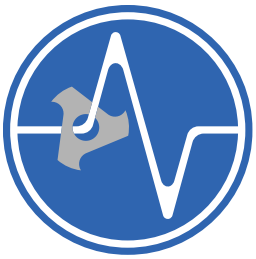


Střední průmyslová škola strojní
a elektrotechnická a Vyšší odborná škola,
Liberec 1, Masarykova 3

NÁVRH A REALIZACE WEBOVÝCH STRÁNEK

Ročníková práce

Autor **Dominik Lazna**
Obor **Technické lyceum**
Vedoucí práce **Ing. Tomáš Kazda, DiS.**
Školní rok **2022/2023**

Anotace

Práce se zabývá pravidly a konvencemi pro psaní optimálně strukturovaných a minimalisticky kódovaných webových stránek. V první kapitole seznamuje čtenáře se základními technologiemi a poté se již věnuje nejlepším postupům ve zmíněných technologiích.

Summary

The work deals with the rules and conventions for writing optimally structured and minimally coded websites. The first chapter introduces the reader to the basic technologies and then deals with the best practices in the mentioned technologies.

Čestné prohlášení

Prohlašuji, že jsem předkládanou maturitní práci vypracoval sám a uvedl jsem veškerou použitou literaturu a bibliografické citace.

V Liberci dne 02.06.2023 .....

Dominik Lazna

Obsah

Úvod

S webovými stránkami se v dnešní době setkal už nejspíš každý. Ať už si chceme přečíst článek, novou zprávu ze světa nebo koupit něco nového na sebe či do bytu, je zde velká pravděpodobnost, že tak učiníme právě na nějaké webové stránce. Web se stal důležitou součástí našeho každodenního života a vývojáři, kteří na něj umisťují webové stránky by měly dělat maximum pro to, aby se na něm lidé dobře orientovali a našli to co hledají.

Rozhodl jsem se tedy napsat práci na téma Návrh a realizace webových stránek, abych se pokusil vylepšit a přiblížit schopnost vytvářet webové stránky nejen sebe, ale i ostatní čtenáře. Z tohoto důvodu se dokument odkazuje na vytvořenou webovou stránku, na které ukazuji reálné příklady a využití nejlepších postupů, které jsou v práci zmíněny. Dalším cílem práce bylo vytvořit stručné kodérovo desatero, které obsahuje výtah nejlepších postupů a konvencí v přístupném formátu a mohlo by být využito i v budoucích třídách oboru Informační technologie.

1 Technologie

1.1 HTML

HTML (HyperText Markup Language) je nejzákladnějším stavebním blokem webových stránek. Definuje strukturu a význam obsahu. Umožňuje propojovat jednotlivé webové stránky mezi sebou za pomoci odkazů. (1)

HTML využívá tagy, kterými můžeme označovat jednotlivý text a přidělovat mu tak jeho význam a následně ho stylovat pomocí CSS. (1)

```
<body>
    <h1>Ukázka HTML syntaxe</h1>
    <p>h1 představuje nadpis první úrovně, zatímco tag p nový
odstavec</p>
</body>
```

1.2 CSS

CSS (Cascading Style Sheets) je technologie, která umožňuje stylovat obsah na webové stránce. Sděluje prohlížeči jak vykreslovat jednotlivé elementy našeho HTML kódu. (2)

```
h1 {
 font-size: 120%;
 color: red;
 text-align: center;
}
```

Výše je ukázka CSS kódu, který styluje jakýkoli text, který je obalený h1 tagem. Tomuto textu zvětšuje velikost fontu na 120% základní velikosti, barvu na červenou, a nakonec ho zarovnává na středu.

2 Psaní optimálně strukturovaného HTML kódu

Psaní HTML kódu není nijak složité, tento jazyk má velmi jednoduchý syntax, díky čemuž je možné se jej v rámci možností naučit i za jedno odpoledne. Stačí se naučit základní syntaxi a můžete začít vytvářet jednoduché webové stránky. To co ale dělí experta a amatéra, který se dnes rozhodl naučit HTML je psaní efektivního a lehce udržovatelného kódu.

Efektivní kód je srozumitelný a rozumí mu i vývojář který ho vidí prvně nebo po dlouhé době. Na rozdíl od špatného kódu, který je napsán bez hlubšího přemýšlení do budoucna je efektivní kód psaný tak, aby byl lehce škálovatelný a mohly se do něj přidávat další funkce bez strachu z přepisování celého souboru. (3)

2.1 Proč se snažit o efektivnější HTML kód

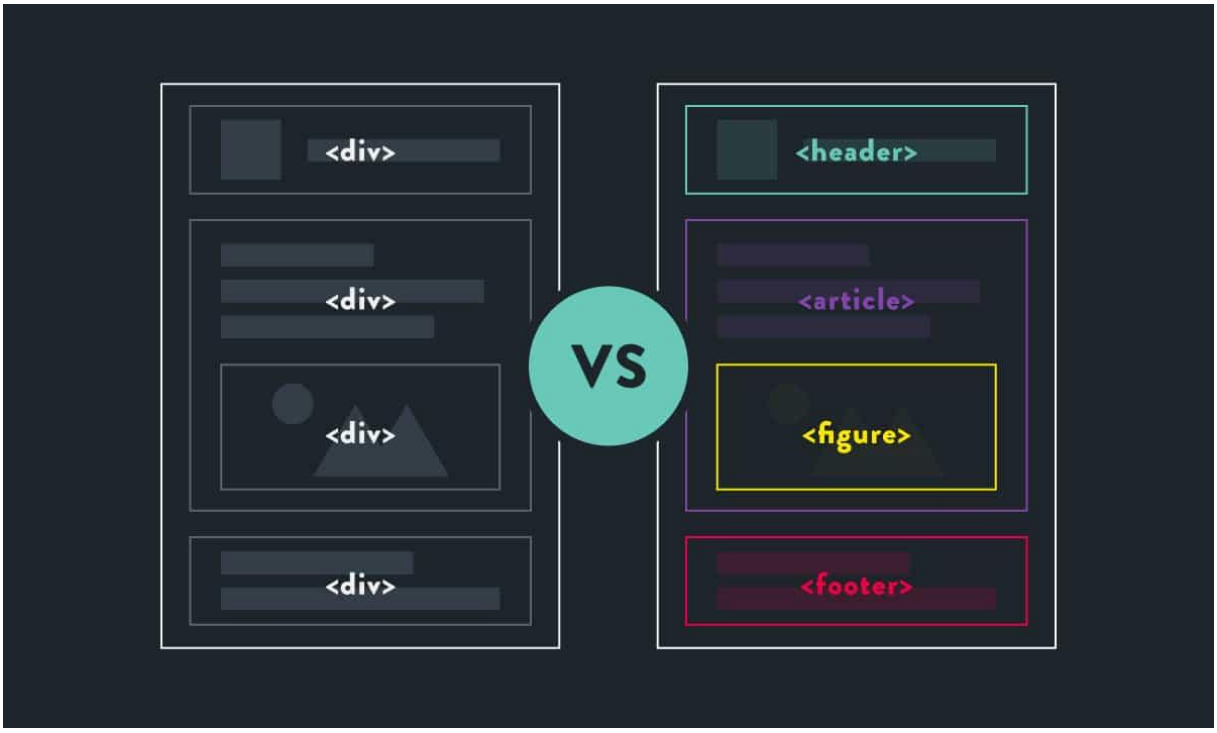
Validní HTML kód nepomáhá jen vývojářům, kteří ho píšou a spravují, ale také uživatelům, kteří jeho výslednou podobu ve formě webové stránky navštěvují. Ideálně strukturovaná stránka dokáže razantně vylepšit přístupnost pro uživatele se zrakovým postižením. Správně rozmístěné HTML elementy s některými dodatečnými atributy dělají nástrojům jako je čtečka mnohem jednodušší práci v popisování obsahu na stránce a dávají tak uživatelům lepší představu o tom co se na stránce děje. (4)

Dalším důvodem proč by vývojáři měli dávat ohled na správně strukturovaný kód je optimalizace pro vyhledávače. Webové vyhledávače totiž využívají roboty, kteří procházejí celý World Wide Web a zaznamenávají slova, která na jednotlivých stránkách vidí. Ideálně navržený HTML kód jim tuto činnost umožňuje dělat efektivněji a naše webová stránka se tak může zobrazovat většímu množství uživatelů. (4)

2.2 Sémantické HTML

Smysl sémantického HTML je přiřazování významu našemu obsahu zabalenému v konkrétních HTML tazích. Tagy, které obsahu dávají nějaký význam nazýváme sémantické tagy a mezi ty nejpoužívanější patří např. <header>, <nav>, <article>, <section>, <figure> nebo <footer>. (5) (6)

Využívání sémantických tagů oproti tagu vylepšuje web v několika již výše zmiňovaných aspektech, mezi které patří zlepšení přístupnosti pro zrakově postižené, lepší optimalizace pro vyhledávače a také přehlednost pro vývojáře. (6)



Obrázek 1 Struktura stránky bez využití sémantických tagů vlevo a s jejich využitím vpravo (7)

2.2.1 Article element

Definuje samostatný článek na webové stránce. Obsah v <article> elementu, by měl obsahovat text, který může fungovat sám na sobě, bez toho, aniž by bylo potřeba kontextu ze zbytku stránky. Využívá se například pro novinový článek, příspěvek na fóru nebo blogu. (5)

2.2.2 Section element

Jedná se pravděpodobně o nejflexibilnější sémantický element. Často se zaměňuje s elementem <article>, na rozdíl od <article> elementu může však obalovat obsah, který potřebuje ke svému pochopení kontext ze zbytku stránky. Využívá se k vytváření oddílu například právě <article> obsahu. (8)

Pokud bychom chtěli například vytvořit stránku, která obsahuje souvislý text o nejlepších basketbalistech, <article> by obaloval celý náš článek, zatímco <section> elementy by obalovaly jednotlivé oddíly tohoto článku.

2.2.3 Header element

Využívá se jako záhlaví stránky či nového obsahu, jakým může být nová sekce nebo článek. Nejvíce se využívá právě v záhlaví stránky a často obsahuje například navigaci, vyhledávač nebo logo společnosti. V případě článku může obsahovat údaje o textu – název článku, kdo je autorem, datum. (5)

2.2.4 Nav element

Jedná se o zkráceninu slova navigace, což je přesně to k čemu by se měl tento tag využívat. Využívá se k vytvoření hlavní navigace webové stránky, což ale neznamená, že se nemůže používat například i jako vedlejší menu, či menu nějaké podsekce. (5) (6)

2.3 Soubor nejlepších postupů a pravidel

2.3.1 Správné nastavení lang atributu v <html> elementu

Tento atribut specifikuje, jaký jazyk se na naší stránce nachází. Vyskytuje se v elementu <html>, kde je jeho správné nastavení také nejdůležitější. Pokud se na naší stránce však nachází více jazyků, které se nevyskytují tak hojně, je možné atribut lang nastavit i dalším elementům jako je například <section>. (9)

Atribut lang je velmi důležitý, co se týče přístupnosti pro zrakově postižené návštěvníky naší stránky, jelikož dává čtečkám najevo jakou výslovnost mají pro text zvolit. (9)

2.3.2 Doplnění tagu <img> o atribut alt

Alt neboli alternativní text slouží k poskytnutí textového popisu našeho obrázku. Využívá se v případě, že se obrázek nedokáže načíst nebo také pro zrakově postižené, jímž je předčítán za pomoci čtečky. Má využití i pro optimalizaci pro vyhledávače, jelikož vyhledávačům sděluje, co se na obrázku nachází. (12)

Text v atributu alt by měl být stručný, přesně vystihovat co se na obrázku nachází a rozhodně by neměl obsahovat náhodný text. (12) (13)

Obrázek 2 Vzorový obrázek z webové stránky

Na výše vloženém obrázku, který se vyskytuje na podkladovém webu, můžeme vidět producenta, který právě pracuje s mixerem na hudbu. Alt atribut proto obsahuje text „Pohled na producenta, který právě pracuje s mixerem na hudbu“. Text není moc dlouhý a perfektně vystihuje co se na obrázku děje.

2.3.3 Ideální struktura obrázku s popiskem

Je zde několik cest, jak docílit vytvoření obrázku, na který se váže popisek. Ta ideální by však měla zahrnovat tagy <figure> a <figcaption>.

```
<div>
  <img src=“./složka/obrázek.jgp“ alt=“Popis obrázku“>
  <p>Popisek k obrázku</p>
</div>
```

```
<figure>
  <img src=“./složka/obrázek.jpg“ alt=“Popis obrázku“>
  <figcaption>Popisek k obrázku</figcaption>
</figure>
```

I přesto že obě výše vyobrazené HTML struktury fungují, ta druhá poskytuje jisté výhody a je tedy lepším řešením.

Element <figure> slouží k vkládání soběstačného obsahu a nejčastěji se využívá právě při vkládání obrázků, grafů, schémat či videí. Zatímco v prvním vyobrazeném

příkladu slouží jen k seskupení elementů <img> a <p> pro následné aplikování stylů, element <figure> říká vyhledávačům, že vše co se v něm nachází k sobě patří. Element <figcaption> pak reprezentuje popisek zbytku obsahu v rodiči. Správně by měl být prvním nebo posledním potomkem elementu <figure>. (10) (14) (15)

3 Psaní čitelného a udržovatelného CSS kódu

Podobně jako HTML je i CSS z počátku velmi jednoduché na pochopení a použití. Stačí se naučit, jak využívat selektory, které nám zaměří naše HTML elementy a poté za pomocí vlastností modifikovat jejich výsledný vzhled na naší stránce. Pokud však nebudeme nad stylem, kterým náš CSS kód píšeme nijak přemýšlet, je velmi pravděpodobné že u větších projektů za chvíli narazíme na problémy. (16)

3.1 Výhody efektivního CSS kódu

3.1.1 Čitelnost a srozumitelnost

Jedním z hlavních benefitů efektivního CSS kódu je čitelnost. Dobrá čitelnost nám pomáhá v porozumění našeho kódu, díky čemuž je pak lehčí se k němu vracet a modifikovat ho i po delší době jeho nečtení. Jistě, když náš kód zrovna píšeme nejspíš víme, co se zhruba děje. Za jak dlouho se v něm však dokážeme zorientovat po 14denním volnu? Právě zde hraje roli čitelnost a srozumitelnost. Dbát na tyto aspekty je pak velmi důležité i v týmovém prostředí, kdy nečteme kód jen my, ale i naši kolegové, kteří na kódu pracují s námi a správně by mu tedy měli rozumět i oni. (17)

3.1.2 Škálovatelnost

Na začátku téhle kapitoly je už zmíněno, že psaní nepromyšleného CSS je velmi jednoduché a zvládne to opravdu i začátečník. Na problém s takovým kódem ale narazíme velmi brzy, jakmile se pokusíme vytvořit nějaký větší projekt, který bude obsahovat stovky řádků kódu. Vzhled naší stránky začne dělat něco co nechceme a my zjistíme, že pokud chceme chybu opravit, musíme upravit naše CSS hned na několika místech najednou, či zcela odstranit třetinu souboru a napsat vše úplně jinak. (17) (16)

Efektivní kód nám na rozdíl od toho nepromyšleného umožňuje provádět změny bez nutnosti přepisování poloviny souboru či psaní dalších stovek řádků kódu. (17)

3.2 Postupy pro psaní efektivního kódu

3.2.1 Princip DRY

Zkratka DRY stojí pro „Dont Repeat Yourself“, česky „neopakuj se“. Jedná se o princip hojně využívaný v celém průmyslu softwarového vývoje a zkrátka nám říká abychom neopakovali kód, který jsme už jednou napsali. Jedná se tak o princip, který pokud je dobře využíván, může velmi rychle zkrátit celý kód. (16)

Obrázek 3 Dvě varianty tlačítka

Na obrázku lze vidět reálný příklad přímo z podkladové stránky ve formě dvou tlačítek. Tlačítka hned na první pohled vypadají úplně stejně až na jejich barvy a délku. Délka souvisí s obsahem uvnitř tlačítka, což se současným tématem nesouvisí ale barva je již zajímavější. Tlačítka by bylo možné vytvořit následujícím kódem, s tím že btn-orange je tlačítko oranžové a btn-blue modré.

```
.btn-blue {
    display: block;
    width: max-content;
    color: var(--light-blue);
    padding: .5rem 1rem;
    margin: 0 auto;
    border: 1px solid var(--light-blue);
    border-radius: 50px;
}
.btn-orange {
    display: block;
    width: max-content;
    color: var(--orange);
    padding: .5rem 1rem;
    margin: 0 auto;
    border: 1px solid var(--orange);
```

```
    border-radius: 50px;
}
```

Tenhle kód ale není moc efektivní a hodně vlastností se v něm opakuje se stejnými hodnotami. Rozumnější by byla tedy varianty níže.

```
.btn {
    display: block;
    width: max-content;
    color: var(--light-blue);
    padding: .5rem 1rem;
    margin: 0 auto;
    border: 1px solid var(--light-blue);
    border-radius: 50px;
}
.btn--orange {
    color: var(--orange);
    border-color: var(--orange);
}
```

Druhá zmíněná varianta je téměř o polovinu kratší a funkci splňuje stejnou. Rozdíl je v tom, že modré tlačítko dostane třídu .btn a oranžové tlačítko bude mít třídy dvě a to .btn a .btn--orange. První třída .btn určuje základní vlastnosti tlačítka jako je velikost, ohraničení a základní barva, zatímco třída .btn--orange jen upraví barvy.

3.2.2 Konvence pro pojmenovávání tříd BEM

BEM je zkratka pro blok, element, modifikátor. Jedná se o metodiku pro organizaci CSS, ale mnozí z ní čerpají jen její způsob pojmenovávání tříd. Výhoda využívání BEMu spočívá v jednoduchosti a přehlednosti zapisování tříd. Hodí se právě u větších projektů, kdy naše dokumenty obsahují veliké množství tříd, ve kterých se musíme vyznat. (18)

3.2.2.1 Způsob zapisování

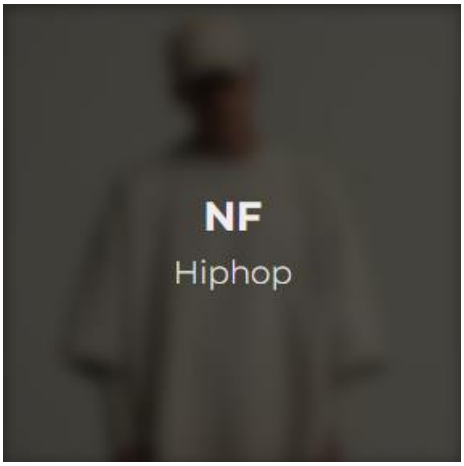
Jak už bylo výše zmíněno BEM se skládá ze tří částí, těmi jsou blok, element a modifikátor. (18)

Blok je samostatný prvek na stránce, který není na ničem závislý a může se tak používat samostatně. Označujeme jej třídou s následující syntaxí: .název-bloku. (18)

Element je součástí uvnitř bloku. Není možné jej použít samostatně, vždy je vázaný na nějaký blok. Zapisuje se následovně: .název-bloku__název-elementu. Odděluje se od bloku, na kterém závisí tedy dvěma podtržítky. (18)

Modifikátor je varianta bloku nebo elementu. Pokud máme tedy nastylované chování nějakého bloku či elementu a chceme pozměnit nějaký detail, jako je třeba barva, použijeme modifikátor. Odděluje se dvěma pomlčkami viz. .název-bloku—žlutý. (18)

3.2.2.2 Ukázka na webové stránce



Obrázek 4 Vzorový obrázek elementu využívajícího BEM

Na obrázku můžeme vidět element ze sekce klienti, který se skládá z obrázku, jména klienta a následně žánru, který klient vytváří. Jedná se o perfektní příklad struktury BEM, kdy je všechen obsah zabalen v tagu <figure> s třídou klient-card od kterého se pak odvíjejí třídy potomků. Struktura karty tedy nakonec vypadá takhle.

```
<figure class="klient-karta">
    <div class="klient-karta__fotka"><img src="url"></div>
    <figcaption class="klient-karta__popis">
        <p class="nadpis">NF</p>
```

```
        <p class="klient-karta__žánr">Hiphop</p>
    </figcaption>
</figure>
```

Struktura kódu je zde v dokumentu mírně zjednodušena z důvodu přehlednosti. Lze si také všimnout, že název interpreta nevyužívá třídu se strukturou, která by navazovala na strukturu BEM téhle karty. Je tak právě z důvodu již zmiňovaného principu DRY, protože název klienta využívá stejné vlastnosti jako kód někde jinde.

3.2.3 Sjednocení stylů ve všech prohlížečích

Každý webový prohlížeč má své výchozí zobrazení elementů na stránce, což nám může nepříjemně zkomplikovat vytváření grafické části naší webové stránky. Z tohoto důvodu se v dnešní době používá buď normalize.css nebo CSS reset. Tyto soubory CSS kódu nám mají zajistit právě to, aby se elementy ve všech prohlížečích chovali stejně. Každý to ale dělá trochu jinak. (19)

3.2.3.1 Normalize.css

Normalize.css sjednocuje styly našich elementů ve všech prohlížečích. Takzvaně mění výchozí chování na jiné, které se bude využívat ve všech prohlížečích a my se tak nemusíme strachovat, že by naše stránka někde vypadala jinak. (20)

3.2.3.2 CSS resety

Na rozdíl od normalize.css, CSS resety prohlížečům říkají, že nás výchozí chování vůbec nezajímá a chceme začít úplně od nuly. To znamená, že náš soubor obsahující CSS reset zaměří všechny elementy na naší stránce a nastaví jim jejich atributy na nejprimitivnější nebo nulové hodnoty. (21)

3.2.3.3 Normalize.css nebo CSS reset

Zatímco normalize.css jde na sjednocení stylů jemnějším přístupem, CSS reset nás posílá úplně na začátek a my tak musíme opravdu nastylovat všechno od nuly. Jaký přístup si chce vývojář zvolit je zcela na něm, ale v dnešní době se stala populárnější volbou normalize.css. (19)

Je však také velmi časté, že vývojáři využívají obojí. Do svého projektu přidají normalize.css a poté si vytvoří malou část kódu, která je zbaví vybraných stylů, které by je mohli při vytváření stránky frustrovat. Mezi často resetované atributy patří například padding, margin, text-decoration či list-style. (19)

3.2.4 Jednotky

> Mnoho vlastností v CSS jako například width, margin, padding a font-size potřebuje za svou hodnotu nějaké číslo s jednotkou a CSS má těchto jednotek spoustu. (21)

3.2.4.1 Absolutní jednotky

Absolutní jednotky jsou pevně dané a jejich hodnota se odvíjí od skutečné fyzické velikosti. Znamená to, že jejich velikost se v průběhu nijak nemění. (21)

Mezi tyto jednotky patří například nejčastěji používané px, neboli pixely, ale také cm, mm, in (palce) a další. (21)

3.2.4.2 Relativní jednotky

Relativní jednotky svoji velikost mění na základě jiné velikosti. Relativní jednotka se může odvíjet například od velikosti rodičovského elementu jako je tak u %, velikosti písma rodičovského elementu jako je u jednotky em nebo od velikosti písma v dokumentu jako u jednotky rem. (21)

3.2.4.3 Kdy používat jaké jednotky

Absolutní jednotky se většinou vyplatí používat pouze v případě, že nějaký element chceme mít vždy v přesně stejné velikosti a nechceme aby byl na něčem závislý. Také se často používají například u borderu, kdy jsou hodnoty často v malých číslech a px jsou tak ideální pro čistý vzhled. (22)

Relativní jednotky ve formě procent jsou pak ideální pro nastavování šířky elementů vůči jejich rodičům. Za pomocí procent tak můžeme například docílit, aby nějaký element vždy zabral přesně polovinu svého rodiče. (22)

Rem a em jednotky se pak často používají například při nastavování paddingu či marginu. Důvodem je automatické zvětšování či zmenšování okrajů elementů na základě textu. Zde už poté záleží jestli chceme, aby okraje závisely na velikosti písma dokumentu nebo konkrétního elementu. (22)

Závěr

Cílem práce bylo čtenáře seznámit s pravidly a konvencemi pro vývoj optimálně strukturovaných webových stránek. K efektivnějšímu naplnění tohoto cíle jsem vytvořil webovou stránku, na které jsem následně ukázal několik příkladů zmíněných pravidel a konvencí. Zároveň jsem vytvořil kodérovo desatero, které má sloužit jako časově přístupnější zdroj nejlepších postupů a může se využít v budoucích třídách i v podobě tištěné formy. K úspěšnému naplnění zmíněných cílů jsem čerpal z mnoha věrohodných zdrojů a shromažďoval ty posstupy, které mi přišly důležité.

Svou práci bych zhodnotil spíše kladně, protože mohu říct, že jsem se naučil a zopakoval si spoustu efektivních technik a postupů psaní HTML a CSS kódu, které věřím, že se mi budou v budoucnu hodit. Největším nedostatkem práce je z mého pohledu malý maximální počet slov, kvůli kterému jsem nebyl o postupech schopen psát detailněji a některé jsem musel kompletně vynechat.

Seznam zkratek a odborných výrazů

HTML

HyperText Markup Language – značkovací jazyk používaný pro tvorbu webových stránek.

CSS

Cascading Style Sheets – jazyk pro stylování obsahu na webových stránkách.

BEM

Blok, element, modifikátor – konvence pro pojmenovávání tříd v CSS

DRY

Don’t repeat yourself – postup psaní kódu, tak abychom neopakovali to co jsme už napsali

Seznam obrázků

Použité zdroje

1. **Mozilla Corporation.** HTML: HyperText Markup Language. *The MDN Web Docs.* [Online] Mozilla Corporation, 13. Sebtember 2022. [Citace: 30. Sebtember 2022.] https://developer.mozilla.org/en-US/docs/Web/HTML.

2. —. CSS Basics. *MDN Web Docs.* [Online] Mozilla Corporation, 19. September 2022. [Citace: 2. December 2022.] https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/CSS_basics.

3. **Geekboots.** Difference between GOOD code and BAD code. *Geekboots - A Place for Programmer & Tech Enthusiast.* [Online] 5. Únor 2018. [Citace: 18. Březen 2023.] https://www.geekboots.com/story/difference-between-good-code-and-bad-code.

4. **Svarrer, Ulrich.** Why Clean and Valid HTML Code is Important for SEO. *Bonzer.* [Online] Bonzer ApS, 28. Prosinec 2022. [Citace: 19. Březen 2023.] https://bonzer.io/html-code-seo/.

5. **James, Oliver.** Semantic HTML. *Internet is Hard.* [Online] internetishard.com. [Citace: 25. Březen 2023.] https://internetingishard.netlify.app/html-and-css/semantic-html/index.html.

6. **Pavlik, Vlado.** Semantic HTML: What It Is and How to Use It Correctly. *Semrush Blog.* [Online] Semrush, 2. Prosinec 2022. [Citace: 25. Březen 2023.] https://www.semrush.com/blog/semantic-html5-guide/.

7. **Brevity Digital Design and Marketing.** SEMANTIC HTML FOR MEANINGFUL WEBPAGES. *Brevity.* [Online] Brevity Digital Design and Marketing, 11. Srpen 2017. [Citace: 25. Březen 2023.] https://seekbrevity.com/semantic-markup-important-web-design/#main-navigation.

8. **Fitzgerald, Anna.** How to Use the Section Element in HTML. *Hubspot.* [Online] HubSpot, Inc., 17. Listopad 2022. [Citace: 25. Březen 2023.] https://blog.hubspot.com/website/section-in-html.

9. **Bhattacharya, Joydeep.** HTML Lang Attribute: What Is It? Why Is It Important? *SEOptimer.* [Online] SEOptimer. [Citace: 17. Duben 2023.] https://www.seoptimer.com/blog/html-lang-attribute/.

10. **CESS.** HTML Best Practices – How to Build a Better HTML-Based Website. *freeCodeCamp.* [Online] Free Code Camp, Inc., 3. Leden 2022. [Citace: 17. Duben 2022.] https://www.freecodecamp.org/news/html-best-practices/.

11. **Silva, Carlos.** What Is an H1 Tag? Why It Matters & Best Practices for SEO. *Semrush Blog.* [Online] Semrush, 17. Listopad 2022. [Citace: 17. Duben 2023.] https://www.semrush.com/blog/h1-tag/.

12. **Olawanle, Joel.** What is Alt Text? Image Alt Text HTML Example. *freeCodeCamp.* [Online] Free Code Camp, Inc., 16. Září 2022. [Citace: 20. Duben 2022.] https://www.freecodecamp.org/news/what-is-alt-text-image-alt-text-html-example/.

13. **SEOMoz, Inc.** Alt Text. *MOZ.* [Online] SEOMoz, Inc. [Citace: 20. Duben 2023.] https://moz.com/learn/seo/alt-text.

14. **Corporation, Mozilla.** <figure>: The Figure with Optional Caption element. *MDN Web Docs.* [Online] Mozilla Corporation, 13. Duben 2023. [Citace: 20. Duben 2023.] https://developer.mozilla.org/en-US/docs/Web/HTML/Element/figcaption.

15. **Kumar, Satyam.** HTML5 - When to use <figure>. *Learning Journal.* [Online] Learning Journal™, 2018. [Citace: 20. Duben 2023.] https://www.learningjournal.guru/article/html5/html5-when-to-use-figure-tag/.

16. **Roberts, Harry.** High-level advice and guidelines for writing sane, manageable, scalable CSS. *CSS Guidelines.* [Online] 1. Září 2022. [Citace: 23. Duben 2023.] https://cssguidelin.es/.

17. **Santhirakumar, Thenusan.** Clean Code: Why Writing Clear, Readable Code Matters. *Medium.* [Online] A Medium Corporation, 8. Únor 2023. [Citace: 23. Duben 2023.] https://enlear.academy/clean-code-why-writing-clear-readable-code-matters-33f96cdb1f32.

18. **Michálek, Martin.** BEM: Pojmenovávací konvence pro třídy v CSS. *Vzhůru Dolů.* [Online] 5. Červen 2017. [Citace: 25. Duben 2023.] https://www.vzhurudolu.cz/prirucka/bem.

19. **Shechter, Elad.** Normalize CSS or CSS Reset?! *Medium.* [Online] A Medium Corporation, 19. Květen 2019. [Citace: 24. Duben 2023.] https://elad.medium.com/normalize-css-or-css-reset-9d75175c5d1e.

20. **Ozanich, Athena.** What is a Normalize CSS File & How Do You Use It? *HubSpot.* [Online] HubSpot, Inc., 1. Srpen 2022. [Citace: 24. Duben 2023.] https://blog.hubspot.com/website/normalize-css.

21. —. What is a CSS Reset File & How Do You Use It? *HubSpot.* [Online] HubSpot, Inc., 28. Červenec 2022. [Citace: 24. Duben 2023.] https://blog.hubspot.com/website/css-reset.

22. **Kis-Herczegh, Petra.** How to Create an SEO- Friendly Website Navigation: Considerations, Navigation Types, and Pro Tips. *Botify.* [Online] BOTIFY SAS. [Citace: 26. Březen 2023.] https://www.botify.com/blog/how-to-create-an-seo-friendly-website-navigation-considerations-navigation-types-and-pro-tips.

A. Seznam přiložených souborů

Na přiloženém datovém nosiči se nacházejí následující soubory a složky:

- **RP2022-23_Lazna-Dominik_Navrh-a-realizace-webovych-stranek.docx** – editovatelná verze dokumentace ročníkové práce
- **RP2022-23_Lazna-Dominik_Navrh-a-realizace-webovych-stranek.pdf** – tisknutelná verze dokumentace ročníkové práce
- **OceanRecords.rar** – názorná webová stránka
- **KodérovoDesatero.pdf** – výtah nejlepších postupů psaní HTML a CSS
- **KodérovoDesateroTisk.pdf** – výtah nejlepších postupů psaní HTML a CSS vhodný k tisku