

Formuláře

Formuláře

`<form>`

`action`

`target`

`method`

`<input>`

`type="..."`

`text`

`checkbox`

`radio`

`submit`

`password`

`number`

`date`

`email`

`tel`

`hidden`

`name="..."`

`value="..."`

`placeholder="..."`

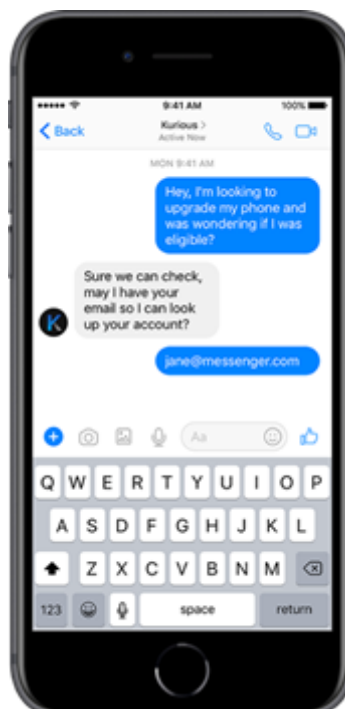
`required`

`pattern="..."`

`<select>`

`<fieldset>` a `<legend>`

Vyplňování formulářů je nedílnou součástí našich digitálních životů, ať už jde např. o hledání na webu, přihlašování na e-mail, nebo vyplňování čtyřstránkové registrace na e-shopu. Někdy dokonce ani jako formuláře nevypadají:



Díky formulářům se **statický** web stává **dynamickým** (interaktivním).

Samotné formuláře k tomu ale nestačí. Na serveru musí být "něco", kdo námi odeslaná data zpracuje. Tématika serverových skriptů vás čeká až ve třetím ročníku. Přesto, či spíše právě proto, je nezbytné umět je používat a rozumět jejich stavbě. Proč? Dynamický web je vystavěný minimálně ze dvou komponent:

1. HTML (formuláře)
2. PHP či jiný jazyk (zpracování dat z nich) - to jsou ty serverové skripty.

Jinými slovy, bez HTML formulářů by dynamický web tak, jak jej známe nemohl existovat.

<form>

Povinným formulářovým elementem je

```
<form></form>
```

Zastřešuje všechny ostatní formulářové elementy. Jeho použití je snadné:

```
<body>
  <form>
    Zde budou formulářové prvky:
    doprovodný text, pole, tlačítka atd.
  </form>
</body>
```

Atributy elementu `form`

action

obsahuje URL (v tomto případě relativní) na skript, kterému je potom formulář poslán

target

určuje cílový rám, ve kterém se budou výsledky zpracovávat. Podobnost s `target` u odkazů není náhodná. Např. hodnota `target="_blank"` posílá výsledek do nového okna.

method

určuje způsob, kterým budou data odesílána. K dispozici jsou hodnoty GET a POST.

- `GET` je výchozí metoda (nemusí se zadávat). Data odeslaná metodou GET jsou součástí URL za otazníkem a jsou vidět. Používá se většinou u krátkých formulářů.
- `POST` je metoda vhodná pro rozsáhlejší vstupy. Data nepřidává do URL (takže nejsou vidět), ale odesílá je jako samostatný HTTP objekt.

Příklad:

```
<body>
  <form action="skript.php" method="get">
    Zde budou formulářové prvky:
    doprovodný text, pole, tlačítka atd.
  </form>
</body>
```

<input>

Input znamená vstup. V žebříčku TOP 10 nejdůležitějších elementů zaujímá input hned druhé místo. Tím prvním je ... (doplňte).

Element input je **nepárový** a **řádkový**. Vkládá se dovnitř elementu `<form>`. Jeho základní (minimální) syntaxe je:

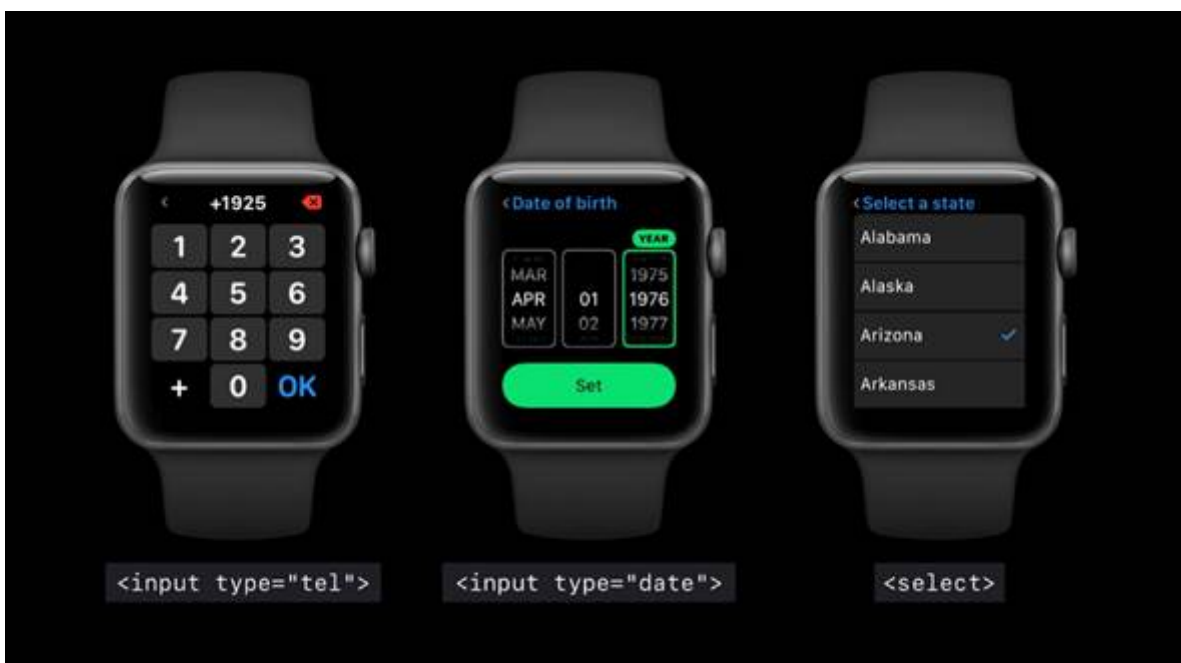
```
<input type="typ_inputu">
```

Atributy elementu `input`

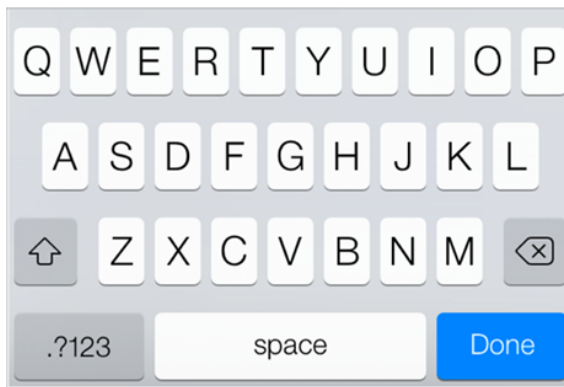
type="..."

Povinný atribut, kterých je přes 20 typů: textové pole, zaškrtnuté pole, přepínač, pole pro zadání hesla, tlačítko atd. atd. Specifikace HTML5 přidala další užitečné typy: pole pro email, URL, číslo, datum... Proč?

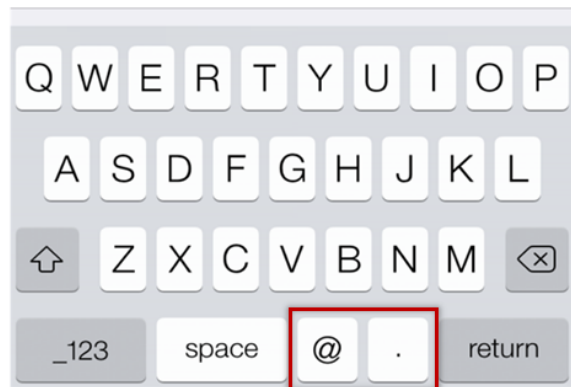
1. Některé z nich v prohlížeči aktivují validátor zadaných hodnot ještě před odesláním.
2. Na mobilních zařízeních vyvolají příslušný typ klávesnice pro pohodlné zadávání hodnot (např. `email`, `url`, `number`).
3. Kompletní výčet https://www.w3schools.com/tags/att_input_type.asp
4. 💡 Testovací web formulářových polí a atributů <https://inputtypes.com>



Formulářové prvky na watchOS 5. Zdroj: Apple



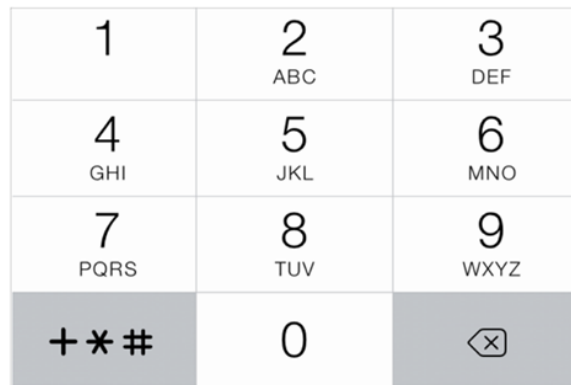
`<input type="text">`



`<input type="email">`



`<input type="url">`



`<input type="tel">`

Různé klávesnice zobrazené podle typu inputu

Zde je přehled 10 hlavních inputů:

text

Vstupní textové pole. Výchozí šířka se změní atributem `size`, např. `size="20"`.

```
<input type="text">
```

Karel

checkbox

Zaškrťovací pole (checkbox). Pokud je nezaškrtnutý, na server se nic neodesílá, jinak se odešle hodnota v atributu `value`. Hodnota `checked` označuje přednastavené zatržení políčka.

```
<input type="checkbox" value="ano"> Text A<br>  
<input type="checkbox" value="ano" checked> Text B
```

Text A

Text B

radio

Přepínač. Každý přepínač je jeden input. Vždy je aktivní pouze jedna volba se **stejným** `name`.
Nepovinná hodnota `checked` předvolí hodnotu:

```
<input type="radio" name="a" value="horní"> Horní<br />
<input type="radio" name="a" value="dolní" checked> Dolní
```

- Horní
- Dolní

submit

Odesílací tlačítko. Odešle všechny hodnoty uvnitř prvku. Hodnota `value` vypíše text na tlačítko.

```
<input type="submit" value="Registrovat">
```

Registrovat

password

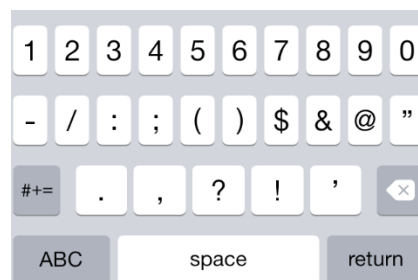
Vstupní pole pro heslo. Pouze skryje hodnoty, jinak nic.

```
<input type="password">
```

.....

number

Pole pro zadání čísla. Jiná než číselná hodnota nejde zadat. Přidáním nepovinného atributu `inputmode="numeric"` zobrazují mobily pouze telefonní klávesnici. Bez něj numerickou:

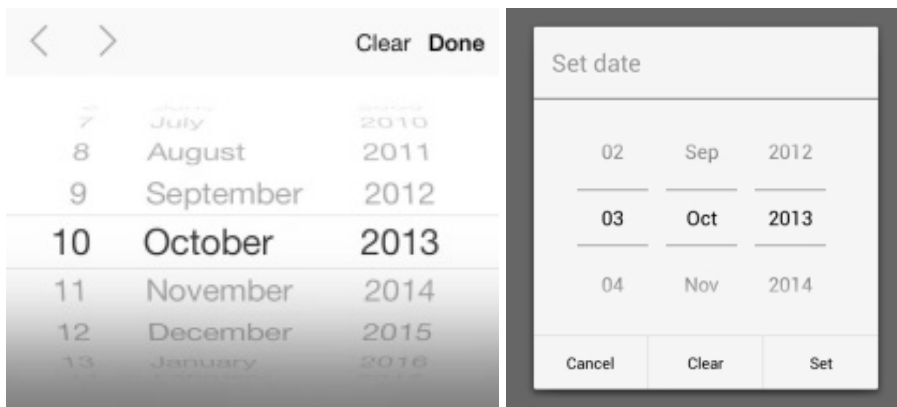


```
<input type="number">
```

12345

date

Pole pro datum. Zobrazí se rozbalovací kalendář.



```
<input type="date">
```

email

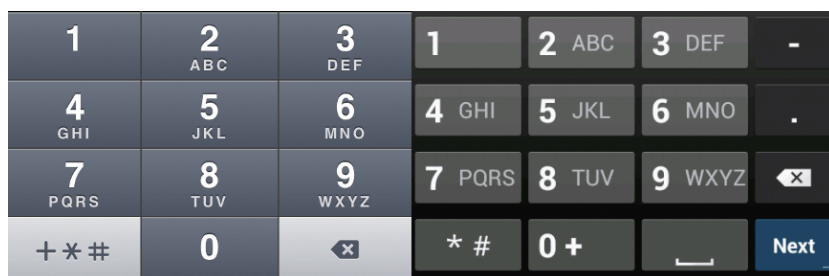
Pole pro email. Před odesláním **automaticky kontroluje tvar emailu**. Na mobilu se navíc zobrazí klávesnice se @ a . :



```
<input type="email">
```

tel

Stejně, jako input `type="text"`. Přidáním nepovinného atributu `inputmode="numeric"` zobrazují mobily telefonní klávesnici. Bez něj alfanumerickou:



```
<input type="tel" inputmode="numeric">
```

hidden

Skryté pole s nastavenou hodnotou. Hodí se pro kontrolu, zda byl formulář odeslán.

```
<input type="hidden" value="odeslano" name="h">
```

Kompletní výčet najdete na www.w3schools.com/tags/att_input_type.asp.

name="..."

Name je **nejdůležitější atribut** formulářových elementů. Každý input jej musí mít.

```
Jméno <input type="text" name="jmeno"><br>
Příjmení <input type="text" name="prijmeni">
```

Identifikuje je a posílá své jméno s daty:

Jméno

Příjmení

Symbolicky po odeslání vypadá URL takto:

```
http://.../?jmeno=Jan&prijmeni=Hus
```

value="..."

Hodnota, obsah polí. U některých typů inputů povinné (checkbox, radio, submit) => odesílá se na server, u některých volitelné (text, tel). U těch místo atributu value doporučuji využít atribut placeholder.

placeholder="..."

atribut, který umožňuje nastavit **výchozí text uvnitř prvku**. Rozdíl od HTML 4 je ten, že placeholder text se zobrazí, jen když je políčko prázdné (nemá focus). Jakmile focus obdrží (například klikem myši nebo tabulátorem) a začnete psát, text zmizí. Šlo by to udělat pomocí atributu title (jako např. u odkazu), ale ten by vyvolal pouze bublinu po najetí myší.

```
Jméno <input type="text" name="prijmeni" placeholder="Křestní jméno">
```

Jméno

Živá ukázka (zejména pro mobily a chytré hodinky) na <http://tux.panska.cz/~machac/form>

tux.panska.cz/~machac/form/



| | |
|---------------|------------|
| type=text | |
| type=password | |
| type=number | |
| type=email | |
| type=tel | |
| type=url | |
| Datum | dd/mm/yyyy |
| Čas | --:-- |
| Odeslat | Vymazat |

HTML 5 přidává 14 nových a užitečných formulářových atributů. Oproti HTML 4 nám výrazně zjednodušují práci. Co by bylo nutné kódovat v PHP nebo JS, je již přímo integrované. My si představíme tři:

1. Vynucení vyplnění pole (`required`)
2. Kontrola hodnoty podle vzoru (`pattern`)
3. Kontrola správného tvaru e-mailové adresy (zabudované v prohlížeči)

Kontrola se provádí ještě před odesláním na server.

Některé atributy jsou booleovské. To znamená, že jsou-li přítomné, jsou zapnuté, a jsou-li nepřítomné, jsou vypnuté.

required

Atribut `required` nepotřebuje moc vysvětlování; dělá právě to, co byste od něj čekali. Když ho přidáte na políčko, prohlížeč po vás bude chtít, abyste ho před odesláním vyplnili. Díky tomu můžeme ušetřit čas, který bychom jinak strávili validací (kontrolou) přes JavaScript nebo PHP.

```
<input type="number" name="vek" required>
```

TIP: Je-li hláška anglicky, přidejte do značky html jazyk: `<html lang="cs">`

Text a vzhled hlášky v bublině generuje prohlížeč - může se lišit.



pattern="..."

(česky vzor). Vzor je nějaký **regulární výraz** (*regex* nebo *regexp*). Je to mechanismus pro porovnávání řetězců (textu, čísel) proti určitým **vzorům**. Ty lze využít ke kontrole formátu PSČ, telefonu a podobně.

Pozn: Regulární výrazy lze použít v celé řadě jiných programovacích jazyků.

Atribut `pattern` potěší zejména vývojáře. Definuje JavaScriptový [regulární výraz](#), který se použije pro kontrolu hodnoty v políčku. Díky němu můžeme lehce psát validace (kontrolu správnosti) na specifická čísla, hesla, e-mailů a podobně. Možnosti tohoto atributu jsou široké.

Příklad na vkládání telefonních čísel:

```
<input type="tel" name="kod" pattern="[0-9]{9}">
```

Kontrola nějakého kódu. Hodnota musí být ve formátu číslo `[0-9]` a tři velká písmena `[A-Z]{3}`:

```
<input type="text" name="kod" pattern="[0-9][A-Z]{3}">
```

Heslo. Velká písmena, malá písmena a čísla:

```
^(?=.*\d)(?=.*[a-z])(?=.*[A-Z])(?!.*\s).*$
```

Heslo. Velká písmena, malá písmena, čísla a spec. znaky. Délka min 8 znaků:

```
(?=.*\.{8,}$)((?=.*\d)|(?.*\W+))(?![\.\n])(?=.*[A-Z])(?=.*[a-z]).*$
```

IPv4 adresa (např. 192.168.100.52):

```
((^|\.)((25[0-5])|(2[0-4]\d)|(1\d\d)|([1-9]?\d)))\{4}$
```

- Další hotové příklady lze najít na stránce HTML5 Pattern <http://html5pattern.com>
- Obecně o syntaxi např. zde <https://www.regularnivyrazy.info/shrnuti-syntaxe.html>
- Testovací web formulářových polí a atributů <https://inputtypes.com>

<select>

```
<select></select>
```

Výběr z více možností. Zobrazí obdélník s rozbalovacími možnostmi nebo roletkové menu. Párová značka, jeho obsahem jsou jednotlivé volby `<option>...</option>`.

Atributy tagu `<select>`

| Atribut | Význam | Hodnoty |
|------------------------|---|-----------------|
| <code>name</code> | jméno pole odesílané s daty (povinné) | libovolné jméno |
| <code>size</code> | počet zobrazených řádků | číslo |
| <code>multiple</code> | umožnění hromadného výběru (s Ctrl) | bez hodnoty |
| <code>disabled</code> | políčko bude šedé a nepůjde měnit | bez hodnoty |
| <code>autofocus</code> | ihned po načtení stránky bude políčko aktivní (půjde hodnotu měnit šipkami). Hodí se, pokud je tento prvek jediný na stránce. | bez hodnoty |

Příklady:

```
<select name="ovoce">
  <option value="jb">jablko</option>
  <option value="hs">hruška</option>
  <option value="sv">švestka</option>
</select>
```

jablko ▾

```
<select name="ovoce" size="3">
  <option value="jb">jablko</option>
  <option value="hs">hruška</option>
  <option value="sv">švestka</option>
</select>
```

jablko ▲
hruška
švestka ▾

`<fieldset>` a `<legend>`

```
<fieldset>
```

Opticky sdruží několik prvků formuláře a vykreslí kolem nich rámeček. Čáru rámečku může přerušit text tagu `<legend>`. `<fieldset>` nemá žádné atributy.

Vzhled se dá upravovat pomocí CSS.

```
<legend>
```

Popisek skupiny polí. Zobrazuje se nad skupinou polí přes horní čáru fieldsetu. Musí být zapsán hned za značkou `<fieldset>`. Obsahem značky `<legend>` může být libovolné HTML (rozumné). Zarovnání popisku se provádí atributem `align=left` (výchozí) nebo `center` nebo `right`.

Příklad:

```
<fieldset>
  <legend>Přihlášení</legend>
  E-mail: <input type="email" name="jmeno"><br />
  Heslo: <input type="password" name="heslo"><br />
  <input type="submit" value="Odeslat">
</fieldset>
```

Přihlášení

E-mail:

Heslo:

Nebo lépe a moderněji:

```
<fieldset>
  <legend>Přihlášení</legend>
  <input type="email" name="jmeno" placeholder="E-mail"><br />
  <input type="password" name="heslo" placeholder="Heslo"><br />
  <input type="submit" value="Odeslat">
</fieldset>
```

Přihlášení

E-mail

Heslo