

Tailwind CSS

Зміст уроку

1. [Що таке Tailwind CSS?](#)
2. [Переваги Tailwind CSS](#)
3. [Ключові концепції Tailwind CSS](#)
4. [Концепції Tailwind CSS на прикладах, простими словами](#)
5. [Огляд та пояснення документації Tailwind CSS](#)
6. [Customization](#)
7. [Base Styles. Preflight](#)
8. [Layout](#)
9. [Flexbox & Grid](#)
10. [Spacing](#)
11. [Sizing](#)
12. [Typography](#)
14. [Backgrounds](#)
15. [Borders](#)
16. [Effects](#)
17. [Filters](#)
18. [Tables](#)
19. [Transitions & Animation](#)
20. [Transforms](#)
21. [Interactivity](#)
22. [SVG](#)
23. [Кастомізація Tailwind CSS](#)
24. [Підсумки](#)
25. [Принципова відмінність Tailwind CSS від Bootstrap](#)

Що таке Tailwind CSS?

Tailwind CSS – це утилітарний фреймворк для CSS, який дозволяє нам швидко створювати індивідуальні дизайни без потреби писати багато кастомного CSS коду.

Замість того, щоб зосереджуватися на стилізації через велику кількість CSS класів, ми використовуємо набір утилітарних класів, які можна застосовувати прямо у HTML.

Це робить процес розробки значно швидшим і гнучкішим, оскільки ми маємо можливість контролювати вигляд нашого вебсайту прямо з розмітки, не переходячи до окремого CSS файлу.

Переваги Tailwind CSS

Швидкість розробки: Завдяки утилітарному підходу до стилізації, ми можемо швидко застосовувати дизайн без потреби в написанні великої кількості CSS коду.

Відповідність дизайн-системі: Tailwind спрощує дотримання єдиної дизайн-системи через налаштування теми, що забезпечує консистентність інтерфейсу.

Гнучкість: Ми маємо повний контроль над стилізацією елементів, що дозволяє легко адаптувати дизайн під будь-які потреби.

Мінімізація зайвого коду: Tailwind включає інструменти для очищення невикористаного CSS, що сприяє зменшенню розміру кінцевого файлу стилів.

Легкість навчання та використання: Завдяки інтуїтивно зрозумілій системі класів, Tailwind забезпечує легкий вхід для новачків у веб-розробці.

Спільнота та ресурси: Велика та активна спільнота надає численні ресурси, компоненти, плагіни та інструменти для розширення можливостей Tailwind.

Ключові концепції Tailwind CSS

Utility-First Fundamentals

Ми використовуємо утилітарний підхід як основу, що дозволяє нам швидко створювати інтерфейси за допомогою готових класів. Це спрощує процес розробки, роблячи код більш читабельним та легким для підтримки.

Hover, Focus, and Other States

Ми застосовуємо стилі для різних станів елементів, таких як `hover` чи `focus`, безпосередньо через утилітарні класи. Це дозволяє нам легко додавати інтерактивність та поліпшувати користувацький досвід.

Responsive Design

Ми створюємо адаптивні веб-дизайни, використовуючи класи, що автоматично адаптують елементи до різних розмірів екрану. Це забезпечує оптимальний перегляд на будь-якому пристрої.

Dark Mode

Ми підтримуємо темну тему з можливістю легкого перемикання між світлим і темним режимами, щоб відповідати перевагам користувачів і зменшити втому очей при роботі в темний час доби.

Reusing Styles

Ми заохочуємо повторне використання стилів через компонентизацію, дозволяючи зберегти консистентність дизайну і зменшити кількість дублювання коду.

Adding Custom Styles

Ми дозволяємо додавання власних стилів для деталізації або розширення стандартних утиліт, що надає більшу гнучкість у персоналізації дизайну.

Functions & Directives

Ми використовуємо функції та директиви для створення складних стилів і логіки, що дозволяє нам ефективно управляти стилями на рівні препроцесора.

Концепції Tailwind CSS на прикладах, простими словами

Концепція використання утилітних класів замість традиційного CSS

Tailwind відходить від традиційного підходу до написання CSS з використанням селекторів, таких як **.class**, **#id** тощо. Замість цього використовується компонентний підхід за допомогою утилітних класів.

Утилітний клас - це невеликий CSS клас, який відповідає за одну конкретну властивість: шрифт, колір, відступи тощо.

Класи для всіх властивостей: розміри, колір, позиціонування, flexbox тощо

Tailwind містить сотні готових утилітних класів для таких властивостей:

- Розміри: padding, margins, width, height тощо (**.p-3**, **.mx-auto**)
- Кольори: тексту, фону (**.text-purple-500**, **.bg-gray-900**)
- Flexbox: **.flex**, **.justify-center**, **.items-start**
- Позиціонування: **.relative**, **.top-0**, **.right-10**
- та багато інших

Це значно спрощує процес стилізації.

Швидка стилізація без написання великої кількості власних CSS стилів

Приклад стилізації кнопки:

```
<button class="bg-blue-500 text-white font-bold py-2 px-4 rounded" >
```

Кнопка

```
</button>
```

Весь CSS реалізовано через комбінацію готових утилітних класів, зменшуючи необхідність написання власних стилів та пришвидшуючи розробку. Зміна стилів легко досягається шляхом заміни чи додавання класів.

Огляд та пояснення документації Tailwind CSS

Customization

В рамках налаштування Tailwind CSS ми маємо можливість кастомізувати багато аспектів фреймворку, щоб вони відповідали конкретним потребам наших проєктів. Ось як ми можемо використовувати кожен з цих аспектів:

- **Configuration**
- **Content**
- **Theme**
- **Screens**
- **Colors**
- **Spacing**
- **Plugins**
- **Presets**

Customization

→ Colors

Neutral											
	50 #fafafa	100 #f5f5f5	200 #e5e5e5	300 #d4d4d4	400 #a3a3a3	500 #737373	600 #525252	700 #404040	800 #262626	900 #171717	950 #0a0a0a
Stone											
	50 #fafaf9	100 #f5f5f4	200 #e7e5e4	300 #d6d3d1	400 #a8a29e	500 #78716c	600 #57534e	700 #44403c	800 #292524	900 #1c1917	950 #0c0a09
Red											
	50 #fef2f2	100 #fee2e2	200 #fecaca	300 #fca5a5	400 #f87171	500 #ef4444	600 #dc2626	700 #b91c1c	800 #991b1b	900 #7f1d1d	950 #450a0a
Orange											
	50 #fff7ed	100 #ffedd5	200 #fed7aa	300 #fdb74	400 #fb923c	500 #f97316	600 #ea580c	700 #c2410c	800 #9a3412	900 #7c2d12	950 #431407
Amber											
	50 #fffbeb	100 #fef3c7	200 #fde68a	300 #fcd34d	400 #fbbf24	500 #f59e0b	600 #d97706	700 #b45309	800 #92400e	900 #78350f	950 #451a03
Yellow											
	50 #fefce8	100 #fef9c3	200 #fef08a	300 #fde047	400 #facc15	500 #eab308	600 #ca8a04	700 #a16207	800 #854d0e	900 #713f12	950 #422006
Lime											
	50 #f7fee7	100 #ecfccb	200 #d9f99d	300 #bef264	400 #a3e635	500 #84cc16	600 #65a30d	700 #4d7c0f	800 #3f6212	900 #365314	950 #1a2e05

Base Styles. Preflight

У нашій роботі з Tailwind CSS, ми використовуємо базові стилі, відомі як **Preflight**. **Preflight** — це набір глобальних стилів на основі **Normalize.css**, які автоматично застосовуються до проєкту, щоб забезпечити більш консистентне відображення елементів у різних браузерах.

Він скидає стандартні маржини, встановлює корисні вихідні налаштування для деяких HTML елементів і вводить загальні стилі, які служать солідною базою для подальшої стилізації.

Layout

В нашій розробці веб-інтерфейсів за допомогою Tailwind CSS, ми використовуємо різні утилітарні класи для створення та контролю розкладки (layout) елементів на сторінці. Кожен із зазначених аспектів грає важливу роль у структуруванні та представленні контенту:

- **Aspect Ratio**
- **Container**
- **Columns**
- **Break After, Break Before, Break Inside**
- **Box Decoration Break**
- **Box Sizing**
- **Display**
- **Floats, Clear**
- **Isolation**
- **Object Fit, Object Position**
- **Overflow, Overscroll Behavior**
- **Position, Top / Right / Bottom / Left**
- **Visibility**
- **Z-Index**

Flexbox & Grid

В нашій роботі з веб-дизайном ми активно використовуємо Flexbox і Grid, які є потужними інструментами CSS для створення гнучких та комплексних макетів веб-сторінок. Ось основні поняття та застосування кожного з них у нашій розробці:

- **Flexbox**
- **Grid**
- **Спільні властивості для Flexbox і Grid**

Flexbox

- **Flex Basis:** Встановлюємо базовий розмір елементів у флекс-контейнері, перш ніж додаткове місце розподіляється.
- **Flex Direction:** Визначаємо напрямок розташування елементів у флекс-контейнері, горизонтально чи вертикально.
- **Flex Wrap:** Контролюємо, чи повинні елементи обгортатися на новий рядок або колонку при недостатньому просторі.
- **Flex:** Комбінований властивість для flex-grow, flex-shrink, і flex-basis, дозволяючи елементам розтягуватися або стискатися для заповнення доступного простору.
- **Flex Grow:** Встановлюємо здатність елемента розширюватися, щоб заповнити невикористаний простір.
- **Flex Shrink:** Визначаємо, як елементи будуть зменшуватися, щоб вміститися в контейнер.
- **Order:** Змінюємо порядок розташування елементів у флекс-контейнері без зміни HTML.

Grid

- **Grid Template Columns / Rows:** Визначаємо розміри колонок та рядків у грід-контейнері, створюючи складні макети.
- **Grid Column Start / End і Grid Row Start / End:** Встановлюємо, де елементи починають і закінчуються у макеті ґріда, дозволяючи їм займати декілька колонок або рядків.
- **Grid Auto Flow:** Керуємо автоматичним розміщенням елементів у грід-контейнері, вирішуючи, як заповнювати порожні клітинки.
- **Grid Auto Columns / Rows:** Встановлюємо розмір колонок або рядків, які створюються автоматично, коли є додаткові елементи.
- **Gap:** Визначаємо простір між елементами ґріда або флекс-елементами, забезпечуючи рівномірні відступи.

Спільні властивості для Flexbox і Grid

- **Justify Content / Items / Self:** Контролюємо вирівнювання елементів уздовж головної осі (для Flexbox) або у макеті ґріда, включаючи індивідуальне вирівнювання.
- **Align Content / Items / Self:** Управляємо вирівнюванням елементів уздовж поперечної осі (для Flexbox) або у ґрид-контейнері, забезпечуючи вертикальне вирівнювання.
- **Place Content / Items / Self:** Комбінує вирівнювання як по головній, так і по поперечній осі, спрощуючи CSS синтаксис.

Spacing

В нашій роботі з Tailwind CSS ми активно використовуємо утилітарні класи для управління відступами, що є ключовим аспектом при створенні візуально привабливих та читабельних веб-інтерфейсів. Ось як ми застосовуємо ці класи:

- **Padding**
- **Margin**
- **Space Between**

Spacing

→ Padding

Class	Properties
<code>p-0</code>	<code>padding: 0px;</code>
<code>px-0</code>	<code>padding-left: 0px;</code> <code>padding-right: 0px;</code>
<code>py-0</code>	<code>padding-top: 0px;</code> <code>padding-bottom: 0px;</code>
<code>ps-0</code>	<code>padding-inline-start: 0px;</code>
<code>pe-0</code>	<code>padding-inline-end: 0px;</code>
<code>pt-0</code>	<code>padding-top: 0px;</code>
<code>pr-0</code>	<code>padding-right: 0px;</code>

Spacing

→ Margin

Class

Properties

`m-0`

`margin: 0px;`

`mx-0`

`margin-left: 0px;`

`margin-right: 0px;`

`my-0`

`margin-top: 0px;`

`margin-bottom: 0px;`

`ms-0`

`margin-inline-start: 0px;`

`me-0`

`margin-inline-end: 0px;`

`mt-0`

`margin-top: 0px;`

`mr-0`

`margin-right: 0px;`

Spacing

- Border Width
- Border Width
- Border Style

Class	Properties
<code>border-0</code>	<code>border-width: 0px;</code>
<code>border-2</code>	<code>border-width: 2px;</code>
<code>border-4</code>	<code>border-width: 4px;</code>
<code>border-8</code>	<code>border-width: 8px;</code>
<code>border</code>	<code>border-width: 1px;</code>
<code>border-x-0</code>	<code>border-left-width: 0px;</code> <code>border-right-width: 0px;</code>
<code>border-x-2</code>	<code>border-left-width: 2px;</code> <code>border-right-width: 2px;</code>

Sizing

В нашій розробці інтерфейсів ми часто зіштовхуємося з необхідністю точно контролювати розміри елементів, для чого використовуємо різноманітні утилітарні класи Tailwind CSS, що дозволяють нам встановлювати ширину, висоту та інші параметри розміру. Ось основні категорії, з якими ми працюємо:

- **Width і Height**
- **Min-Width і Min-Height**
- **Max-Width і Max-Height**
- **Size**

Typography

У нашій роботі з дизайном та версткою веб-сторінок, ми приділяємо особливу увагу типографіці, використовуючи різноманітні утилітарні класи Tailwind CSS для налаштування тексту. Ось основні аспекти, на які ми звертаємо увагу:

- **Font Family**
- **Font Size**
- **Font Smoothing**
- **Font Style, Weight, Variant Numeric**
- **Letter Spacing, Line Height**
- **Line Clamp**
- **List Style**
- **Text Align, Color**
- **Text Decoration**
- **Text Overflow, Wrap, Indent**
- **Vertical Align, Whitespace, Word Break**
- **Hyphens, Content**

Backgrounds

В нашій розробці веб-інтерфейсів, ми ретельно працюємо з фонами елементів, використовуючи різноманітні утилітарні класи Tailwind CSS для створення візуально привабливих дизайнів. Кожен з аспектів фонів відіграє ключову роль у визначенні загального візуального сприйняття сторінки:

- **Background Attachment**
- **Background Clip**
- **Background Color**
- **Background Origin**
- **Background Position**
- **Background Repeat**
- **Background Size**
- **Background Image**
- **Gradient Color Stops**

Borders

В нашій роботі з дизайном веб-інтерфейсів, ми активно використовуємо утилітарні класи для налаштування меж та обрамлень елементів, що дозволяє нам створювати визначені, стилістично привабливі та функціональні дизайни. Ось як ми застосовуємо ці класи:

- **Border Radius**
- **Border Width**
- **Border Color**
- **Border Style**
- **Divide Width, Color, Style**
- **Outline Width, Color, Style, Offset**
- **Ring Width, Color, Offset Width, Offset Color**

Effects

У нашій розробці веб-інтерфейсів, ми застосовуємо різноманітні ефекти для збагачення візуального досвіду користувачів. Ось детальне пояснення, як ми використовуємо кожен з них:

- **Box Shadow**
- **Box Shadow Color**
- **Opacity**
- **Mix Blend Mode**
- **Background Blend Mode**

Filters

У нашій роботі над веб-дизайном, ми використовуємо фільтри для додавання візуальних ефектів до елементів і фонів, що дозволяє нам створювати унікальні атмосфери та покращувати візуальну привабливість наших проєктів. Ось як ми застосовуємо різні фільтри:

- **Blur**
- **Brightness**
- **Contrast**
- **Drop Shadow**
- **Grayscale**
- **Hue Rotate**
- **Invert**
- **Saturate**
- **Sepia**
- **Backdrop Filters**

Tables

При створенні веб-дизайнів, ми часто працюємо з таблицями для представлення даних чи інформації. Ось як ми використовуємо ключові стилістичні особливості таблиць у нашій роботі:

- **Border Collapse**
- **Border Spacing**
- **Table Layout**
- **Caption Side**

Transitions & Animation

У нашій роботі з дизайном веб-інтерфейсів, ми використовуємо переходи та анімації для додання динаміки та покращення користувацького досвіду. За допомогою Tailwind CSS, ми легко інтегруємо ці ефекти у наші проекти:

- **Transition Property**
- **Transition Duration**
- **Transition Timing Function**
- **Transition Delay**
- **Animation**

Transforms

У нашій розробці веб-інтерфейсів, ми використовуємо трансформації для додавання інтерактивності та візуальних ефектів до елементів. Tailwind CSS надає нам потужні утиліти для цього, дозволяючи легко застосовувати наступні трансформації:

- **Scale**
- **Rotate**
- **Translate**
- **Skew**
- **Transform Origin**

Interactivity

В нашій розробці веб-інтерфейсів, ми приділяємо велику увагу інтерактивності, щоб забезпечити користувачам зручне та інтуїтивно зрозуміле взаємодію з елементами веб-сайту. Tailwind CSS надає нам різноманітні утилітарні класи для покращення інтерактивності, включаючи:

- **Accent Color**
- **Appearance**
- **Cursor**
- **Caret Color**
- **Pointer Events**
- **Resize**
- **Scroll Behavior**
- **Scroll Margin, Padding, Snap Align, Snap Stop, Snap Type**
- **Touch Action**
- **User Select**
- **Will Change**

SVG

У нашій розробці веб-інтерфейсів, ми часто використовуємо SVG (масштабовані векторні графіки) для створення графічних елементів, таких як іконки, логотипи та ілюстрації, які мають високу якість при будь-якому розмірі екрану. Ось як ми застосовуємо основні властивості SVG у нашій роботі:

- **Fill**
- **Stroke**
- **Stroke Width**

Кастомізація Tailwind CSS

Налаштування конфігураційного файлу `tailwind.config.js` дозволяє кастомізувати кольори, шрифти, простори, розміри елементів, медіа-запити, теми оформлення і багато іншого.

Для додавання власних стилів, класів, кольорів, розмірів тощо можна використовувати наступний код:

```
module.exports = {  
  theme: {  
    extend: {  
      colors: {  
        primary: "#cc3333", // Додаємо новий колір 'primary'  
      },  
    },  
  },  
};
```

Після цього можна використовувати `.bg-primary` для задання фонового кольору.

Розширення можливостей фреймворку включає створення власних **утилітних класів**, компонентів та інтеграцію з іншими бібліотеками, забезпечуючи повну свободу кастомізації під потреби проекту.

Використання плагінів забезпечує додаткові можливості, наприклад, для додавання нових утиліт або функціональних елементів:

- @tailwindcss/forms
- @tailwindcss/typography
- @tailwindcss/aspect-ratio
- ... etc ...

Підсумки

1. Tailwind CSS значно прискорює і спрощує процес верстки сайтів.
2. За рахунок утилітних класів можна швидко створювати адаптивні дизайни під усі пристрої.
3. Гнучкість та простота масштабування проектів в майбутньому.
4. Він допомагає дотримуватися кращих практик в організації CSS коду.
5. Основні переваги для адаптивної верстки:
 - a. Готові утилітні класи замість ручного написання CSS
 - b. Вбудована підтримка респонсивності
 - c. Оптимізований та чистий CSS код
 - d. Легкість підтримки та зміни стилів в майбутньому

Принципова відмінність Tailwind CSS від Bootstrap

Bootstrap - це фреймворк, який базується на компонентах.

Tailwind CSS - це фреймворк, який базується на утилітарних класах.

Принципова відмінність іменування класів Bootstrap і Tailwind CSS полягає у тому, що **Bootstrap** надає **готовий набір компонентів з певною стиллю** іменування класами CSS (наприклад: **btn-primary**), а Tailwind CSS надає **готовий набір утиліт з певною стиллю іменування класами CSS** (наприклад: **bg-blue-500 text-red**).

Домашнє завдання

Tailwind CSS: від нуля до майстра веб-дизайну

Складність: **Важке**

Для вашого домашнього завдання не ставляться жодні обмеження за складністю. Якщо відчуваєте, що хочете спробувати свої сили в чомусь простому, будь ласка, реалізуйте простий проект за використанням Tailwind CSS. Якщо ж ви прагнете виклику та хочете зробити щось складніше, сміливо втілюйте свої ідеї в життя та надсилайте результати. Ви також можете повторити проект, що розглядався в класі, за викладачем, якщо це більше до вашого смаку.

При здачі проекту, будь ласка, надайте два посилання:

1. Посилання на сторінку GitHub Pages, де розміщено результат вашої роботи з Tailwind CSS, щоб його можливо було переглянути у дії.
2. Посилання на сторінку свого репозиторію на GitHub, де знаходиться код проекту.

Бажаю успіху та чекаю на ваші проекти!