

**Ročníková práce**

Vývoj vlastního front-end frameworku pro tvorbu webdesignu

Jan, Prokop, C3

Konzultant: Lukáš Zelenka

Školní rok: 2015 / 2016

**Čestné prohlášení**

Prohlašuji, že jsem předloženou ročníkovou práci vypracoval zcela samostatně a uvádím v ní veškeré prameny, které jsem použil. Všechny převzaté informace jsem řádně ocitoval.

V Rakovníku, dne

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (podpis studenta).

# Anotace

Tématem mé ročníkové práce je tvorba CSS Framework, které má za úkol webdesignerům usnadnit jejich práci s navrhnutím struktury HTML dokumentu. Zde budu rozebírat jeho vytvoření, jeho části a jeho využití v praxi.

Tato práce má sloužit jako návod a vysvětlení, co je to Framework, kde a jak ho použít a kdo Framework využije.

V tomto návodu budete postupně seznámeni s veškerými dostupnými funkcemi tohoto frameworku a navíc je vše doplněno o příklady použití tak, aby bylo viditelné, kterými funkcemi Framework disponuje.

# Klíčová slova

Framework, Grid systém, CSS, HTML, Web, Front-end

Obsah

[1 Anotace 3](#_Toc442199578)

[2 Klíčová slova 3](#_Toc442199579)

[3 Úvod 5](#_Toc442199580)

[4 Metodika 5](#_Toc442199581)

[5 Co je to Framework 5](#_Toc442199582)

[5.1 Příklady frameworku 5](#_Toc442199583)

[6 Grid systém 6](#_Toc442199584)

[6.1 Pravidla grid sytému 6](#_Toc442199585)

[6.2 Příklad grid sysému 6](#_Toc442199586)

[6.3 Odsazení sloupců v containeru 7](#_Toc442199587)

[6.4 Zarovnání sloupců v containeru 8](#_Toc442199588)

[7 Formuláře 8](#_Toc442199589)

[7.1 Formulář v řádce 9](#_Toc442199590)

[7.2 Vodorovný formulář 9](#_Toc442199591)

[7.3 Základní textové pole 10](#_Toc442199592)

[7.4 Textarea 10](#_Toc442199593)

[7.5 Zaškrtávací políčka a volba možností 11](#_Toc442199594)

[7.6 Zaškrtávací políčka a volba možností v řádce 12](#_Toc442199595)

[8 Tabulky 12](#_Toc442199596)

[8.1 Ohraničená tabulka 13](#_Toc442199597)

[8.2 Pomocné třídy u tabulek 13](#_Toc442199598)

[9 Možnosti rozšíření a úprav frameworku 14](#_Toc442199599)

[10 Závěr 14](#_Toc442199600)

[11 Seznam použité literatury 14](#_Toc442199601)

# Úvod

V mé ročníkové práci se budu zabývat tvorbou vlastního CSS Framework a následně seznámím uživatele této ročníkové práce se všemi možnostmi a funkcemi tohoto frameworku.

Tato práce má sloužit jako následný manuál a vysvětlení různých funkcí tohoto frameworku, aby měl uživatel používání tohoto frameworku co nejjednodušší a nemusel hledat, co daná třída dělá a jaké jsou její možnosti.

# Metodika

Nejdříve jsem začal s prototypováním celého projektu na papír, abych věděl, co všechno bude tento Framework obsahovat a následně jsem navrhnutý prototyp převedl do elektronické pdooby a začal s tvorbou vlastního kódu CSS pomocí programu PSPad, kde jsem začal sepisovat jednotlivé funkce frameworku. Začal jsem s grid systémem. V grid systému jsem nejdříve vytvořil fluidní container a poté klasický container následně jsem zadal přesnou šířku každému sloupci a nastavil další funkce, jako je odsazení a také posunutí do stran.

Po vytvoření celého grid systému jsem začal s tvorbou formulářů. Jednotlivým položkám formuláře byly nastaveny výchozí hodnoty, které si dále uživatel může upravit. Formuláře jsem koncipoval tak, aby je bylo možné aplikovat kdekoliv na stránce a jejich strukturu ovlivňovat za pomocí grid systému.

Následně jsem nastavil výchozí hodnoty klasickým tabulkám a poté přidal další třídu, aby uživatel mohl používat i ohraničenou tabulku a využít tak všech funkcí, kterými Framework disponuje.

Po dokončení tabulek jsem přidal zakulacení u obrázků a tlačítek.

Pro kód jsem použil své znalosti z předmětu web design a také jsem se inspiroval v již některých vytvořených frameworcích.

Tento postup jsem si vybral, protože pro mě bylo nejjednodušší vytvořit kód a poté pro něj napsat manuál a samotnou ročníkovou práci.

Data a informace pro vytvoření celé práce jsem hledal na různých webových stránkách o frameworcích.

# Co je to Framework

Framework neboli aplikační rámec je softwarová struktura a šablona sloužící k usnadnění práce uživateli a zjednodušení práce při různých projektech.

Cílem frameworku je co nejvíce usnadnit uživateli svou práci, aby se mohl soustředit pouze na svou práci.

Někdy se můžeme setkat s názory, že použití frameworku je špatné a že často uživatel musí daný Framework důkladně nastudovat, tak to ale není. Při jednorázovém použití tomu tak může být, ale když daný Framework používáte opakovaně, je to opravdu veliké usnadnění.

Velkou výhodou je, že dostanete již předformátovanou funkční šablonu, kterou můžete dále upravovat.

Další z výhod je, že Framework můžeme použít prakticky na každý projekt a vždy si ho podle vlastní libosti upravit.

Naopak nevýhodou je, že Framework musíte nastudovat a přečíst si jeho manuál, nebo vždy vyhledat jeho funkce.

## Příklady frameworku

**Jquery:** „*je javascriptová knihovna s širokou podporou prohlížečů, která klade důraz na interakci mezi JavaScriptem a HTML. Byla vydána Johnem Resigem v lednu 2006 na newyorském BarCampu*.“

**Bootstrap:** *„je volně stažitelná sada nástrojů pro tvorbu webu a webových aplikací. Obsahuje návrhářské šablony založené na HTML a CSS, sloužící pro úpravu typografie, formulářů, tlačítek, navigace a dalších komponent rozhraní, stejně jako další volitelná rozšíření JavaScriptu. Je to nejúspěšnější projekt na GitHub s 65000+ hvězdičkami.“*

**Foundation Framework:** *„ je responsivní front-end Framework pro tvorbu webu a webových apliakcí. Obsahuje návrhářské šablony založené na HTML a CSS sloužící pro úpravu typografie, formulářů, tlačítek, navigace a dalších komponent rozhraní. Bylo vydáno v září 2011 firmou ZURB.*

**Yahoo Pure CSS:** *je responsivní front-end Framework pro tvorbu webu a webových apliakcí.*

A spousta dalších.

# Grid systém

Grid systém se využívá při tvorbě rozložení webových stránek pomocí sloupců a řádků. Pomáhá nám určit strukturu a rozložení webové stránky.

Řádky a sloupce mohou být umístěny do container-fluid (využívá plnou šíři stránky), nebo do container(využívá pevně danou šířku ze stránky).

Obsah dokumentu by měl být vždy umístěn správně do sloupců, aby se zachovala struktura webu a obsah dokumentu nebyl rozházen volně po stránce.

Grid systém tvoří dvanáct sloupců a je možné je do sebe různě vnořovat.

Nejužší sloupec je .col-1 naopak .col.12 je přes celou stránku

## Pravidla grid sytému

*„Obsah by měl být umístěn uvnitř sloupce, a sloupce mohou být přímí potomci řádků.“*

*„Řádky musí být umístěny buď do .container-fluid (plná šířka stránky), nebo do .container(pevně daná šířka stránky), pro správné zarovnání a odsazení.“*

*„Předdefinované třídy, jako je .row a například .col-4 jsou k dispozici pro rychlé vytváření řádků a sloupců.“*

*„Sloupce jsou vytvořeny zadáním přesného počtu z 12 dostupných sloupců. Například, tři stejné sloupce by používaly třídu .col-4.“*

## Příklad grid systému

Použitím třídy .col-1 rozdělíme šířku stránky na dvanáct stejných sloupců, přičemž musíme dvanáctkrát vytvořit element s touto třídou.

**HTML:**

<div class=“row“>

<div class=“col-1“>.col-1</div>

<div class=“col-1“>.col-1</div>

<div class=“col-1“>.col-1</div>

<div class=“col-1“>.col-1</div>

<div class=“col-1“>.col-1</div>

<div class=“col-1“>.col-1</div>

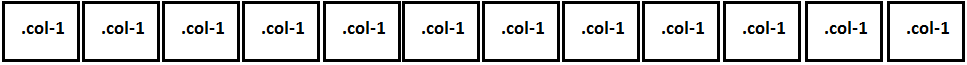
<div class=“col-1“>.col-1</div>

<div class=“col-1“>.col-1</div>

<div class=“col-1“>.col-1</div>

<div class=“col-1“>.col-1</div>

<div class=“col-1“>.col-1</div>

<div class=“col-1“>.col-1</div>

Pokud chceme stránku rozdělit pouze na dva sloupce různé šířky, použijeme k tomu jen dvě třídy a to : .col-9 + .col-3.

**HTML:**

<div class="container">

<div class="col-9">

<h1>.col-9</h1>

</div>

<div class="col-3">

<h1>.col-3</h1>

</div>

</div>



Kdybychom chtěli stránku rozdělit na poloviny, využijeme k tomu 2 sloupce stejné šíře a to .col-6 a .col-6.

**HTML:**

<div class=“row“>

<div class=“col-6“>.col-6</div>

<div class=“col-6>.col-6</div>

</div>



## Odsazení sloupců v containeru

Pro odsazení sloupce z leva aplikujeme navíc speciální třídu .col-offset-x. X v tomto případě interpretuje počet sloupců, o které dotyčný element chceme odsadit.

**Příklad**: .col-offset-2 .col-6

Vytvoří sloupec o šíři 6, který bude z levé strany odsazen o šířku 2 sloupců

**HTML:**

<div class="container">

<div class="row">

<div class="col-3">.col-3</div>

<div class="col-3 col-offset-3">.col-3 col-offset-3</div>

</div>

</div>

****

Pro vycentrování jednoho sloupce můžeme využít speciální třídy .col-offset-x a to tak, že hodnota X bude polovina z (12 – (šíře sloupce, který chceme zastředit))/2. Tento vzorec je platný pouze za předpokladu, že šíře centrovaného sloupce je sudé číslo.

Pro liché šíře sloupců neexistuje jednoduché řešení pro vycentrování.

**Příklad:** Chceme-li vycentrovat sloupec, který má třídu .col-8. Podle vzorce spočítáme odsazení(offset):offset = (12 – 8) / 2 = 2. Proto v případě centrování .col-8 musíme přidat třídu .col-offset-2.

**HTML:**

<div class="container">

<div class="row">

<div class="col-6 col-offset-3">.col-6 .col-offset-3</div>

</div>

</div>

## Zarovnání sloupců v containeru

Sloupce v containeru můžeme zarovnat dvěma způsoby, jedním z nich je, že sloupce posuneme do prava a k tomu použijeme .col-\*-pull-right. A druhý způsob je, že zarovnáme sloupce do leva a k tomu použijeme .col-\*-pull-left.

**HTML:**

<div class="container">

<div class="row">

<div class="col-9 pull-left">.col-9 pull-left</div>

<div class="col-3 pull-right">.col-3 pull-right</div>

</div>

</div>****

# Formuláře

Formuláře a jejich základní funkce v mém frameworku jsou nastaveny na celou šířku sloupce a samy se upravují podle šířky monitoru.

Ve formulářích nalezneme základní textová pole, která mohou mít různé typy, např.: text, heslo, textarea a tak dále..

**Příklad formuláře:**

**HTML:**

<form action="submit.php" method="post">

<input required="required" pattern=".{1,} .{1,}" type="text" name="name" placeholder="Jméno Příjmení"><br>

<input required="required" type="email" name="email" placeholder="vas@email.cz"><br>

<input required="required" pattern=".{4,32}" type="text" name="subject" placeholder="Předmět"><br>

<textarea name="zprava" placeholder="Vaše zpráva ..."></textarea><br>

<input type="submit" class="odeslat pull-right" value="Odeslat">

</form>

## Formulář v řádce

Přidáním třídy .form-inline do našeho formuláře, zajistíme, že položky formuláře budou v jedné rovině.

Klasický formulář má na svých položkách nastavenou plnou šířku 100%, ale inline formuláře mají nastaveno auto, aby položky ve formuláři mohly být vedle sebe.

**HTML:**

<form class="form-inline">

<div>

<label for="Jméno">Jméno</label>

<input type="text" class="form-control" id="Jmeno2" placeholder="Vaše Jméno">

</div>

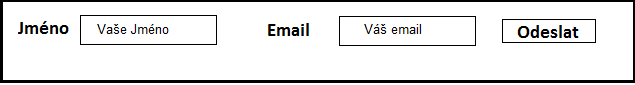
<div>

<label for="Email">Email</label>

<input type="email" class="form-control" id="exampleInputEmail2" placeholder="Váš email">

</div>

<button type="submit" class="btn btn-default">Odeslat</button>

****</form>

## Vodorovný formulář

Použitím předdefinovaného grid systému a použitím předdefinované třídy .form-horizontal, docílíme toho, že formulář bude vodorovný. Použitím třídy .form-group zajistíme, že každá položka je na nové řádce a nemusíme používat třídu .row.

**HTML:**

<form class="form-horizontal">

<div class="form-group">

<label for="inputEmail3" class="col-sm-2 control-label">Email</label>

<div class="col-10">

<input type="email" class="form-control" id="Zadatemail" placeholder="Email">

</div>

</div>

<div class="form-group">

<label for="zadej heslo">Password</label>

<div class="col-sm-10">

<input type="password" class="form-control" id="zadatheslo" placeholder="Heslo">

</div>

</div>

<div class="form-group">

<div>

<div class="checkbox">

<label>

<input type="checkbox"> Pamatovat si mě

</label>

</div>

</div>

</div>

<div class="form-group">

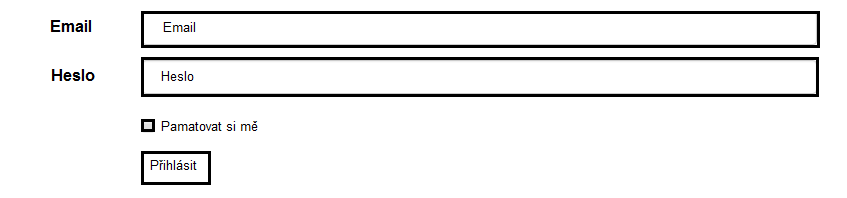
<div>

<button type="submit" class="btn btn-default">Přihlásit</button>

</div>

</div>

</form>



## Základní textové pole

<input type="text" class="form-control" placeholder="Zadejte text">

## Textarea

Položka formuláře, která umožňuje psát více textu na více řádků. Počet řádků se dá manuálně upravit podle potřeby uživatele.

<textarea class="form-control" rows="5"></textarea>

## Zaškrtávací políčka a volba možností

Zaškrtávací políčka slouží pro výběr z jedné nebo více možností. Pokud je u některých zaškrtávacích políček možnost zakázána je nutné přidat třídu .disabled.

**HTML:**

<div class="checkbox">

<label>

<input type="checkbox" value="">

První možnost

</label>

</div>

<div class="checkbox disabled">

<label>

<input type="checkbox" value="" disabled>

Druhá možnost je zablokována

</label>

</div>

<div class="radio">

<label>

<input type="radio" name="volbamoznosti" id="moznost1" value="moznost1" checked>

První možnost

</label>

</div>

<div class="radio">

<label>

<input type="radio" name="volbamoznosti" id="moznost2" value="moznost2">

Druhá možnost

</label>

</div>

<div class="radio disabled">

<label>

<input type="radio" name="volbamoznosti" id="moznost3" value="moznost3" disabled>

Třetí možnost je zablokována

</label>

</div>

## Zaškrtávací políčka a volba možností v řádce

Použitím třídy .checkbox-inline nebo .radio-inline zajistíme, aby zaškrtávací políčka a volba možností byla vedle sebe v řádce.

<label class="checkbox-inline">

<input type="checkbox" id="pole1" value="moznost1"> 1

</label>

<label class="checkbox-inline">

<input type="checkbox" id="pole2" value="moznost2"> 2

</label>

<label class="checkbox-inline">

<input type="checkbox" id="pole3" value="moznost3"> 3

</label>

<label class="radio-inline">

<input type="radio" name="volbavradce" id="volbavradce" value="moznost1"> 1

</label>

<label class="radio-inline">

<input type="radio" name="volbavradce" id="volbavradce2" value="moznost2"> 2

</label>

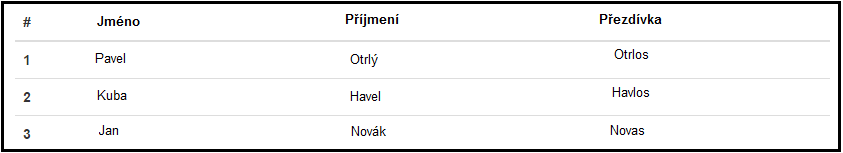
<label class="radio-inline">

<input type="radio" name="volbavradce" id="volbavradce3" value="moznost3"> 3

</label>

# Tabulky

Pro vytvoření tabulky v mém frameworku použijeme klasický tag <table>.

**Příklad klasické tabulky:**

## Ohraničená tabulka

Přidáním třídy .table-bordered docílíme toho, že se na naší tabulky přidá ohraničení ze všech stran.

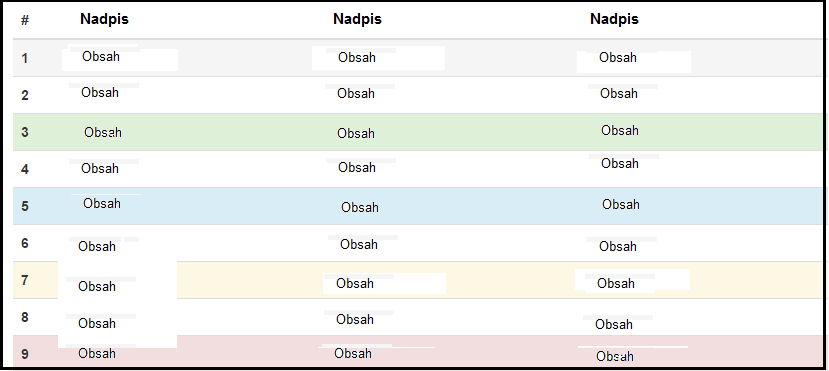
## Pomocné třídy u tabulek

**.active** – použije barvu na konkrétní řádek nebo na konkrétní buňku.

**.succes** – označí úspěšně splněnou buňku nebo řádek.

**.info** – označí nějakou informaci v buňce nebo řádku.

**.warning** – označí buňku nebo řádek s varováním, že je potřeba této buňce nebo řádku věnovat pozornost.

**.danger –** označí buňku nebo řádek, kde by mohlo vzniknout potencionální nebezpečí.

**HTML:**

*<!-- pro řádky -->*

<tr class="active">...</tr>

<tr class="success">...</tr>

<tr class="warning">...</tr>

<tr class="danger">...</tr>

<tr class="info">...</tr>

*<!-- Pro buňky (`td` nebo `th`) -->*

<tr>

<td class="active">...</td>

<td class="success">...</td>

<td class="warning">...</td>

<td class="danger">...</td>

<td class="info">...</td>

</tr>

# Možnosti rozšíření a úprav frameworku

V HTML dokumentu je možnost připojit další stylopis a při dodržení CSS specifity v něm můžeme přetížit selektory frameworku a podle vlastní libosti můžeme dále celý dokument stylovat a upravovat.

Jelikož Framework neslouží ke stylování stránek, ale pouze jako pomocník k usnadnění pozicování elementů na stránce pomocí tříd, tak je doporučeno vytvořit vlastní stylopis a v něm následně upravovat svou stránku.

# Závěr

Při této práci jsem dospěl k tomu, že Framework je opravdu velmi užitečná věc a velké usnadnění veškeré práce a že při použití frameworku se uživatel může soustředit jen na svou práci.

Tato práce mi přinesla mnoho zkušeností v oblasti webdesignu a také v oblasti programování.

Můj vytvořený Framework rozhodně najde uplatnění v menších například školních projektech, ale je možno ho využít i v nějakých menších projektech. Můj Framework bude volně k dispozici a bude volně šiřitelný.

# Seznam použité literatury