur custom</div>
<button class="bg-primary">Bouton</button>

```

## 3.2 Espacements personnalisés

```

@theme {
  --spacing-18: 4.5rem;
  --spacing-72: 18rem;
}

<div class="p-18 m-72">Espacements custom</div>

```

## 3.3 Fonts personnalisées

```

@theme {
  --font-display: "Inter", sans-serif;
  --font-mono: "Fira Code", monospace;
}

<h1 class="font-display">Titre</h1>
<code class="font-mono">Code</code>

```

# 4. Classes utilitaires essentielles

## 4.1 Couleurs

Tailwind a 19 couleurs avec des nuances de 50 (très clair) à 950 (très foncé) :

Couleurs : red, orange, amber, yellow, lime, green, emerald, teal, cyan, sky, blue, indigo, violet, purple, fuchsia, pink, rose + grays (slate, gray, zinc, neutral, stone)

```

<!-- Background -->
<div class="bg-blue-500">Fond bleu standard</div>
<div class="bg-blue-50">Fond bleu très clair</div>
<div class="bg-blue-950">Fond bleu très foncé</div>

<!-- Texte -->

```

```
<p class="text-red-600">Texte rouge foncé</p>

<!-- Bordure -->
<div class="border-2 border-green-500">Bordure verte</div>
```

## 4.2 Spacing (E spacements)

Échelle : 0, 1, 2, 3, 4, 5, 6, 8, 10, 12, 16, 20, 24, 32, 40, 48, 56, 64...

1 unité = 0.25rem = 4px

Donc p-4 = padding de 16px

```
<!-- Padding -->
<div class="p-4">Padding 16px partout</div>
<div class="px-8 py-2">Horizontal 32px, vertical 8px</div>
<div class="pt-6">Padding top uniquement</div>

<!-- Margin -->
<div class="m-4">Margin 16px</div>
<div class="mx-auto">Centré horizontalement</div>
<div class="mt-8">Margin top 32px</div>
<div class="-mt-4">Margin négatif</div>
```

## 4.3 Typographie

```
<!-- Taille -->
<p class="text-xs">12px</p>
<p class="text-sm">14px</p>
<p class="text-base">16px (défaut)</p>
<p class="text-lg">18px</p>
<p class="text-xl">20px</p>
<p class="text-2xl">24px</p>
<p class="text-4xl">36px</p>

<!-- Poids -->
<p class="font-thin">100</p>
<p class="font-normal">400</p>
<p class="font-bold">700</p>
<p class="font-extrabold">800</p>

<!-- Alignement -->
<p class="text-left">Gauche</p>
<p class="text-center">Centré</p>
<p class="text-right">Droite</p>
```

## 4.4 Bordures et arrondis

```

<!-- Bordures -->
<div class="border">1px</div>
<div class="border-2">2px</div>
<div class="border-t-2">Top uniquement</div>
<div class="border-2 border-blue-500">Bleue</div>

<!-- Arrondis -->
<div class="rounded">4px</div>
<div class="rounded-lg">8px</div>
<div class="rounded-full">Cercle</div>

```

## 5. Layout : Flexbox et Grid

### 5.1 Flexbox

```

<!-- Activer flex -->
<div class="flex">
  <div>Item 1</div>
  <div>Item 2</div>
</div>

<!-- Direction -->
<div class="flex flex-row">Horizontal (défaut)</div>
<div class="flex flex-col">Vertical</div>

<!-- Justify (horizontal) -->
<div class="flex justify-center">Centré</div>
<div class="flex justify-between">Espacé</div>

<!-- Align (vertical) -->
<div class="flex items-center">Centré vertical</div>

<!-- Centrage parfait -->
<div class="flex justify-center items-center h-screen">
  Parfaitement centré
</div>

```

### 5.2 Grid

```

<!-- Grid 3 colonnes -->
<div class="grid grid-cols-3 gap-4">
  <div>Colonne 1</div>
  <div>Colonne 2</div>
  <div>Colonne 3</div>
</div>

<!-- Grid responsive -->

```

```

<div class="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-4 gap-6">
  <!-- 1 col mobile, 2 tablette, 4 desktop -->
</div>

<!-- Span (occuper plusieurs colonnes) -->
<div class="grid grid-cols-3">
  <div class="col-span-2">2 colonnes</div>
  <div>1 colonne</div>
</div>

```

## 6. Responsive Design

### 6.1 Breakpoints

Tailwind utilise une approche mobile-first :

Préfixe	Taille	Description
(rien)	0px	Mobile (défaut)
sm:	640px	Petit écran
md:	768px	Tablette
lg:	1024px	Desktop
xl:	1280px	Grand écran
2xl:	1536px	Très grand

💡 Mobile-first : sans préfixe = mobile, puis md:, lg: pour plus grand

```

<!-- Taille texte responsive -->
<h1 class="text-2xl md:text-4xl lg:text-6xl">
  Mobile: 24px, Tablette: 36px, Desktop: 60px
</h1>

<!-- Largeur responsive -->
<div class="w-full md:w-1/2 lg:w-1/3">
  Mobile: 100%, Tablette: 50%, Desktop: 33%
</div>

<!-- Grid responsive -->
<div class="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3">
  1 col mobile, 2 tablette, 3 desktop
</div>

```

### 6.2 Cacher/Afficher selon écran

```

<!-- Visible mobile uniquement -->
<div class="block md:hidden">
  Menu hamburger mobile

```



```

</div>

<!-- Visible desktop uniquement -->
<div class="hidden md:block">
  Menu complet desktop
</div>

```

## 7. États et interactions

```

<!-- Hover -->
<button class="bg-blue-500 hover:bg-blue-700">
  Survole-moi
</button>

<!-- Focus -->
<input class="border-2 focus:border-blue-500 focus:ring-2">

<!-- Transitions -->
<button class="bg-blue-500 hover:bg-blue-700
  transition duration-300">
  Transition douce
</button>

<!-- Transform -->
<div class="hover:scale-110 transition">
  Zoom au survol
</div>

```

## 8. Composants courants

### 8.1 Bouton

```

<button class="bg-blue-500 hover:bg-blue-600 text-white
  px-6 py-3 rounded-lg font-semibold
  transition shadow-lg">
  Mon bouton
</button>

```

### 8.2 Card

```

<div class="bg-white rounded-xl shadow-lg overflow-hidden">
  
  <div class="p-6">
    <h3 class="text-2xl font-bold mb-2">Titre</h3>
    <p class="text-gray-600 mb-4">Description...</p>
  </div>
</div>

```

```

        <button class="bg-blue-500 text-white px-4 py-2 rounded">
            Action
        </button>
    </div>
</div>

```

## 8.3 Navbar

```

<nav class="bg-blue-600 text-white p-4">
    <div class="container mx-auto flex justify-between items-center">
        <div class="text-2xl font-bold">Logo</div>
        <ul class="flex gap-6">
            <li><a href="#">Accueil</a></li>
            <li><a href="#">Services</a></li>
            <li><a href="#">Contact</a></li>
        </ul>
    </div>
</nav>

```

## 9. Bonnes pratiques

- ✓ CDN uniquement pour apprendre et tester
- ✓ CLI + --minify pour production (10-50 KB)
- ✓ Configurer @theme pour couleurs/fonts custom
- ✓ Mobile-first : sans préfixe puis md: lg:
- ✓ Utiliser @layer components pour composants réutilisables
- ✓ Transitions pour interactions fluides
- ⚠ JAMAIS CDN en production (3+ MB)
- ⚠ Ne pas mélanger avec Bootstrap/Materialize
- ⚠ Ne pas générer classes dynamiquement (concat)

## 10. Résumé v4

Points clés Tailwind v4 :

1. Configuration CSS : @theme { } au lieu de tailwind.config.js
2. Import simple : @import "tailwindcss"
3. Un seul fichier : app.css contient tout
4. 10x plus rapide : Engine Oxide (Rust)
5. CSS natif : Plus besoin de JavaScript pour configurer

- 6. Utility-first : Composer avec classes atomiques
- 7. Performance : 10-50 KB après purge
- 8. Responsive : Mobile-first sm: md: lg:
- 9. Flexbox/Grid : Layout moderne
- 10. Ressources : Flowbite 450+ composants gratuits

✅ **v4 = Plus simple, plus rapide, plus moderne !**