

Michał Kortas



Tailwind CSS

Projektowanie stron WWW i podejście utility-first

Helion 

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz wydawca dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz wydawca nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Redaktor prowadzący: Grzegorz Krzystek
Projekt okładki: Studio Gravite

Helion S.A.
ul. Kościuszki 1c, 44-100 Gliwice
tel. 32 231 22 19, 32 230 98 63
e-mail: helion@helion.pl
WWW: <https://helion.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
<https://helion.pl/user/opinie/tailwi>
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

ISBN: 978-83-283-8377-7

Copyright © Helion S.A. 2022

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

Spis treści

Od autora	9
Podziękowania	11
Przedmowa	13
Rozdział 1. Krótka historia Tailwinda	19
Rozdział 2. Wsparcie przeglądarek	21
Rozdział 3. Tailwind w teorii	23
Instalacja	23
PostCSS 8	23
PostCSS 7	24
Bez PostCSS	25
Używanie źródła z CDN	25
Tailwind CSS i Next.js	25
Podejście utility-first	27
Różnica pomiędzy stylami liniowymi	29
Generowanie własnych klas dla najczęściej powtarzanych elementów	29
Responsywność	30
Punkty graniczne w responsywności	30
Podejście mobile-first	30
Stany i pseudoklasy	31
Style elementu względem stanu rodzica	31
Tryb ciemny (nocny)	32

Style bazowe	33
Własne style bazowe	34
Wyłączanie domyślnych stylów bazowych	34
Wydzielanie komponentów i ich ponowne wykorzystanie	34

Rozdział 4. Przykłady stron wykorzystujących Tailwind CSS 37

Laravel	37
Eurovision	38
GitHub Copilot	38
Netlify	39

Rozdział 5. Przygotowanie do projektu 41

Tworzenie szablonu startowego za pomocą Flexbokska	41
Struktura szablonu z wykorzystaniem Flexbokska	41
Typografia	44
Przyciski	46
ButtonSuccess	48
ButtonDanger	49
ButtonInfo	49
ButtonPrimary	49
ButtonWarning	50
Podgląd przycisków	50
Elementy formularza	51
Pole tekstowe	51
Lista rozwijana	52
Rozszerzone pole tekstowe	52
Pole zaznaczania wielokrotnego	53
Pole zaznaczania jednorazowego	53
Podgląd elementów formularza	53
Karty	54
Odnośniki do logowania i rejestracji	55

Rozdział 6. Praktyczne projekty — budowa panelu administracyjnego aplikacji 59

Założenia	59
Wybrane elementy interfejsu użytkownika	60
Formularz logowania	60
Formularz rejestracji	66
Dwukolumnowy szablon panelu administratora	68
Nawigacja główna	71

Rozwijane menu użytkownika	76
Wyszukiwarka	80
Stopka	81
Kafelki stanu aplikacji	83
Zapis na newsletter	88
Wezwanie do działania — Call To Action	92
Lista z komentarzami	96
Okno modalne	100
Warianty cenowe	105
Ładowanie	110
Tabela	112
Paginacja	117
Breadcrumbs	120
Artykuł	124
Alerty	129
Pytania i odpowiedzi — FAQ	134
Formularz kontaktowy	136
Galeria	140
Kalendarz	144

Rozdział 7. Just-in-Time — kompilacja w czasie rzeczywistym 151

Czym jest JIT mode w Tailwind CSS?	151
Możliwości trybu JIT	151
Wszystkie warianty są dostępne	151
Wartości dla atrybutów podajemy (prawie) dokładnie tak, jak wyglądają w rzeczywistości	152
Łączenie wielu wariantów	152
Dostępny modyfikator !important	152
Mnóstwo dostępnych pseudoklas	152
Selektor sąsiada	152
Brak konieczności używania klas .transform, .filter i .backdrop-filter	153

Rozdział 8. Palety kolorów 155

Rozdział 9. Zaawansowana konfiguracja 157

Własne przedziały responsywności (breakpointy)	157
Dodawanie kolejnego, szerszego punktu	157
Dodawanie kolejnego, pośredniego punktu	158
Specjalna klasa dla druku	159

Własne palety kolorów	159
Podmiana startowych kolorów na inne	159
Własna paleta kolorów	160
Rozszerzanie domyślnej palety kolorów	160
Rozszerzenia (pluginy)	161
Oficjalne rozszerzenia Tailwind CSS	161
Prefiksy klas	165
Tworzenie własnych rozszerzeń	165
Dodawanie wariantów	166
Własne komponenty	167
Rozszerzenia stylów bazowych	167
Podsumowanie	168

Rozdział 10. Przydatne miejsca i narzędzia ułatwiające pracę z Tailwindem 171

Dokumentacja Tailwind CSS	171
Tailwind UI	171
Oficjalny blog Tailwind	171
Twitter @adamwathan	173
Tint	173
Windy	174
Tailwind.pl	175

Rozdział 11. Optymalizacja projektu dla środowiska produkcyjnego 177

Analiza i usuwanie zbędnych klas w kodzie wynikowym	177
Wskazywanie lokalizacji plików do analizy	178
Ręczne wskazywanie klas, które mają być zawsze dołączone	178
Usuwanie nieużywanych animacji	179

Zakończenie 181

Rozdział 7.

Just-in-Time — kompilacja w czasie rzeczywistym

O JIT (*Just-in-Time*) w Tailwindzie kilkakrotnie już wspominałem, gdy omawiałem wykorzystywanie dynamicznych klas w kodzie naszej aplikacji. Warto przyrzeć się mu nieco bliżej.

Czym jest JIT mode w Tailwind CSS?

JIT to — w chwili pisania tej książki — eksperymentalna funkcja kompilatora na żądanie. Dzięki niej nie musimy już tworzyć pluginów lub nowych wariantów w pliku konfiguracyjnym Tailwinda. Unikalne i wymyślne klasy tworzymy w intuicyjny sposób, np. `.top-[100px]`, gdzie w nawiasie kwadratowym znajduje się faktyczna wartość, którą chcemy przypisać do nowej klasy CSS.

Kompilacja kodu CSS w tym trybie również jest znacznie szybsza — wynosi od niecałej sekundy przy pierwszym uruchomieniu do kilku milisekund przy przyrostowym rozszerzaniu już istniejącego pliku wynikowego.

Możliwości trybu JIT

Zalet trybu JIT jest wiele, ale wymienię tylko te według mnie najciekawsze. Cała lista zalet dostępna jest w dokumentacji¹.

Wszystkie warianty są dostępne

Nie trzeba już aktywować wariantów z góry niedostępnych w niektórych pluginach Tailwinda. Jeśli potrzebujemy wariantu `.focus-visible:`, to po prostu go używamy. Żadnej zbędnej konfiguracji.

¹ <https://tailwindcss.com/docs/just-in-time-mode>.

Wartości dla atrybutów podajemy (prawie) dokładnie tak, jak wyglądają w rzeczywistości

Wyjątkiem są wartości zawierające spacje. Należy te spacje usunąć lub jeśli są konieczne — zastąpić przecinkami.

Spójrz na kod dostępny w dokumentacji:

```
<!--Rozmiary i pozycjonowanie -->


<!-- Kolory -->
<button class="bg-[#1da1f1]">Share on Twitter</button>

<!-- Siatki -->
<div class="grid-cols-[1fr,700px,2fr]">
  <!-- ... -->
</div>
```

Wszystko można skonfigurować i nawet największy perfekcjonista będzie zadowolony. Dzięki nawiasom kwadratowym od razu wiemy, że opuszczamy bezpieczne, predefiniowane opcje i tworzymy własne.

Łączenie wielu wariantów

Dzięki JIT możemy łączyć wiele wariantów w jeden. Nie trzeba już pisać własnego CSS-a, aby poradzić sobie ze skomplikowanymi sytuacjami:

```
<button class="md:dark:disabled:focus:hover:bg-gray-400">
```

Dostępny modyfikator !important

Jeśli zostaliśmy zmuszeni do nadpisania czyjegoś stylu i nie możemy go zmodyfikować u źródła, wystarczy zastosować znak wykrzyknika przed klasą.

Mnóstwo dostępnych pseudoklas

Oprócz znanych pseudoklas `.hover:` czy `.focus:` możemy używać teraz wielu innych, jak na przykład `.invalid:` lub `.required:`.

Selektor sąsiada

Podobnie jak w przypadku selektora rodzica (`.group` i na przykład `.group-hover`), możemy teraz do sąsiada wybranego elementu przypisać klasę `.peer`. Następnie style dla samego elementu należy poprzedzić wariantem `.peer-hover:` lub innym.

Brak konieczności używania klas `.transform`, `.filter` i `.backdrop-filter`

O tym również wspominałem, ale dla usystematyzowania wiedzy przypominam ponownie. Aby użyć filtru, nie trzeba już klasy odpowiedzialnej za efekt poprzedzać klasą `.filter`. To samo dotyczy transformacji elementu oraz filtru pod elementem.

PROGRAM PARTNERSKI

— GRUPY HELION —



1. ZAREJESTRUJ SIĘ
2. PREZENTUJ KSIĄŻKI
3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

<http://program-partnerski.helion.pl>

GRUPA
Helion 

Poznaj rewelacyjną nowość na rynku frontendu

- Zalety i wady Tailwind CSS
- Tailwind CSS w teorii
- Praktyczne projekty

Framework Tailwind CSS to prawdziwy webmasterski hit ostatniego roku. Wyjątkowym i nowatorskim czyni go przyjęta filozofia pracy. Twórcy frameworka proponują bowiem oryginalne podejście do tworzenia stron i aplikacji internetowych. Zamiast ściśle semantycznego sposobu tworzenia klas CSS Tailwind opiera się na fundamencie zwanym utility-first. Dzięki temu architekt interfejsu nie marnuje czasu na generowanie wymyślnych nazw opisujących poszczególne elementy, a podchodzi do nich bardziej pragmatycznie — definiuje je w sposób, który najlepiej odpowiada rzeczywistości.

To książka przeznaczona przede wszystkim dla początkujących i średnio zaawansowanych projektantów stron WWW. Znalazło się w niej mnóstwo praktycznych informacji dotyczących frameworka Tailwind. Poznasz jego możliwości, mocne i słabe strony. Opanujesz rządzące nim zasady, a także dostępne w nim narzędzia. Ich obsługi nauczysz się w praktyce, ponieważ znaczną część podręcznika zajmuje realizacja konkretnego zadania — autor przeprowadzi Cię przez proces tworzenia kompletnego konceptu interfejsu z wykorzystaniem wszystkiego, co oferuje w swoich zasobach Tailwind.

Michał Kortas jest autorem bloga *webroad.pl* poświęconego szeroko pojętym zagadnieniom technologii webowych. Nieprzerwanie od blisko dziesięciu lat spełnia się zawodowo jako architekt projektów i full-stack web developer. Jest autorem książki *Bootstrap. Praktyczne projekty*, wydanej nakładem wydawnictwa Helion.

Helion



helion.pl



HELION SA
ul. Kościuszki 1c
44-100 Gliwice
tel.: 32 230 98 63
helion@helion.pl

Sprawdź nasze szkolenia!



AKADEMIA IT & BUSINESS

HELIONSZKOLENIA.PL

KOD KORZYŚCI
Sięgnij po więcej! ▶



ISBN 978-83-283-8377-7



INFORMATYKA W NAJLEPSZYM WYDANIU

Cena: 59,00 zł