



VŠB TECHNICKÁ
UNIVERZITA
OSTRAVA

FAKULTA
ELEKTROTECHNIKY
A INFORMATIKY

KATEDRA
INFORMATIKY

Základy informačních technologií

Ing. Michal Radecký, Ph.D. MBA

Front-end frameworky, atd.



Front-end frameworky

Nabízejí nástroje pro efektivní implementaci webdesignu

Vychází z CSS frameworků (např. Blueprint – mřížka, typografie, formulářové prvky)

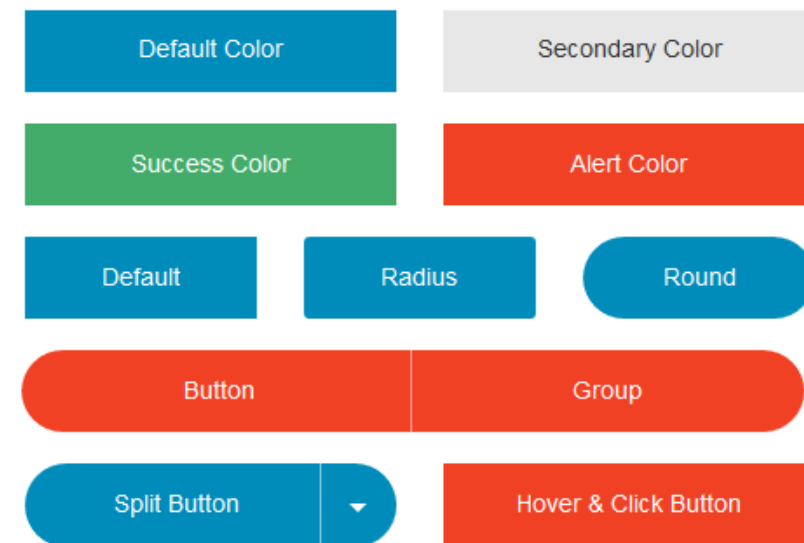
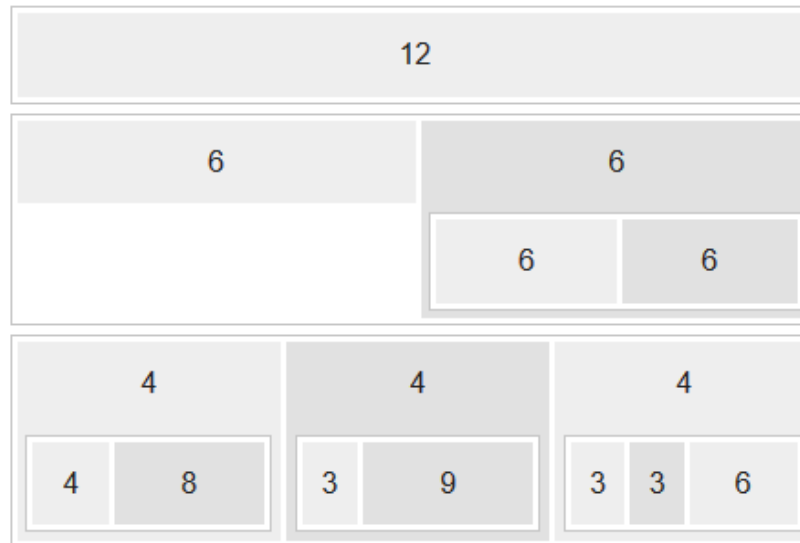
Přinášejí především ...

- podporu responsibility a multiplatformnost
- tvorbu layoutů
- práci s typografií
- uživatelské ovládací prvky
- rozšiřující knihovny

Bootstrap, Tailwind, Foundation, Skeleton, atd.

Foundation

- <http://foundation.zurb.com/>
- Verze 6, MIT licence, cca 150 kB
- Preprocesor SASS, Vanilla JS (ES5)
- Modulární architektura (vlastní build) vč. JS komponent
- Komplexní a flexibilní



Bootstrap

- <http://getbootstrap.com/>
- Verze 5, MIT licence, cca 250 kB
- Preprocesor SASS od verze 4, Vanilla JS (ES6)
- Částečně modulární (na úrovni kódu)
- Velká komunita, mnoho komponent, šablon atd.
- Snadná a flexibilní integrace

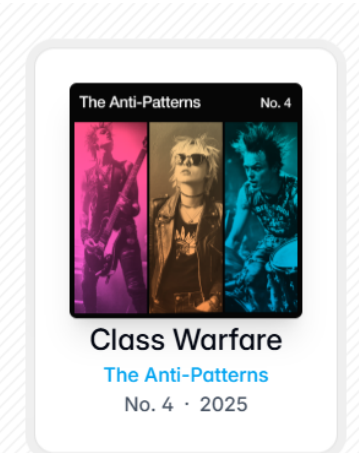
span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1
span 4				span 4				span 4			
span 4				span 8							
span 6						span 6					
span 12											



Tailwind

- <http://tailwindcss.com/>
- Verze 4, MIT licence, cca 300 kB (dynamický)
- Nepracuje s komponenty, ale „utilitami“ – sada tříd nahrazujících přímou tvorbu CSS
- Vysoká flexibilita a popularita
- Tailwind Plus (UI) – modulární a komponentní rozšíření vč. JS

```
1 <div class="flex flex-col items-center gap-6 p-7 rounded-2xl">
2   <div>
3     
4   </div>
5   <div class="flex items-center">
6     <span class="text-2xl font-medium">Class Warfare</span>
7     <span class="font-medium text-sky-500">The Anti-Patterns</span>
8     <span class="flex gap-2 font-medium text-gray-600 dark:text-gray-400">
9       <span>No. 4</span>
10      <span>•</span>
11      <span>2025</span>
12    </span>
13  </div>
14 </div>
```



Proč používat stylovací frameworky

Vždy je třeba zvážit efektivitu a rozsah použití pro konkrétní projekt

Výhody

- Rychlý vývoj v rámci „standardizovaného“ prostředí
- Nativní podpora CSS preprocesorů
- Implementace většiny běžně používaných prvků (i s ohledem na multiplatformnost)
- Malá vstupní bariéra při efektních výsledcích
- Optimalizace díky kompozitní konstrukci
- Podpora přizpůsobení a témat
- Podpora ve vývojových nástrojích (rozšíření ve VSCode) a webových frameworkcích
- Licenční pravidla pro široké nasazení (MIT)

Nevýhody

- Z určitého pohledu až příliš komplexní
- Vazba na celou řadu dalších nástrojů a knihoven s nutností jejich propojení, správy, apod. (node.js, Grunt, Rails, Compass, atd.)

Bootstrap - grid

<https://www.w3schools.com/bootstrap5/index.php>

- Základem rozložení stránky je **grid systém** (12 buněk)
- Kontejner (responzivní/fluidní) – řádek – buňky
- Rozdělení podle rozlišení – jinak pod sebe, možnost kombinace

`.col-` (extra small devices - screen width less than 576px)

`.col-sm-` (small devices - screen width equal to or greater than 576px)

`.col-md-` (medium devices - screen width equal to or greater than 768px)

`.col-lg-` (large devices - screen width equal to or greater than 992px)

`.col-xl-` (xlarge devices - screen width equal to or greater than 1200px)

`.col-xxl-` (xxlarge devices - screen width equal to or greater than 1400px)

Bootstrap – stylování vzhledu

Nativní stylování – přímé stylování některých elementů (h1-6, attr, blockquote, code, ...)

Komponentní stylování – v první řadě využívat existující komponenty (.btn)

Utility stylování – pro přizpůsobení primárně používat utility (.text-center)

Nastavování proměnných – nastavení CSS proměnných přizpůsobí výsledný vzhled (--bs-link-color)

Uživatelské stylování – přizpůsobování existujícího stylování (vlastní class)

Nikdy nezasahovat do originálních zdrojů Bootstrapu, ale vytvářet vlastní v režimu „přepisování“!

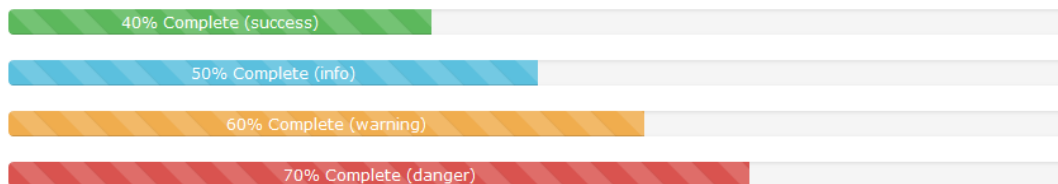
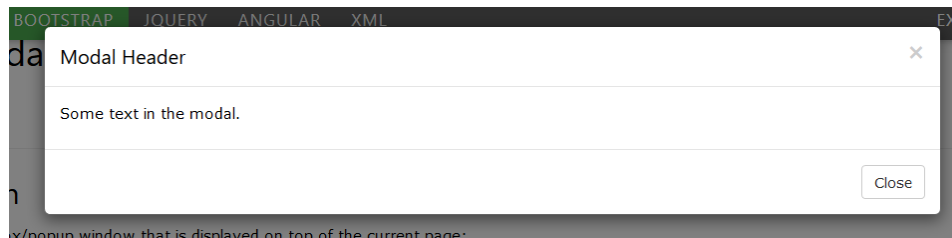
- Základní princip je „nestylovat přímo“, ale vše pomocí předdefinovaných stylů. V podstatě (pro hlavní úroveň) není třeba vůbec vytvářet vlastní stylování.
- Tomu musí odpovídat i struktura HTML (modal dialog)
- Pro složitější interakci se využívá API postavené na JS
- Díky gridu jednoduchá implementace přístupu „mobile-first“

Bootstrap – komponenty

Výběr konzistentních vizuálních prvků (včetně funkčnosti) jen díky kombinaci HTML a nastavených CLASS, příp. JS.

- **Strukturální prvky** - navbar, header, footer, grid, container, layout utilities
- **Obsahové prvky** - cards, list groups, tables, alerts, badges, text utilities
- **Formulářové prvky** - inputs, selects, toggles, sliders, floating labels, validation, states
- **Interaktivní prvky (JS)** – modal, tooltip, accordion, carousel

Není potřeba psát JS, stačí data atributy (data-bs-*)



Success! This alert box indicates a successful or positive action.

Info! This alert box indicates a neutral informative change or action.

Warning! This alert box indicates a warning that might need attention.

Danger! This alert box indicates a dangerous or potentially negative action.

Bootstrap – interakce

Využívá DOM-driven přístup – vše je definováno v HTML, a to pomocí data atributů. Ty se při načtení Bootstrapu vyhledají a aktivují odpovídající JS. Stále je možnost doplnit vlastní JS implementaci (události, apod.) vč. Bootstrap API.

Data atributy umožňují

- aktivovat komponenty
- měnit stav
- řídit interakce

Současně Bootstrap využívá **ARIA atributy** (aria-expanded, aria-controls, role=) k zajištění plné přístupnosti a správného chování pro čtečky a asistivní technologie.

```
<button type="button" class="btn btn-primary"  
  data-bs-toggle="modal"  
  data-bs-target="#exampleModal">
```

Open modal

```
</button>
```

```
<!-- Modal -->  
<div class="modal fade" id="exampleModal" tabindex="-1" aria-hidden="true">  
  <div class="modal-dialog">  
    <div class="modal-content">  
  
      <div class="modal-header">  
        <h5 class="modal-title">Modal title</h5>  
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>  
      </div>  
  
      <div class="modal-body">  
        Short modal text.  
      </div>  
  
      <div class="modal-footer">  
        <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Close</button>  
      </div>  
  
    </div>  
  </div>  
</div>
```

Bootstrap – pluginy a témata

Mimo nativních komponent je možné využívat i rozšíření třetích stran či vlastní

- Material Design - <https://mdbootstrap.com/>
- Pixel UI Kit - <https://github.com/themesberg/pixel-bootstrap-ui-kit>
- Telerik - <https://www.telerik.com/design-system/docs/themes/kendo-themes/bootstrap>

Verze Bootstrapu nejsou kompatibilní, takže je vždy nutné zohledňovat verze!

Mimo balíky komponent je možné i implementovat komplexní šablony (templates), které jsou zaměřené na konkrétní vizuální úpravy a specifickou funkčnost

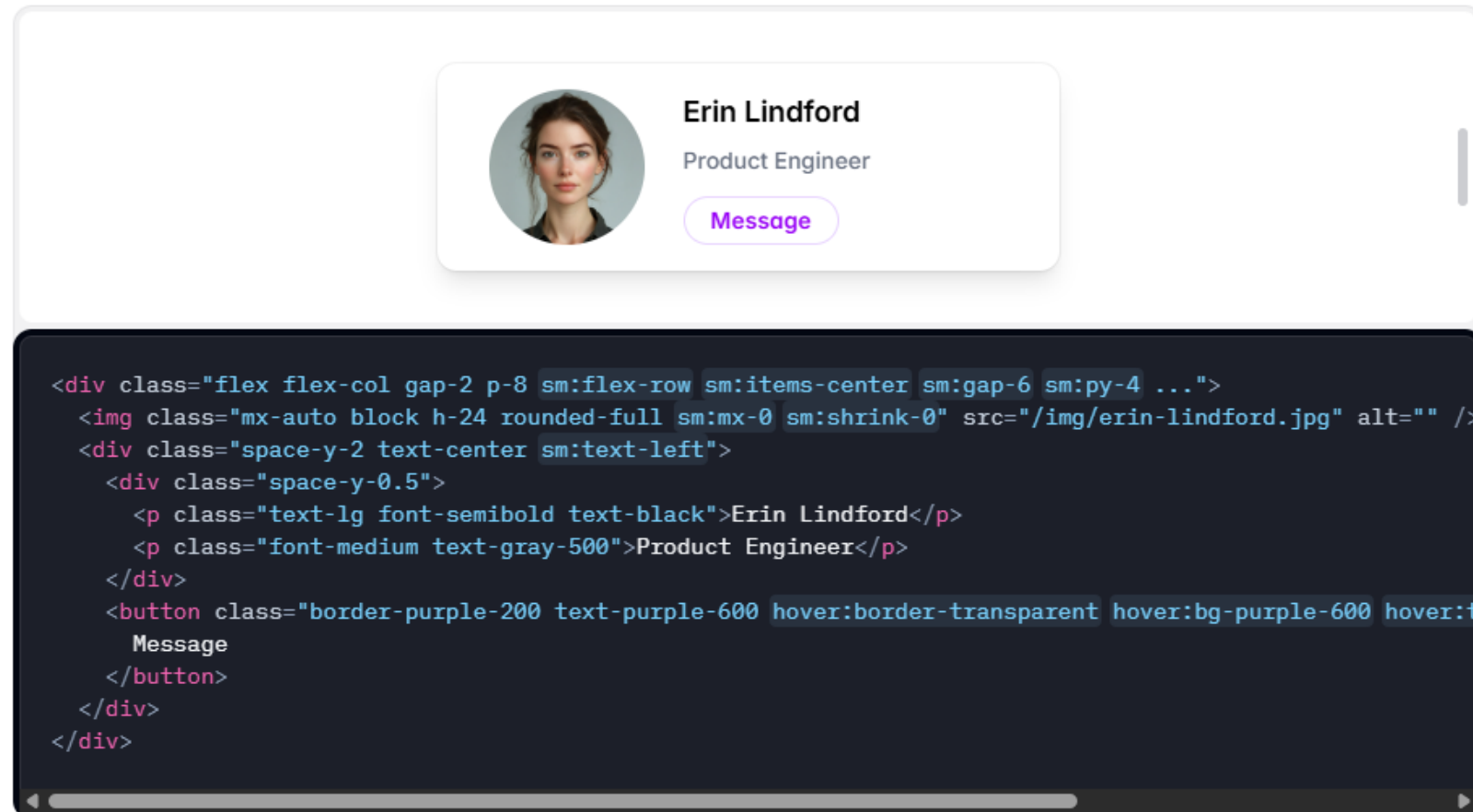
- AdminKit - <https://admindkit.io>
- AyroUI - <https://ayroui.com>
- CoreUI - <https://coreui.io/product/free-bootstrap-admin-template>

Rozšiřující balíky mohou být zdarma (MIT, General Public License) nebo komerční.

Tailwind – základní koncept

Neobsahuje komponenty - **Utility-first CSS framework** - stylování pomocí malých, opakovaně použitelných tříd, kdy každá třída má jednu odpovědnost (např. p-4, text-center, bg-blue-500)

- **Kompozice** - kombinací utilit vzniká složitější stylování a vzhled
- **Mobile-first** - třídy lze cílit podle breakpointů (sm:, md:, lg:...)
- **Varianty stavu** - hover, focus, active, disabled (hover:bg-blue-700)
- **Dark mode** - dark: varianty
- **JIT engine** - generuje pouze použité utility – minimální výsledné CSS

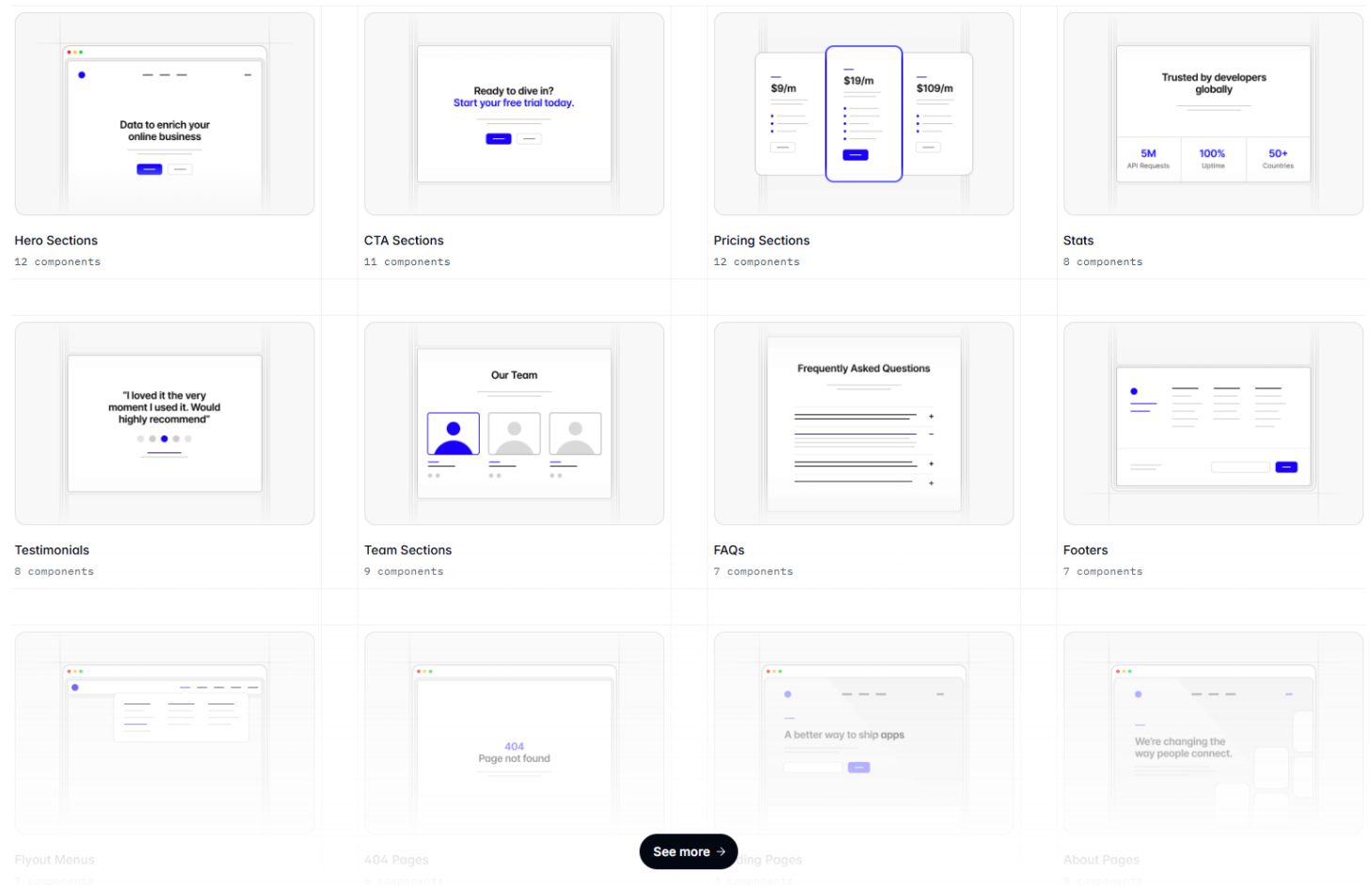


Tailwind – komponenty

Tailwind je pouze CSS. Rozšíření o komponenty nebo interaktivní prvky lze pomocí knihoven.

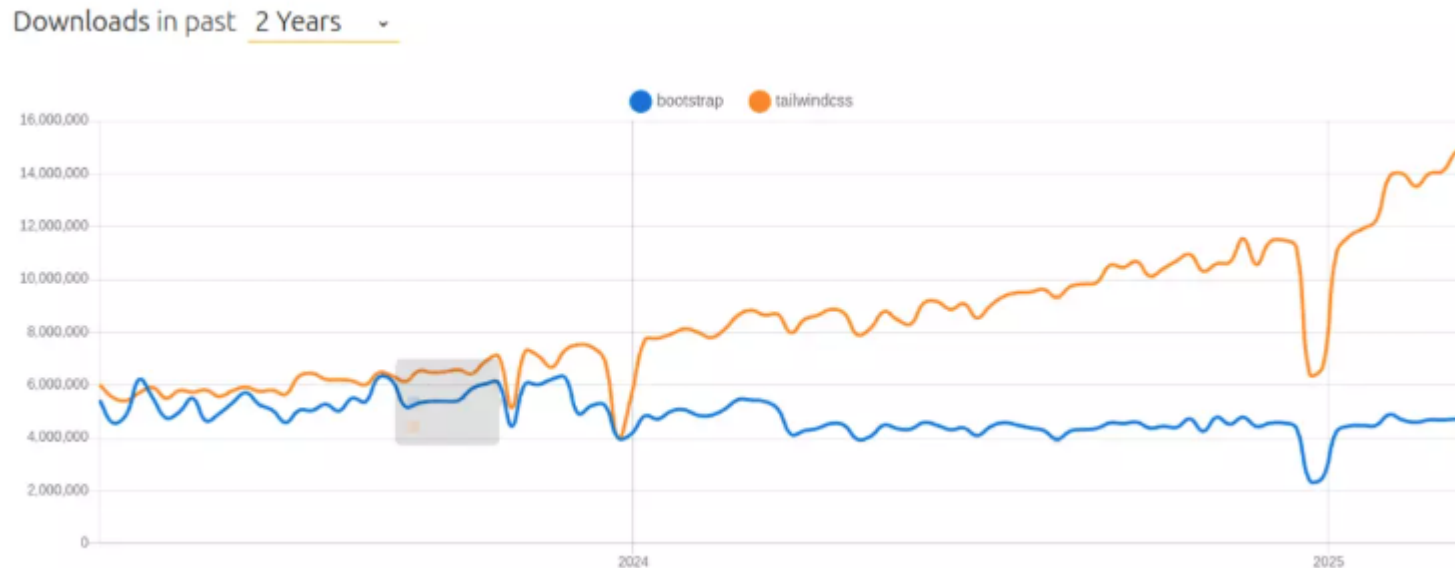
- Tailwind Plus - <https://tailwindcss.com/plus>
- DaisyUI - <https://daisyui.com>
- Flowbite - <https://flowbite.com>

Je možné vzhled přizpůsobovat,
a to nastavováním CSS proměnných
nebo pracovat se šablonami (@theme)



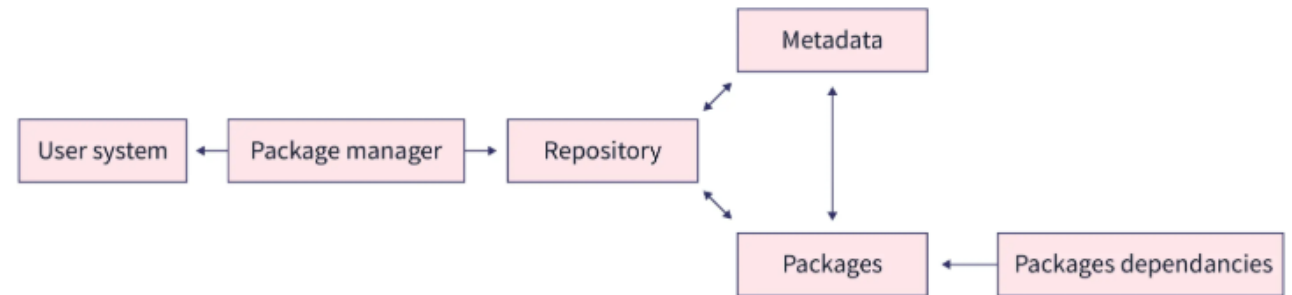
Shrnutí

- Frontend CSS frameworky mají své opodstatnění a mohou výrazně zvýšit efektivitu a výsledný efekt webových aplikací.
- Hodně věcí je možné řešit bez znalosti CSS či JS, nicméně u komplexních aplikací je vazba na další webové technologie nutnost.
- Využití frameworků na hlubší úrovni znamená využití CSS preprocesoru a dalších integračních nástrojů.
- Klíčová je volba konkrétního frameworku a navazujících možností.



Balíčkovací nástroje

- Nástroje pro instalaci, správu a aktualizaci knihoven, frameworků a závislostí. Moderní projekty mají stovky závislostí – ruční správa je prakticky nemožná.
- Umožňují propojit projekt s externími balíčky: CSS frameworky, JS knihovny, utility
- Zajišťují konzistenci verzí podle **package.json** - odstranění ručního stahování ZIP archivů, správa probíhá automaticky
- Automatizace build procesu (kompilace SCSS, TypeScriptu, minifikace atd.)
– CI/CD (Continuous Integration/Continuous Delivery/Deployment)



Balíčkovací nástroje – jak fungují

Každý balíček je umístěn v registru (privátní/veřejné, přidání do registru vyžaduje publikační a schvalovací proces)

- **NPM registry – největší, webové frameworky, nástroje (npm, yarn, pnpm - registry.npmjs.org)**
- PyPI – centrální registr pro Python balíčky (pip)
- RubyGems – balíčky pro Ruby (gems)
- NuGet registry - .NET platforma (nuget)
- GitHub, GitLab, atd.

Integrace balíčku

1. Balíček se podle jména (verze) vyhledá v registru (npm install bootstrap)
2. Získají se metadata balíčku (verze, závislosti, konfigurace, ...) a zapíší se do lock souboru (package-lock.json)
3. Stáhnou se zdrojové soubory balíčku (node_modules)
4. Vyřeší se závislosti – balíček potřebuje další balíčky, dependency tree/graph – lock soubory
5. Kontroluje se integrita a podpisy, zda nedošlo ke změně oproti registru
6. Je možné také spouštět „postinstalační“ skripty, např. pro kompilaci či build

Balíčkovací nástroje – package.json

- Jedná se o manifest projektu (npm/yarn/pnpm).
- Definuje informace o projektu, závislostech, skriptech, build procesech a metadatech.
- Komplexní problematika řízení a vyhodnocení verzí – lock soubor udržuje stav
- Dále může obsahovat
 - Konfigurace prostředí projektu
 - Konfigurace konkrétních balíčků
 - Data pro publikaci balíčku
- Podpora definice a spouštění skriptů (automatizace)

npm run release
- lint, test, build, deploy

```
{
  "scripts": {
    "dev": "vite",
    "lint": "eslint src/",
    "test": "jest",

    "prebuild": "npm run lint && npm run test",
    "build": "vite build",

    "release": "npm run build && npm run deploy",
    "deploy": "node ./scripts/deploy.js"
  }
}
```

```
{
  "name": "version-demo",
  "version": "1.0.0",
  "private": true,

  "scripts": {
    "dev": "vite",
    "build": "vite build"
  },

  "dependencies": {
    "react": "^18.2.0",           // 1) caret: povoluje minor + patch
    "react-dom": "~18.2.0",      // 2) tilde: povoluje POUZE patch
    "axios": "1.6.2",            // 3) přesná verze, žádné aktualizace
    "dayjs": ">=1.11.0 <2.0.0",  // 4) rozsah verzí (range)
    "uuid": "*"                  // 5) libovolná verze (nedoporučuje se)
  },

  "devDependencies": {
    "vite": "^5.0.0",            // nejběžnější zadání (caret)
    "eslint": "~9.0.0",          // patch aktualizace
    "jest": "29.x",              // 6) povoluje libovolný minor/patch v major verzi
    "tailwindcss": "latest"      // 7) speciální npm tag (instaluje poslední verzi)
  }
}
```

Balíčkovací nástroje

NPM, Yarn, Pnpm

- Velmi podobné principy
- Různé principy práce s cache (node_modules)
- Rozdíly v rychlostech, API
- Umožňují lokální i globální instalace

Features	NPM	Yarn	PNPM
Package Installation	'npm install package-name'	'yarn add package-name'	'pnpm add package-name'
Dependency Tracking	package.json	yarn.lock or package.json	'package.json and pnpm-lock.yaml'
Scripts	'npm run script-name'	'yarn run script-name'	'pnpm run script-name'
Global Packages	Supported	Supported	Supported
Private Packages	Supported	Supported	Supported
Version Control	package-lock.json	yarn.lock	pnpm-lock.yaml
Security	Basic HTTPS download validation	Checksum validation	Checksum validation
Performance	Slower than Yarn (sequential downloads)	Faster than NPM (parallel downloads)	Faster than NPM (parallel downloads)
Offline Support	Yes	Yes	Yes

Balíčkovací nástroje

Výhody

- Automatické aktualizace
- Menší riziko konfliktů verzí a kontrola závislostí
- Rychlejší vývoj díky skriptům (npm run build)
- Vazby na další pokročilé vývojářské mechanismy
 - Buildovací nástroje – webpack, vite, esbuild, atd.
 - Transpilery – TypeScript, SASS, atd.
 - Frameworky – Angular, React, atd.
- Bezpečnostní mechanismy pro závislosti
- Snadné sdílení projektu (bez obsahu balíčků) a reprodukovatelnost projektu v různých prostředích

Nevýhody

- Možná závislost na použitých nástrojích a jejich internímu fungování
- Složitější ekosystém a řízení projektů
- Velká dynamika verzí balíčků a jejich problematické sledování/kontrola
- Složitější konfigurace ve vazbě na navazující nástroje a prostředí