

Créer une animation personnalisée

Temps de lecture : 2 minutes



Créer des animations et transitions

Pour créer des animations et des transitions personnalisées avec [Tailwind CSS](#), vous devez manipuler le fichier de configuration [tailwind.config.js](#).

Vous pouvez faire comme dans la vidéo, mais nous voyons ici une méthode alternative que nous reverrons encore de manière détaillée plus tard.

Ce fichier vous permet de personnaliser les thèmes, y compris les animations et les [keyframes](#).

Configurer une animation personnalisée

Dans votre fichier [tailwind.config.js](#), vous pouvez étendre la configuration par défaut pour inclure vos propres animations.

Utilisez la clé [extend](#) dans l'objet [theme](#) pour ajouter des animations personnalisées sans écraser les valeurs par défaut.

Par exemple :

```
module.exports = {
  theme: {
    extend: {
      animation: {
        'spin-slow': 'spin 3s linear infinite',
      }
    }
  }
}
```

Dans cet exemple, une animation nommée [spin-slow](#) est ajoutée. Cette animation utilise les [@keyframes](#) nommés [spin](#) (préfourmis par [Tailwind CSS](#)) et se déroule sur 3 secondes avec une fonction de timing linéaire, se répétant indéfiniment.

Ajouter des [keyframes](#) personnalisés

Pour des animations plus spécifiques, vous pouvez définir vos propres `@keyframes` dans le fichier de configuration.

Par exemple :

```
module.exports = {
  theme: {
    extend: {
      keyframes: {
        wiggle: {
          '0%, 100%': { transform: 'rotate(-3deg)' },
          '50%': { transform: 'rotate(3deg)' },
        }
      }
    }
  }
}
```

Référencer les keyframes personnalisés dans une animation

Après avoir défini vos `@keyframes`, vous pouvez les référencer par nom dans la section `animation` de votre configuration pour créer une animation personnalisée.

```
module.exports = {
  theme: {
    extend: {
      animation: {
        wiggle: 'wiggle 1s ease-in-out infinite',
      }
    }
  }
}
```

Cette configuration crée une animation `wiggle` qui applique les `@keyframes wiggle` pendant 1 seconde avec une fonction de timing `ease-in-out`, se répétant indéfiniment.

Vous pouvez ensuite utiliser votre animation personnalisée de cette manière :

```
<div class="animate-wiggle">...</div>
```

Code de la vidéo

src/style.css

```
@tailwind base;
@tailwind components;
@tailwind utilities;

@layer components {
  .animate-spin-slow {
    animation: spin 5s linear infinite;
  }

  .animate-fade-in-from-top {
    animation: fadeInFromTop 1s ease-out;
  }
}

@keyframes fadeInFromTop {
  from {
    transform: translateY(-10px);
    opacity: 0;
  }
  to {
    transform: translateY(0);
    opacity: 1;
  }
}
```

tailwind.config.js

```
/** @type {import('tailwindcss').Config} */
module.exports = {
  content: ["./src/**/*.html", "js"],
  safelist: ["animate-spin"],
  theme: {
    extend: {},
  },
}
```

```
    plugins: [],  
  };
```

src/index.html

```
<!doctype html>  
<html>  
  <head>  
    <meta charset="UTF-8" />  
    <meta name="viewport" content="width=device-width, initial-scale=1.  
0" />  
    <link href="./output.css" rel="stylesheet" />  
    <script  
      src="https://kit.fontawesome.com/33a775bcd5.js"  
      crossorigin="anonymous"  
    ></script>  
  </head>  
  <body>  
    <div class="flex h-screen items-center justify-center gap-x-3">  
      <button  
        class="animate-fade-in-from-top rounded-full bg-blue-500 px-6 py  
-4 text-xl font-medium text-white"  
      >  
        <i class="fa-solid fa-spinner animate-spin-slow"></i>  
        Cliquez !  
      </button>  
    </div>  
  </body>  
</html>
```