Úvod

Formulář je prvek stránky, který je určen pro odeslání dat ze stránky pryč. Většinou se posílají na nějaký server, který požadavek zpracuje a vrátí nějaký výsledek. Použití je tedy např.:

- kontaktní formuláře
- objednávky
- přihlášení
- registrace
- a další

Naše cíle s formuláři budou ale skromnější. Cílem bude především vytvářet hezké formuláře a později, kdy se naučíme s JavaScriptem si tyto data můžeme i předávat a číst je. Tedy budou spíše cílit na uživatele a tak, aby je uměl používat. Neukážeme si ani pokročilejší validace formulářů (např. hesla), ale to se lze jednoduše naučit ve VOPech Webového frontendu a backendu.

Nejhlavnější prvek formulářů je samotný formulář. V HTML jej označujeme pomocí tagu `form` a má dva atributy:

- `action`: definuje cestu (URL), kam se data odešlou
- `method`: definuje způsob odeslání dat - dvě možnost `get` a `post`

Při odesílání formuláře se odešlou právě data, které jsou v daném formuláři. Tyto data nazýváme prvky formuláře. Formulář definujeme např. následovně:

```
<form action="zpracuj.php" method="get">


</form>
```

Prvky formuláře

Prvky formuláře znáte např. z PVA (C# forms či WPF), např. tedy obecnými názvy textbox, selectbox, radio a další. My si ukážeme pár z nich, které se hodí do většiny formulářů. Nějaké tyto prvky mají validaci na straně klienta a mohou např. omezit, že se nepošle daný prvek nevyplněný. Většina prvků používá tag `input` společně s atributem `type`, ale existují i výjimky.

Na spoustě vstupů lze nastavovat různé atributy. Některé atributy jsou společné, např.:

- `placeholder`: nastaví text, který se bude zobrazovat, pokud uvnitř prvku nebude žádná hodnota, např. `tady vyplňte heslo` atp.
- `value`: nastaví základní hodnotu, která bude na prvku ve formuláři, když se vytvoří.

Textový vstup (`type="text"`)

Vytvoří obecný textový vstup na stránce. Může obsahovat prakticky cokoliv, ale používá se i na číselné vstupy. Nemá žádnou validaci. Všechen obsah je taktéž na jedné řádce.

```
<form action="akce.php">
    <input type="text">
    <input type="text">
    <input type="text">
</form>
```

Tlačítko (`type="button"`)

Slouží pro místo, kde uživatel může aktivovat různé akce ve formuláři. Obecně nemá žádný význam. Atributem `value` určujeme jméno, které se vykreslí.

```
<form action="akce.php">
    <input type="button">
    <input type="button" value="Vymňoukovat">
    <input type="button" value="Koupit!">
</form>
```

Odesílač (`type="submit"`)

Vytvoří tlačítko pomocí kterého můžeme odeslat formulář dle nastavení tagu `form`. Stejné atributy jako obyčejné tlačítko.

```
<form action="akce.php">
    <input type="submit">
    <input type="submit" value="Vymňoukovat">
    <input type="submit" value="Koupit!">
</form>
```

E-mail (`type="email"`)

Textové pole, které validuje při odeslání, zda daný e-vstup je platná e-mailová adresa.

```
<form action="akce.php">
    <input type="email">
    <input type="email" placeholder="Váš e-mail">
    <input type="email" placeholder="Váš e-mail" value="@neplatnost.cz" >
```

```
    <input type="submit" value="Odeslat">
</form>
```

Heslo (`type="password"`)

Textové pole ve kterém se obsah ale nezobrazuje (a ani nejde např. zkopírovat) a místo něho se zobrazují zástupné znaky.

```
<form action="akce.php">
    <input type="password">
    <input type="password" placeholder="Vaše heslo od Roblox účtu">
    <input type="password" value="12456sspsag">
</form>
```

Datum (`type="datetime/datetime-local"`)

Textové pole k zadávání data. Dle prohlížeče se otevře vybírač datumu. Pomocí atributu min a max můžeme určit minimální a maximální datum.

```
<form action="akce.php">
    <input type="date">
    <input type="date" min="1900-01-01" max="1920-01-01">
    <input type="date" min="2022-02-01">

    <input type="submit" value="Odeslat">
</form>
```

Soubor (`type="file"`)

Slouží k výběru soubor/ů z lokálního počítače. Můžeme určit požadovaný počet souborů (`multiply`) a seznam akceptovaných přípon či MIME typů (`accept`).

```
<form action="akce.php">
    <input type="file">
    <input type="file" multiple>
    <input type="file" multiple accept=".jpg,.svg,.png,.webp">
</form>
```

Zaškrtávací políčko (`type="checkbox"`)

Prvek který lze zaškrtnout (označit) a zrušit mu zaškrtnutí (odznačit). Pomocí atributu `checked` ovládáme, zda je v základu zaškrnut.

```
<form action="akce.php">
    <input type="checkbox">
    <input type="checkbox" checked>
</form>
```

Označovací políčko (`type="radio"`)

Jedná se o omezené zaškrtávací políčko, protože se seskupuje pro různé názvy - v dané kategorii může být označený pouze jeden prvek. Prvek taktéž nelze odznačit. V níže uvedeném příkladě může být označena jenom jedna kočka.

```
<form action="akce.php">
    <input type="radio" name="cat" value="Yoko">
    <input type="radio" name="cat" value="Bambi">
    <input type="radio" name="dog" value="Yenda">
</form>
```

Dlouhý text (`textarea`)

Jedná se o prvek se stejným účelem, jako text, ale dovoluje zvětšovat prvek (v základním designu) a psát více řádek (shift enter)

```
<form action="akce.php">
    <textarea></textarea>
    <textarea>:)</textarea>
    <textarea placeholder="Sem vyplňte všechny soukromé údaje"></textarea>
</form>
```

Výběr z možností (`select`)

Dovoluje možnost vybírat jednu či více (atribut `multiple`) možnosti ze seznamu. Možnosti definujeme uvnitř prvku a udělujeme jim do atributu value hodnoty, které se případně odešlou na server.

```
<label for="grade">Známka</label>
<select name="grade" id="grade">
    <option value="1">Výborná</option>
    <option value="2">Chvalitebná</option>
    <option value="3">Dobrá</option>
```

```
    <option value="4">Dostačující</option>
    <option value="5">Nedostačující</option>
</select>

<label for="dogs">Oblíbení pejsci</label>
<select name="dogs" multiple id="dogs">
    <option value="hafan-1">Hafan 1</option>
    <option value="hafan-2">Hafan 2</option>
    <option value="rosta-kekw">Rosťa určitě ne</option>
</select>
```

Mezi další prvky patří např.:

- color
- month
- number
- range
- reset
- search
- tel
- url
- week
- hidden

Popisky

Veškeré prvky ve formuláři musí být popsané. K tomu slouží popisky, které se tvoří tagem label. Každý popisek by měl být určen jednomu prvku (jeho identifikátoru). Některé popisky jsou velmi kritické pro formulář a jedná se např. o checkboxy a radio.

Komu popisek patří se určuje pomocí atributu `for`, ten ukazuje na nějaký identifikátor na stránce. Například tedy:

```
<label>Popis ničeho</label>

<label for="name">Jméno</label>
<input type="text" id="name">
```

Pro checkboxy to funguje např. následovně:

```
<div>
    <input type="radio" name="cat" value="Yoko" id="yoko">
    <label for="yoko">Yoko</label>
</div>
<div>
    <input type="radio" name="cat" value="Bambi" id="bambi">
    <label for="bambi">Bambi</label>
</div>
<div>
    <input type="radio" name="dog" value="Yenda" id="yenda">
    <label for="yenda">Yenda</label>
</div>
```

Popisky mají i vedlejší účinek a to, že když na něj klikneme, tak se daný formulářový prvek zaktivní. Pro vstupní prvky se nám tam přesune kurzor. Pro výběrové prvky se daná možnost označí.

Stylování formulářů

Stylování prvků formuláře je jednoduché z pohledu nastavování vlastností - většinu z nich uhodnete již podle toho, jak prvek vypadá. V základu je label inline prvek a např. input či textarea blokový. Často musíme jednotlivé popisky a vstupy seskupovat do jednotlivých rodičů, abychom je mohli stylovat. Používáme tedy často flexbox a grid.

Vlastnost `resize`

Jedná se o vlastnost, která dovoluje uživateli jednotlivý prvek zvětšovat či zmenšovat. Můžeme omezit v jakých dimenzích půjde zvětšování omezit. Často se používá s vlastnostmi pro nastavení minimální a maximální šířky, aby se omezilo jejich zvětšování.

```
.resize {
    resize: both;
}

.resize.vertical {
    resize: vertical;
}

.resize.horizontal {
    resize: horizontal;
```

```
}
.resize.disabled {
    resize: none;
}
```

Vlastnost `backdrop-filter`

Jedná se o stejnou vlastnost, jako `filter`, ale jen, že se nastavuje na pozadí daného prvku. Např.:

```
.bg-blur { /* <---- takhle třída je špatná, proč? */
    backdrop-filter: blur(15px);
}
```

Předávání dat

Data se z formulářů předávají právě dvěma způsoby - pomocí get nebo post. Tyto názvy reflektují metody uvnitř protokolů HTTP/S.

GET

Způsob GET funguje na bázi toho, že jednotlivé prvky formuláře jsou spojeny jako parametry URL a jsou poslány na danou stránku. Tedy v URL může uživatel vidět zadané údaje a tam již jde vidět problém této metody. Uživatel to může jednoduše vidět a obecně je to náchylnější na podvody (omylem zkopíruje url s heslem, ...). Tady hned vidíme první problém. Druhým problémem je to, že nepodporuje různé typy vstupů, nejdůležitější jsou soubory, které v této metodě posílat nelze.

Parametry se tvoří vždy pomocí dvojce jména (atribut `name` na prvku) a její hodnoty. Je důležité si uvědomit, že URL má nějaký limit a nemusí se tam všechny parametry vejít.

POST

Data se v podobném formátu jako v parametrech ukládají do těla požadavku a uživatel je nemůže zkopírovat (lze je vidět v DevTools). Data ale musí být komplikovanější a i binární - lze tedy posílat i soubory.

Pozicování

Umístění na webové stránce slouží k uložení prvků stránky na jinou pozici než mu určila stránka nebo jiné rozložení. Všechny různé umístění se ovládají pomocí vlastností `top`, `bottom`, `left` a `right`. Tyto vlastnosti pohybují prvkem z dané strany.

Pozicování nastavujeme pomocí vlastnosti `position` s hodnotami `static`, `relative`, `absolute`, `fixed`, `sticky`.

Nyní si ukážeme veškeré základní umístění:

Statické pozicování je to klasické pozicování které je základně nastaveno. Je to jediné pozicování neumí pracovat s vlastnostmi `top`, `bottom`, `left` a `right`. Prvek je umístěn na jeho normální pozici, které mu určí stránka a vlastnosti (display, flex, grid, margin, ...)

Relativní pozicování je velmi podobné tomu statickému a v funguje na něm pozicování pomocí vlastnosti `top`, `bottom`, `left` a `right`. Prvek se umístí podle pozicování statického a výše zmíněnými vlastnostmi jej odsadíme od této pozice.

Fixované pozicování se váže na velikost obrazovky a při scrollování na webové stránce prvek zůstane umístěn dle obrazovky. Toto pozicování také způsobí že prvek se umístí na jinou úroveň stránky a tím pádem neovlivňuje žádné další prvky.

Absolutní pozicování způsobí že se to zachytí na nejbližší nadřazený prvek na kterému je nastaveno nějaké pozicování tedy kromě toho statického. Pozicování znova způsobí, že neovlivňuje žádné další prvky.

Lepící pozicování je vlastně přechod dvěma stavy pozicování a to relativním a fixovaným. Dokud stránka nesplní odsazení daného lepícího prvků bude se prvek chovat jako relativním pozicováním. Až prvek splní odsazené trošku fixované pozicování a tady prvek bude zachycen dle velikosti obrazovky.

Na stránce taky můžeme změnit vrstvu prvku ručně a to pomocí vlastnosti `z-index`. Vlastnosti předáváme hodnotu číselnou která určuje v jaké vzdálenosti prvek je. Čím větší číslo tím výše bude prvek na ose Z umístěn.

Frameworky

Framework je předpřipravená knihovna či styly, které ulehčují práci na nějaké aplikaci (webové stránce či čímkoliv jiným). Většinou se skládá z více částí, které spojujeme dohromady pro to, abychom vytvořili funkční celek.

Frameworky jsou ve WBA velmi neomezené téma, jedná se tedy např. o náš reset, fonty či ikonky z fontawesome. V jiných oborech se setkáte s více definovaným pojmem.

Výhody frameworků jsou velmi zřetelné:

- rychlejší práce
- nemusíme něco pást vícekrát
- věci pouze skládáme a tedy nemusíme mít takovou znalost

Nevýhody:

- máme vše hotové a těžko se věci modifikují
- stránky vypadají stejně
- licenční problémy

Druhý bod nevýhod je velmi klíčový. Proto se často setkáte s tím, že se frameworky designové používají na prototypy či neprezentační webové stránky, např. panely či jiné, kde na designu tolik nezáleží.

Mezi nejznámější frameworky patří:

- https://bulma.io
- https://material.io/design
- https://tailwindcss.com
- https://get.foundation
- https://semantic-ui.com
- https://getbootstrap.com

Bootstrap

Bootstrap je jedním z prvních nejpoužívanějších frameworků. Důvod je proto jednoduchý - dříve neexistoval flex, grid či jiné věci a proto bylo tvoření responzivních stránek velmi těžké. Používali se různé okliky a hacky, které bylo těžké sledovat. Nejpoužívanější částí bootstrapu byl jeho grid systém, který umožnoval stránky velmi hezky tvořit.

Cílem bootstrapu je spíše Vás seznámit s tím, jak funguje dokumentace, a že spoustu věcí uděláte přímo z jeho dokumentace. Dokumentaci naleznete ZDE.

Responzivita je v Bootstrapu vytvořena pomocí breakpointů. Breakpoint označuje místo zlomu, kde se očekává, že se začne používat jiné zařízení. Bootstrap má šest breakpointů:

Název	Zápis	Velikosti
Extra small		≤ 576px
Small	sm	≥ 576px
Medium	md	≥ 768px
Large	lg	≥ 992px
Extra large	xl	≥ 1200px
Extra extra large	xxl	≥ 1400px

Tyto názvy můžeme používat skoro u každé komponenty (zejména grid systému, kontejnerech a dalších) pro určení zalomení, např. pro grid sloupec:

`col-2 col-xl-8`, na zařízení nejmenším do lg (tj. do 991px) bude mít velikost 2/12 místa, na zařízeních větších poté 8/12 místa

`col-12 col-md-6 col-lg-4 col-xxl-3`, na nejmenším a sm 12/12, na md 6/12, na lg a xl 4/12, na xxl a větších 3/12