

Guide Debugging - Tailwind CSS v4

Résoudre les erreurs courantes - Version 4

Formation DWWM - La Plateforme

 Nouveau v4 : Erreurs différentes avec la nouvelle configuration CSS !

Erreur 1 : @import ne fonctionne pas

Symptôme

Message d'erreur : "@import 'tailwindcss' not found" ou les styles ne s'appliquent pas

Cause

Version incorrecte de Tailwind installée (v3 au lieu de v4)

Solution

Vérifier la version installée :

```
npm list tailwindcss
```

Si < 4.0, installer v4 :

```
npm uninstall tailwindcss
npm install tailwindcss@next @tailwindcss/cli@next
```

Fichier app.css correct :

```
/* app.css */
@import "tailwindcss"; /* v4 syntax */

@theme {
  /* Config ici */
}
```

 Vérifier que tu utilises bien v4, pas v3 !

Erreur 2 : @theme ne fonctionne pas

Symptôme

Classes custom définies dans @theme ne s'appliquent pas

Cause

Syntaxe incorrecte dans @theme ou mauvaise utilisation

Solution

Syntaxe correcte :

```
@theme {  
    --color-brand: #FF6B6B;      /* ✅ CORRECT */  
    --spacing-18: 4.5rem;  
    --font-display: "Inter", sans-serif;  
}  
  
/* Utilisation */  
<div class="bg-brand">Correct</div> /* Pas bg-color-brand */
```

Erreurs courantes :

```
/* ❌ INCORRECT */  
@theme {  
    color-brand: #FF6B6B;      /* Manque -- */  
    --color_brand: #FF6B6B;    /* Underscore au lieu de - */  
}
```

⚠ Toujours utiliser -- avant le nom et - pour séparer les mots

Erreur 3 : Classes ne s'appliquent pas (v4)

Symptôme

Classes Tailwind ajoutées mais aucun effet visuel

Cause

CSS pas généré ou fichier output.css pas inclus

Solution 1 : Vérifier le build

```
# Vérifier que le build tourne  
npx @tailwindcss/cli -i app.css -o output.css --watch  
  
# Vérifier que output.css contient du code  
# Le fichier doit faire au moins quelques KB
```

Solution 2 : Vérifier le link HTML

```
<!DOCTYPE html>
```

```
<html>
<head>
    <!-- ✅ Vérifier ce link -->
    <link rel="stylesheet" href="output.css">
</head>
<body>
    <h1 class="text-3xl">Test</h1>
</body>
</html>
```

✅ Le fichier output.css doit exister et être lié dans le HTML

Erreur 4 : Migration v3 → v4 échouée

Symptôme

Erreurs après tentative de migration depuis v3

Cause

Ancienne syntaxe v3 mélangée avec v4

Solution : Migration complète

Étape 1 : Remplacer @tailwind par @import

```
/* ❌ ANCIEN (v3) */
@tailwind base;
@tailwind components;
@tailwind utilities;

/* ✅ NOUVEAU (v4) */
@import "tailwindcss";
```

Étape 2 : Convertir tailwind.config.js en @theme

```
/* ❌ ANCIEN (v3) */
// tailwind.config.js
module.exports = {
  theme: {
    extend: {
      colors: { brand: "#FF6B6B" }
    }
  }
}

/* ✅ NOUVEAU (v4) */
/* app.css */
@theme {
```

```
--color-brand: #FF6B6B;  
}
```

Étape 3 : Supprimer tailwind.config.js (optionnel)

💡 tailwind.config.js n'est plus nécessaire en v4, tout est dans app.css

Erreur 5 : Classes custom ne fonctionnent pas

Symptôme

Classes inventées (bg-blue-550, p-7) ne marchent pas

Cause

Tailwind a une échelle prédéfinie, on ne peut pas inventer de valeurs

Solution 1 : Utiliser valeurs existantes

```
/* ❌ INCORRECT - valeurs inventées */  
bg-blue-550  p-7  text-3.5xl  
  
/* ✅ CORRECT - valeurs existantes */  
bg-blue-500  p-6  text-4xl
```

Solution 2 : Créer dans @theme

```
@theme {  
  --color-blue-550: #4A90E2;  
  --spacing-7: 1.75rem;  
}  
  
<div class="bg-blue-550 p-7">Custom values</div>
```

Solution 3 : Valeurs arbitraires

```
<!-- Syntaxe [valeur] pour valeurs arbitraires -->  
<div class="bg-[#123456] p-[17px] w-[347px]">  
  Valeurs arbitraires  
</div>
```

Erreur 6 : Responsive ne fonctionne pas

Symptôme

Classes responsive md: lg: ne s'appliquent pas aux bons breakpoints

Cause

Mauvaise compréhension du mobile-first

Solution : Comprendre mobile-first

```
/* Mobile-first = classes sans préfixe pour mobile */

<!-- ✅ CORRECT -->
<div class="text-sm md:text-base lg:text-xl">
    Mobile: text-sm (14px)
    Tablette ≥768px: text-base (16px)
    Desktop ≥1024px: text-xl (20px)
</div>

<!-- ❌ INCORRECT - Penser desktop-first -->
<div class="text-xl md:text-base sm:text-sm">
    Ordre inversé, ne marche pas comme prévu
</div>
```

💡 Toujours commencer par mobile (sans préfixe), puis ajouter md: lg: pour plus grand

Erreur 7 : Fichier CSS trop lourd

Symptôme

output.css fait plusieurs MB, temps de chargement énorme

Cause

Pas de purge automatique ou mauvais build

Solution

Build production avec --minify :

```
# ❌ MAUVAIS - Development
npx @tailwindcss/cli -i app.css -o output.css
# Résultat : ~3 MB

# ✅ BON - Production avec purge
npx @tailwindcss/cli -i app.css -o output.css --minify
# Résultat : 10-50 KB !
```

✓ En production, le CSS doit faire 10-50 KB, pas 3 MB

Erreur 8 : @layer ne fonctionne pas

Symptôme

Classes définies dans @layer ne s'appliquent pas

Cause

@layer doit être après @import

Solution : Ordre correct

```
/* ✅ ORDRE CORRECT */
/* app.css */

/* 1. Import d'abord */
@import "tailwindcss";

/* 2. Theme ensuite */
@theme {
  --color-brand: #FF6B6B;
}

/* 3. Layers après */
@layer base {
  h1 { @apply text-4xl font-bold; }
}

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

@layer utilities {
  .content-auto { content-visibility: auto; }
}
```

Erreur 9 : Hover/Focus ne fonctionnent pas

Symptôme

hover: et focus: ne s'appliquent pas

Cause

États hover:/focus: ne fonctionnent que sur certains éléments

Solution

```
<!-- ✓ CORRECT - Éléments interactifs -->
<button class="hover:bg-blue-700">Bouton</button>
<a href="#" class="hover:underline">Lien</a>
<input class="focus:border-blue-500">

<!-- ⚡ Pour div, ajouter cursor-pointer -->
<div class="hover:bg-gray-100 cursor-pointer">
    Div cliquable
</div>
```

Erreur 10 : Breakpoints custom ne marchent pas

Symptôme

Breakpoints personnalisés définis dans @theme ne fonctionnent pas

Cause

Syntaxe incorrecte ou mauvaise utilisation

Solution

```
/* app.css */
@theme {
    --breakpoint-tablet: 768px;
    --breakpoint-desktop: 1440px;
}

<!-- HTML -->
<div class="text-sm tablet:text-lg desktop:text-xl">
    Utilise les breakpoints custom
</div>
```

✓ Les breakpoints custom définis dans @theme sont utilisables directement

Checklist debugging v4

- ✓ Version Tailwind v4 installée (@next)
- ✓ @import "tailwindcss" dans app.css
- ✓ @theme {} avec syntaxe -- correcte
- ✓ Build CLI en cours (--watch)
- ✓ output.css lié dans HTML
- ✓ Mobile-first : sans préfixe puis md: lg:

- Build production avec --minify (10-50 KB)
- @layer après @import et @theme
- Pas de classes dynamiques (concat)
- Console F12 pour voir erreurs

Outils debugging

DevTools (F12)

- Inspect Element → Vérifier classes appliquées
- Console → Voir erreurs CSS/JS
- Device Toolbar → Tester responsive

Tailwind Play

- <https://play.tailwindcss.com/>
- Tester code rapidement en ligne
- Voir résultat immédiat

VS Code Extensions

- Tailwind CSS IntelliSense
- Autocomplétion classes
- Aperçu couleurs

Résumé v4

Différences debug v3 vs v4 :

v3 :

- Vérifier tailwind.config.js
- Vérifier @tailwind directives
- Vérifier content: [...] dans config

v4 :

- Vérifier @import "tailwindcss"
- Vérifier @theme { } syntaxe
- Vérifier ordre : @import puis @theme puis @layer
- Vérifier version v4 (@next)

v4 simplifie le debug : 1 fichier au lieu de 2 !

⚠ Si erreur, vérifier d'abord la version Tailwind installée

Formation DWWM - La Plateforme_ - 2025