

## 12 FORMULÁRE

S formulármi na webových stránkach sa stretávame praktický dennodenne. Veď už len prihlásenie do e-mailového účtu znamená vyplnenie formulára s dvoma položkami: prihlasovacie meno a heslo. Takisto pri vyhľadávaní zadávame hľadaný výraz do textového políčka nejakého formulára. Aj mnohé úradné, predtým len papierové formuláre, už dnes majú aj svoju elektronickú podobu.

### DISKUTUJTE

S akými formulármi ste sa stretli na internete? Na akých stránkach? Aké formuláre vyplňate pravidelne? Už ste sa niekedy rozhodovali medzi papierovou a webovou verziou formulára? Pri akej príležitosti?



Formuláre využívame všade tam, kde potrebujeme od používateľa získať nejaké vstupy, údaje. Tie sa potom nejakým spôsobom spracujú. Pod spracovaním formulára si môžete predstaviť napr. kontrolu údajov, ktoré používateľ zadal, uloženie údajov zväčša do databázy, spätnú väzbu pre používateľa (napr. potvrdenie objednávky, rekapitulácia údajov, ktoré zadal, alebo výpis chýb, ak niečo vo formulári vyplnil nesprávne).

V tejto kapitole sa naučíme **vytvárať** formuláre s rôznymi prvkami. Nebudeme sa však zaoberať ich spracovaním. Formulár takmer vždy finálne spracováva nejaký skript (program) na strane servera, napr. PHP, ASP, Perl a pod. (nie JavaScript). Len pomocou jazyka HTML formuláre spracovávať nevieme.

### 12.1 Definovanie formulára

Definovanie formulára a jeho dôležité atribúty si ukážeme na príklade jednoduchého vyhľadávacieho formulára (*obrázok 12.1*).

The image shows a simple web form with a single text input field on the left and a button labeled "Hľadaj" on the right. The entire form is enclosed in a thin black border.

Obrázok 12.1 Jednoduchý vyhľadávací formulár.

#### PRÍKLAD 12.1

Vytvoríme vyhľadávací formulár s textovým poľom na zadanie hľadaného textu a tlačidlom *Hľadaj*.

Formulár vytvoríme v novom súbore, takže najskôr musíme definovať všetky základné elementy (`doctype`, `html`, `head`, `body`), nastaviť titulok a kódovanie (pozri koniec 1. kapitoly). Súbor uložíme ako `hladaj.html`. Do tela stránky (`body`) napíšeme nasledujúci kód:

```
<body>
  <form>
    <input type="text">
    <input type="submit" value="Hľadaj">
  </form>
```



```
</body>
```

Formulár definujeme pomocou elementu `<form>`. Obsahom elementu `form` sú jednotlivé prvky formulára a ich popisy. Náš formulár obsahuje dva prvky, textové pole a odosielacie tlačidlo.

Textové pole definujeme pomocou elementu `input` s atribútom `type` nastaveným na hodnotu `text`, teda `<input type="text">`. Elementom `input` sa definujú rôzne typy prvkov formulára, pomocou atribútu `type` určujeme, o aký typ prvku formulára ide.

Odosielacie tlačidlo je špeciálne tlačidlo, ktorým odosieme údaje z formulára na spracovanie. Je nevyhnutnou súčasťou každého formulára, inak nedokážeme preniesť, a teda ani spracovať, hodnoty zadané používateľom. Odosielacie tlačidlo definujeme pomocou elementu `<input type="submit">`. V atribúte `value` uvádzame text, ktorý sa vypíše na tlačidlo.



## ÚLOHA 12.2

Experimentujte s formulárom z príkladu 12.1.

- zmeňte hodnotu atribútu `value` pre tlačidlo na *Search*,
- zrušte atribút `value` pre tlačidlo.

Element `form` môže mať niekoľko atribútov. Uvedieme dva z nich: `action` a `method`.

Hodnotou atribútu `action` je adresa súboru, ktorý má údaje z formulára spracovať, t.j. súboru, ktorý obsahuje program vo vhodnom jazyku, ktorý nejakým spôsobom spracuje hodnoty odoslané z formulára. Napríklad, ak by sme formulár z príkladu 12.1 zadefinovali elementom `<form action="vyhladaj.php">`, tak po odoslaní (stlačení tlačidla *Hľadaj*) by sa výraz zadaný do textového poľa poslal súboru `vyhladaj.php`. Ak atribút `action` neuvedieme, bude formulár spracovávať ten súbor, v ktorom je formulár uložený – v našom prípade súbor `hladaj.html`.

Atribút `method` určuje spôsob odosielania údajov z formulára. Môže mať hodnotu `get` alebo `post`.

- pri odosielaní metódou `get` sa všetky údaje zadané do formulára prenesú do adresy prehliadača. Znamená to, že ich môže vidieť niekto, kto stojí pri používateľovi, resp. si ich môže neskôr pozrieť v histórii prehliadača. Pomocou metódy `get` môžeme preniesť len obmedzený objem údajov. Metóda sa používa na prenos údajov, ktoré nie sú citlivé, ako napr. vyhľadávaný výraz (vyhľadávače), alebo číslo strany pri stránkovaných webových dokumentoch,
- pri odosielaní metódou `post` sa hodnoty z formulára posielajú "skryté", teda nie sú prenášané cez adresu prehliadača, ani ich nevieme zistiť z histórie prehliadača. Z toho vyplýva, že je o niečo bezpečnejšia ako metóda `get`. Používame ju na prenos citlivých a osobných údajov. Pomocou metódy `post` môžeme preniesť väčší objem údajov, aj uploadovať súbory.

Formulár, ktorý sme vytvorili v príklade 12.1 síce vyzerá, ako sme chceli, ale nesplnil by svoju funkciu, pretože zadané údaje by sa vôbec nedoslali.

**Každý prvok formulára musí mať definovaný atribút `name`.** Pomocou jeho hodnoty vieme pri spracovaní formulára pristupovať k údajom zadaným v jednotlivých položkách formulára. Až na isté výnimky, ktoré si ukážeme neskôr, platí, že hodnota atribútu `name` musí byť pre každý prvok formulára unikátna. Mená formulárových prvkov sa nesmú začínať číslom, nesmú obsahovať medzeru a neodporúčame ani diakritiku.

### PRÍKLAD 12.3

Upravme formulár z príkladu 12.1. Definujme každému prvku formulára atribút `name` a nastavme spôsob odoslania formulára na metódu `post`.

```
<form method="post">
  <input type="text" name="retazec">
  <input type="submit" value="Hľadať" name="odosli">
</form>
```

### ÚLOHA 12.4

Vo formulári z príkladu 12.3 zadajte do textového poľa nejaký text a odošlite ho. Zmeňte metódu odoslania údajov na `get`. Stránku s formulárom načítajte znova (nie pomocou F5, ale kliknite na adresu stránky a stlačte Enter!). Zadajte do textového poľa nejaký text a odošlite. Všímite si adresu v prehliadači.

## 12.2 Prvky formulára

Vo formulároch zväčša buď niečo doplníme (kratší text, dlhší text, čísla) alebo vyberáme z ponuky možností. Pomocou jazyka HTML vieme zdefinovať niekoľko rôznych typov prvkov určených na dopĺňanie textu aj na výber z ponuky.

### ÚLOHA 12.5

Nájdite na internete nejaký formulár s aspoň piatimi položkami. Aké prvky sa v ňom nachádzajú? Skúste nájsť formulár, v ktorom je čo najviac rôznych typov prvkov.

V nasledujúcej časti budeme postupne vytvárať formulár pre uchádzača o prácu v pizzerii IT Pizza. Ukážeme si na ňom, aké rôzne typy formulárových prvkov môžeme používať a ako sa definujú.

### Jednoriadkové textové pole

Textové pole je jeden z najčastejšie používaných formulárových prvkov. Už vieme, že ho definujeme elementom `<input type="text">`, aj to, že každému takému prvku musíme dať atribút `name`.



## ÚLOHA 12.6

Vytvorte novú stránku `prihlaska.html` a v nej definujte formulár s dvoma textovými poľami, jedno na zadanie mena, druhé na zadanie priezviska (obrázok 12.2). Pre element `form` nenastavujte žiadne atribúty. Pre elementy `input` nastavte všetky dôležité atribúty. Doplňte aj popisy k formulárovým prvkom *Meno*, *Priezvisko* – obyčajné texty, v zdrojovom kóde pred príslušným formulárovým elementom. Na formátovanie do viacerých riadkov môžete využiť napr. element `<br>`.

Obrázok 12.2 Formulár na zadanie mena a priezviska.

Pre textové pole môžeme nastaviť šírku pomocou atribútu `width`. Hodnotou atribútu je číslo, ktoré definuje šírku textového poľa v znakoch. Atribút nelimituje počet znakov, ktoré dokážeme do poľa zapísať, iba to, koľko ich môžeme naraz vidieť. Ak zapíšeme do poľa viac znakov, začne sa obsah poľa posúvať.

V prípade, že chceme pre textové pole nastaviť nejakú počiatočnú hodnotu, ktorá sa v ňom zobrazí už pri načítaní stránky, použijeme atribút `value`, ktorého hodnotou môže byť ľubovoľný reťazec (obrázok 12.3).

Obrázok 12.3 Formulár: položka Mobil má prednastavenú hodnotu +421.

### PRÍKLAD 12.7

Do formulára z úlohy 12.6 pridáme textové pole na zadanie telefónneho čísla. Jeho veľkosť nastavíme na 15 znakov a počiatočnú hodnotu na +421 (obrázok 12.3).

```
<form>
  Meno: ...
  Priezvisko: ...
  Mobil: <input type="text" name="mobil" size="15"
value="+421"><br>
  <input type="submit" value="Odošli" name="odosli">
</form>
```

### ÚLOHA 12.8

Do formulára z príkladu 12.7 doplňte textové pole *E-mail* s prednastavenou hodnotou @. Experimentujte s hodnotou atribútu `width` pre položku *Meno*. Vyskúšajte 5, 10, 20, 30,...

### Prepínač (radiobutton)

Prepínače (radiobuttony) vo formulári používame vtedy, ak chceme, aby používateľ zvolil práve jednu z istého (menšieho) počtu možností.

Prepínač definujeme pomocou elementu `<input type="radio">`. Každý prepínač by mal mať nastavené atribúty `name` a `value`. Ak chceme definovať skupinu prepínačov, z ktorých sa má dať vybrať len jedna možnosť, tak **všetky položky skupiny musia mať rovnaké meno** (atribút `name`), ale budú sa líšiť hodnotami (atribút `value`). Vybraná hodnota bude po odoslaní formulára hodnotou celej skupiny (elementu).

### PRÍKLAD 12.9

Do formulára z úlohy 12.8 pridáme dva prepínače na výber prevádzky, do ktorej chce uchádzač nastúpiť, jeden pre Bratislavu a druhý pre Banskú Bystricu.

```
<form>
  ...
  E-mail: ...
  Prevádzka:
  <input type="radio" name="mesto" value="BA"> Bratislava
  <input type="radio" name="mesto" value="BB"> Banská Bystrica
  <br>
  <input type="submit" value="Odošli" name="odosli">
</form>
```

Atribútom `value` definujeme hodnotu, ktorá sa po odoslaní formulára pošle na spracovanie (pošle sa len hodnota toho prepínača, ktorý je zvolený). Popis, ktorý sa pri prepínači zobrazí, nie je súčasťou elementu `input`, je to obyčajný text, ktorý umiestňujeme zväčša za prepínač. Môže, ale nemusí sa zhodovať s hodnotou (`value`) prepínača.

Ak chceme mať niektorý prepínač označený už pri načítaní stránky, pridáme do jeho definície atribút `checked`, napríklad `<input type="radio" name="mesto" value="BB"`

checked>.

Meno:   
Priezvisko:   
Mobil: +421   
E-mail: @   
Prevádzka:  Bratislava  Banská Bystrica

Obrázok 12.4 Formulár s prepínačmi pre výber prevádzky.



### ÚLOHA 12.10

Do formulára z príkladu 12.9 pridajte ďalšiu skupinu prepínačov na zadanie pozície, na ktorú chce uchádzač nastúpiť. Pozíciu *kuchár* zvolte ako prednastavenú (pozri obrázok 12.5).

E-mail: @   
Prevádzka:  Bratislava  Banská Bystrica  
Pozícia:  kuchár  čašník  vodič  pomocná sila

Obrázok 12.5 Prepínač pre výber pracovnej pozície.

### Začiarkávacie políčko (checkbox)

Začiarkávacie políčka zvyčajne používame vtedy, ak používateľ nemusí zvoliť žiadnu z ponúkaných možností, alebo ich môže označiť viacero. Pri začiarkávacích políčkach, na rozdiel od prepínačov, môžeme každú možnosť označiť aj odznačiť.

Na stránku môžeme vkladať začiarkávacie políčka samostatne alebo môžeme vytvoriť skupinu (podobne ako pri prepínačoch). Tvorba skupiny začiarkávacích políčok však nie je taká jednoduchá ako pri prepínačoch, pričom tiež záleží, ako budeme formulár spracovávať. Preto sa nebudeme zaoberať možnosťou tvorby skupiny začiarkávacích políčok, ale len samostatnými začiarkávacími políčkami.

Začiarkávacie políčko definujeme pomocou elementu `<input type="checkbox">`. Každé začiarkávacie políčko musí mať nastavené atribúty `name` a `value`. Ak chceme mať niektoré začiarkávacie políčko označené už pri načítaní stránky, pridáme mu atribút `checked`.

## PRÍKLAD 12.11



Vo formulári chceme zisťovať, aké jazyky uchádzač o prácu ovláda. Do formulára z úlohy 12.10 pridáme štyri začiarkávacie políčka s popismi *anglický*, *nemecký*, *taliansky* a *španielsky* (obrázok 12.6). Ako mená a hodnoty použijeme skratky, napr. aj, nj, tj a sj. Anglický jazyk bude začiarknutý už pri načítaní stránky.

```
<form>
  ...
  Jazyky:<br>
  <input type="checkbox" name="aj" value="aj" checked> anglický
  <br>
  <input type="checkbox" name="nj" value="nj"> nemecký<br>
  <input type="checkbox" name="tj" value="tj"> taliansky<br>
  <input type="checkbox" name="sj" value="sj"> španielsky
  <br>
  <input type="submit" value="Odošli" name="odosli">
</form>
```

Jazyky:

anglický

nemecký

taliansky

španielsky

Odošli

Obrázok 12.6 Formulár so začiarkávacími políčkami.

## ÚLOHA 12.12



Do formulára z príkladu 12.11:

- pridajte začiarkávacie políčko s nejakým ďalším jazykom – zvolte vhodne atribút `name`, tak, aby bol unikátny,
- zrušte začiarknutie anglického jazyka,
- pred odosielacie tlačidlo pridajte začiarkávacie políčko s menom `suhlas`, hodnotou `suhlasim` a popisom *Súhlasím so spracovaním osobných údajov* (obrázok 12.7).

Jazyky:

anglický

nemecký

taliansky

španielsky

maďarský

Súhlasím so spracovaním osobných údajov

Odošli

Obrázok 12.7 Začiarkávacie políčka.

## Výberová ponuka **SELECT**

Výberová ponuka umožňuje používateľovi vybrať jednu z viacerých ponúkaným možností. Možnosti však nie sú viditeľné hneď pri načítaní stránky, ale až po rozbalení ponuky, preto sa niekedy nazýva aj rozbaľovacia ponuka (v angličtine *drop down list*). Používa sa zväčša vtedy, keď je v ponuke veľa možností, takže riešenie pomocou prepínačov by zabralo príliš veľa miesta, napr. na zadanie dňa v mesiaci (31 možností).

Výberovú ponuku definujeme pomocou elementu `<select></select>`. Obsahom elementu sú jednotlivé prvky ponuky, ktoré definujeme pomocou elementu `<option></option>`. Popis jednotlivých prvkov sa vkladá medzi značky `<option></option>`. Element `select` musí mať nastavený atribút `name`. Element `option` nemá atribút `name`, ale mal by mať nastavený atribút `value`. Atribút `value` zvolenej možnosti sa posiela po odoslaní formulára ako hodnota celého elementu `select`. Štandardne je nastavený (a viditeľný) prvý prvok ponuky. Ak chceme prednastaviť nejakú inú možnosť, urobíme tak v príslušnom elemente `option` pomocou atribútu `selected`. Vo výberovej ponuke je teda vždy niektorá z hodnôt prednastavená, kým pri prepínačoch a začiarkávacích políčkach nie.



### PRÍKLAD 12.13

Vo formulári z predchádzajúcej úlohy nahradíme prepínače na výber pozície výberovou ponukou. Pozícia *čaušník* bude prednastavená (obrázok 12.8).

```
...
Pozícia:
<select name="pozicia">
  <option value="kuchár">kuchár</option>
  <option value="čaušník" selected>čaušník</option>
  <option value="vodič">vodič</option>
  <option value="pomocná sila">pomocná sila</option>
</select>
<br>
Jazyky:<br>
...
```

Pozícia:

- kuchár
- čaušník
- vodič
- pomocná sila

Obrázok 12.8 Výberová ponuka.



### ÚLOHA 12.14

Do výberovej ponuky z predchádzajúcej úlohy pridajte možnosť: *roznášač*.





### ÚLOHA 12.15

Do formulára z predchádzajúcej úlohy za položku *E-mail* pridajte výberovú ponuku na zadanie vekovej kategórie. Vytvorte možnosti podľa obrázka 12.9.

The image shows a web form with several input fields. The 'E-mail' field contains '@'. Below it, a dropdown menu is open, showing age categories: '18 - 25' (selected), '25 - 30', '30 - 40', '40 - 50', and '50 a viac'. To the right of the dropdown, there are radio buttons for 'Trnava' and 'Banská Bystrica'. Below the dropdown, there are radio buttons for 'slovensky' and 'nemecky'.

Obrázok 12.9 Výberová ponuka.

Element `select` môže mať navyše dva atribúty:

- `size`, ktorým určíme počet viditeľných možností (štandardne je jedna)
- a `multiple`, ktorým umožníme používateľovi zvoliť viac ako jednu hodnotu.

Ak by sme element `<select name="pozicia">` z nášho formulára zmenili na `<select name="pozicia" size="3" multiple>`, zobrazila by sa ponuka po načítaní stránky tak, ako na obrázku 12.10.

The image shows a web form with a dropdown menu labeled 'Pozícia:'. The dropdown is open, showing three options: 'kuchár', 'čajašník', and 'vodič'. The 'vodič' option is selected.

Obrázok 12.10 Výberová ponuka s možnosťou výberu viacerých prvkov.

### Viacriadkové textové pole TEXTAREA

Viacriadkové textové pole, ako už názov hovorí, slúži na zadanie dlhšieho textu, ktorý obsahuje viac riadkov (čiže pri jeho písaní môžeme použiť aj kláves Enter).

The image shows a web form with a label 'Vzdelanie:' followed by a text area input field.

Obrázok 12.11 Viacriadkové textové pole.

Viacriadkové textové pole definujeme pomocou elementu `<textarea></textarea>`. Element musí mať nastavený atribút `name`.

Element `textarea` nemá atribút `value` ako ostatné formulárové elementy. Ak chceme, aby vo viacriadkovom textovom poli bola pri načítaní stránky nejaká prednastavená hodnota (text), uvedieme ju medzi počiatočnú a koncovú značku elementu v zdrojovom kóde, napr. `<textarea>základná škola</textarea>`.



### PRÍKLAD 12.16

Do formulára s prihláškou pridáme viacriadkové textové pole s popisom *Vzdelanie* (obrázok 12.11). Umiestnime ho za prepínače na výber prevádzky.

```
...
Prevádzka:
 Bratislava
 Banská Bystrica
<br>
Vzdelanie: <textarea name="vzdelanie"></textarea>
<br>
Pozícia:
...
```

Veľkosť viacriadkového textového poľa môžeme nastaviť pomocou atribútov `cols` (počet stĺpcov) a `rows` (počet riadkov), napr. `<textarea name="text" cols="25" rows="5"></textarea>`. Podobne ako pri jednoriadkovom textovom poli ide len o šírku a výšku "viditeľnej časti", **neobmedzuje** sa tým počet riadkov, ktoré môžeme do viacriadkového textového poľa zapísať (ak ich zapíšeme viac, zobrazí sa posúvač).



### ÚLOHA 12.17

Do formulára z predchádzajúceho príkladu pridajte viacriadkové textové pole *Prax*, veľkosti 25 stĺpcov a 5 riadkov, bez prednastavenej hodnoty. Umiestnite ho za viacriadkové pole *Vzdelanie* (obrázok 12.12).

Obrázok 12.12 Viacriadkové textové polia *Vzdelanie* a *Prax*.

### Tlačidlo

V úvode kapitoly sme si ukázali, ako pomocou elementu `<input type="submit">` definujeme odosielacie tlačidlo. Ak zmeníme hodnotu `submit` na hodnotu `reset`, t.j. `<input type="reset">`, definujeme tzv. resetovacie tlačidlo – tlačidlo, ktoré "zmaže" všetky údaje zadané používateľom a nastaví preddefinované hodnoty.

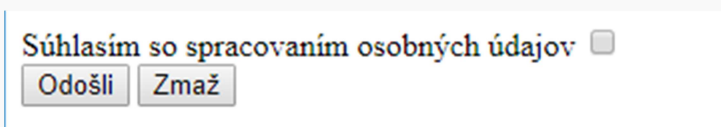
Tlačidlá môžeme definovať aj pomocou elementu `<button></button>`. Na rozdiel od elementu `<input>`, element `button` ponúka oveľa viac možností (pomocou CSS). Na tlačidlo môže byť zobrazený nielen text, ale napríklad aj obrázok. Element je párový, jeho obsahom je text, ktorý sa zobrazí na tlačidle. Elementu môžeme definovať atribúty `name`, `value` a `type`. Pomocou atribútu `type` určujeme, o aký typ tlačidla ide. Hodnota `submit` znamená

odosielacie tlačidlo, hodnota `reset` resetovacie tlačidlo a hodnota `button`, obyčajné tlačidlo (zvykne sa používať v spojení s JavaScriptom na vykonanie nejakého JS kódu.)

### PRÍKLAD 12.18

Vo formulári s prihláškou zmeníme definíciu odosielacieho tlačidla pomocou elementu `button`. Taktiež pridáme tlačidlo na zmazanie údajov.

```
...  
<br>  
<button type="submit" value="odosli" name="odosli">Odošli  
</button>  
<button type="reset" value="reset" name="reset">Zmaž</button>  
</form>
```



Súhlasím so spracovaním osobných údajov

Odošli Zmaž

Obrázok 12.13 Tlačidlá na odoslanie formulára a zmazanie údajov.

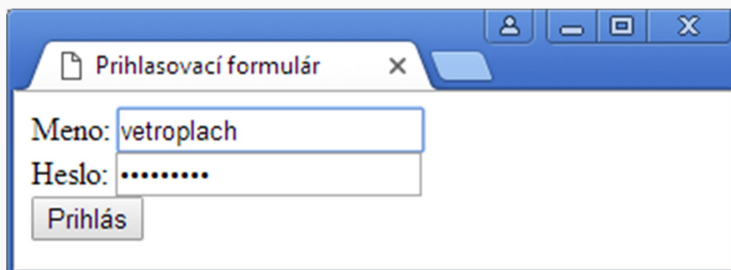
### Jednoriadkové textové pole na zadanie hesla

Textové pole na zadanie hesla je v podstate rovnaké ako obyčajné jednoriadkové textové pole s tým rozdielom, že text, ktorý do neho používateľ píše, sa nezobrazuje, ale nahrádza hviezdičkami. Ak máme vo formulári tento prvok, určite by sme ho mali odosielať metódou `post`.

Textové pole na zadanie hesla definujeme pomocou elementu `<input type="password">`. Ako všetky ostatné formulárové elementy, musí mať definovaný atribút `name`. Môžeme mu nastaviť veľkosť pomocou atribútu `size` ako obyčajnému textovému poľu.

### ÚLOHA 12.19

Vytvorte novú stránku `login.html` s prihlasovacím formulárom (obrázok 12.14). Formulár bude obsahovať jedno obyčajné textové pole, jedno textové pole na zadanie hesla a tlačidlo na odoslanie. Tlačidlo realizujte pomocou elementu `button`.



Prihlasovací formulár

Meno: vetroplach

Heslo: .....

Prihlás

Obrázok 12.14 Prihlasovací formulár.

## Nové formulárové prvky a atribúty v HTML5

Štandard HTML5 zaviedol niekoľko nových prvkov formulára. Všetky definujeme pomocou elementu `input`, typ prvku určujeme pomocou atribútu `type`. Za všetky spomenieme nasledujúce tri:

- prvok na výber **farby** – element `<input type="color">`,
- prvok na zadanie či výber **dátumu** – element `<input type="date">`,
- prvok na zadanie **času** – element `<input type="time">`.

Vo všetkých troch prípadoch ide o komplexnejšie prvky, ktoré sú sami o sebe malými formulármi. Pri ich použití si musíme byť vedomí, že **nie sú podporované staršími prehliadačmi**. Staršie prehliadače zobrazia nové typy prvkov ako jednoriadkové textové pole.



### ÚLOHA 12.20

Vyskúšajte formulár v súbore `12/nove.html`. Zobrazujú sa vo vašom prehliadači podobne ako na *obrázku 12.15*, alebo vidíte len obyčajné textové polia?

The screenshot shows a web browser window titled "Nové prvky formulára v H". The form contains three input fields:

- Farba:** A color picker showing an orange color.
- Dátum:** A date picker showing "dd. mm. rrrr" and a time picker showing "a čas: 16:05".
- Odošli:** A submit button.

The date picker is open, showing a calendar for August 2018. The calendar has a header with days of the week (po, ut, st, št, pi, so, ne) and a grid of dates from 30 to 2.

Obrázok 12.15 Formulár s novými prvkami na výber farby, zadanie dátumu a času.

Formulárovým prvkom definovaným pomocou elementu `input` môžeme pridať atribút `required`, ktorým určíme, že daná položka formulára je povinná. V prípade, že používateľ stlačí odosielacie tlačidlo a niektorý z údajov s atribútom `required` nebude vyplnený, formulár sa neodošle a prehliadač používateľa upozorní na "chybu" (*obrázok 12.16*).

## PRÍKLAD 12.21



Položky *Meno*, *Priezvisko* a *Prevádzka* vo formulári `prihlaska.html` z úlohy 12.18 definujeme ako povinné.

```
<form method="get">
  Meno: <input type="text" name="meno" size="20" required><br>
  Priezvisko: <input type="text" name="priezvisko" required><br>
  ...
  ...
  Prevádzka:
  <input type="radio" name="mesto" value="BA" required>
  Bratislava
  <input type="radio" name="mesto" value="BB" required>
  Banská Bystrica
  ...
```

Obrázok 12.16 Reakcie na nezadanie povinných údajov.

Pre elementy `input` typu `text` a `password` môžeme definovať atribút `placeholder`. Slúži na definovanie pomocného textu, ktorý popisuje očakávanú hodnotu v príslušnom prvku formulára. Môže ísť o vysvetlenie očakávaného formátu položky, príklad vyplnenia, alebo nejaké obmedzenia na položku (napr. *len čísla*). Pomocný text sa zobrazuje priamo v položke, kým do nej používateľ nezadá nejakú hodnotu (obrázok 12.17).

## PRÍKLAD 12.22



Vo formulári z predchádzajúcej úlohy zrušíme prednastavenú hodnotu položke *Mobil* a namiesto nej definujeme `placeholder` pre položku *Mobil* (obrázok 12.17).

Obrázok 12.17 Placeholder pre položku Mobil.

```
...
Mobil: <input type="text" name="mobil" size="15"
  placeholder="+421903123456"><br>
...
```

Keďže atribúty `required` a `placeholder` sú definované až v štandarde HTML5, nemajú podporu v starších prehliadačoch. Pri tvorbe a spracovaní formulára na to musíme myslieť.



### ÚLOHA 12.23

Vo formulári z príkladu 12.22:

- položky *Mobil*, *E-mail* a *Súhlas so spracovaním údajov* zmeňte na povinné,
- zrušte prednastavenú hodnotu pre položku *E-mail*,
- pre položku *E-mail* definujte rozumný `placeholder`.



### ÚLOHA 12.24

Vytvorte formulár `rezervacia.html` na rezerváciu stola v pizzérii podľa obrázka 12.18. Stôl bude možné rezervovať v prevádzkach Bratislava alebo Banská Bystrica pre 0 až 10 osôb v čase od 10:00 po 19:45, len na rok 2018 alebo 2019. Zvážte, ktoré položky majú byť povinné.

**Rezervácia**

Meno:

Telefón:

Počet osôb:

Mesto:  Bratislava |  Banská Bystrica

Dátum:

Čas:

- 15
- 30
- 45

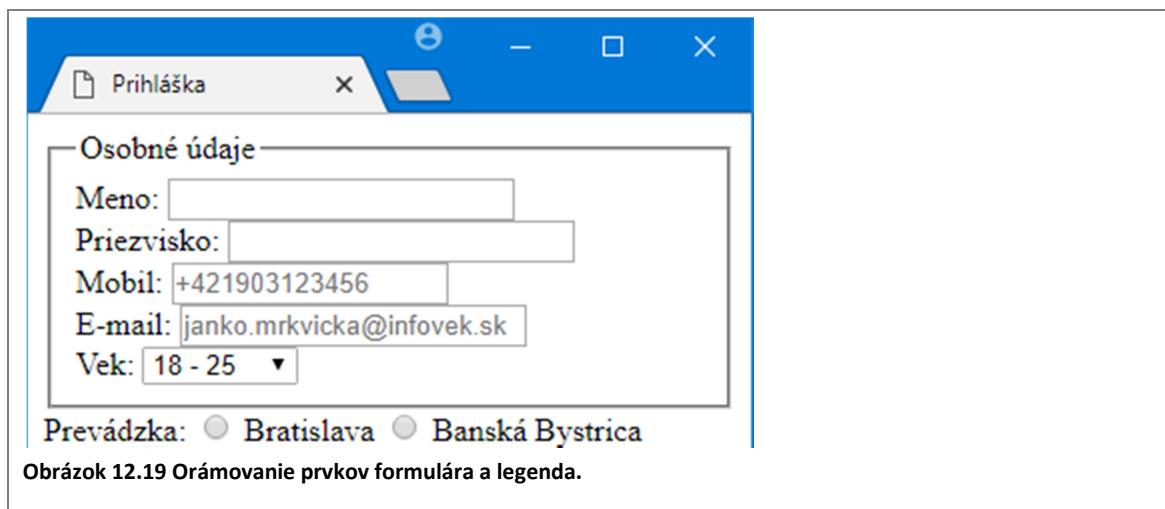
Obrázok 12.18 Rezervačný formulár.

## 12.3 Štruktúrovanie formulára

Ak formulár obsahuje väčšie množstvo prvkov, môže sa stať neprehľadným. Jedna z možností, ako formulár sprehľadniť, je použiť blokové elementy alebo tabuľky. Špeciálne na použitie vo formulároch je určený blokový element `<fieldset></fieldset>`.

Element `fieldset` umožňuje logické i optické zoskupenie prvkov formulára – zobrazí orámovanie okolo elementov, ktoré sú jeho obsahom (obrázok 12.19). Každý element `fieldset` môže mať svoju legendu, ktorú definujeme pomocou elementu `<legend></legend>`. Element `legend` musí byť definovaný vnútri elementu `fieldset`.

Elementy `fieldset` a `legend` nemajú žiaden vplyv na funkčnosť formulára.



Obrázok 12.19 Orámovanie prvkov formulára a legenda.

### PRÍKLAD 12.25



Vo formulári `prihlaska.html` z úlohy 12.23 zoskupíme osobné údaje (meno, priezvisko, mobil, email, vek) a definujeme pre ne legendu (obrázok 12.19).

```
<form>
  <fieldset>
    <legend>Osobné údaje</legend>
    Meno: <input type="text" name="meno" size="20" required"><br>
    Priezvisko: <input type="text" name="priezvisko" required>
    <br>
    Mobil: <input type="text" name="mobil" size="15"
      placeholder="+421903123456"><br>
    E-mail: <input type="text" name="email"
      placeholder="janko.mrkvicka@infovek.sk"><br>
    Vek:
    <select name="vek">
      <option value="18-25">18 - 25</option>
      <option value="25-30">25 - 30</option>
      <option value="30-40">30 - 40</option>
      <option value="40-50">40 - 50</option>
      <option value="50-">50 a viac</option>
    </select>
  </fieldset>
  ...

```

### ÚLOHA 12.26



Vo formulári z predchádzajúceho príkladu zoskupte pomocou elementu `fieldset` položky *Prevádzka, Vzdelanie, Prax, Pozícia a Jazyky*. Legendu uvádzať nemusíte.

## 12.4 Popisy prvkov formulára

Najjednoduchší spôsob popisovania formulárových prvkov je intuitívny: pomocou textu pred, resp. za elementom. Nevýhodou tohto spôsobu je, že príslušný formulárový element a jeho popis nie sú logicky nijako prepojené. Pre nevidiacich je takýto spôsob popisovania formulára nevhodný, pretože si nedokážu vytvoriť spojitosť medzi prvkom formulára a okolitým textom, najmä ak je text za prvkom formulára, prípadne je to dlhší text.

Pre popis prvku formulára, ktorý s ním bude jednoznačne spojený, slúži element `<label></label>`. Jeho použitie si ukážeme na príklade.



### PRÍKLAD 12.27

Vo formulári z predchádzajúcej úlohy definujeme popis položky *Meno* pomocou elementu `label`:

- Pre element `input`, ktorý definuje prvok formulára na zadanie mena, definujeme jednoznačný identifikátor `id="meno"`.
- Pôvodný popis prvku, text *Meno*, obalíme elementom `label`, ktorému pridáme atribút `for="meno"`, čím vytvoríme spojenie popisu s formulárovým prvkom.

```
...  
<legend>Osobné údaje</legend>  
<label for="meno">Meno:</label> <input type="text" name="meno"  
id="meno" size="20" required"><br>  
...
```

Popis prvku teda definujeme ako `<label for="id_prvku">popis</label>`, kde `id_prvku` je hodnotou `id` atribútu toho prvku formulára, ku ktorému patrí popis. Hodnota atribútu `id` môže byť rovnaká ako hodnota atribútu `name`. Vo všeobecnosti to odporúčame.



### PRÍKLAD 12.28

Pomocou elementu `label` vytvoríme popis výberovej ponuky *Vek* a prepínačov v časti *Prevádzka*:

```
<label for="vek">Vek:</label>  
<select name="vek" id="vek">  
  <option value="18-25">18 - 25</option>  
  <option value="25-30">25 - 30</option>  
  <option value="30-40">30 - 40</option>  
  <option value="40-50">40 - 50</option>  
  <option value="50-">50 a viac</option>  
</select>  
</fieldset>  
<br>  
<fieldset>  
  Prevádzka:  
  <input type="radio" name="mesto" id="ba" value="BA" required>  
  <label for="ba">Bratislava</label>  
  <input type="radio" name="mesto" id="bb" value="BB" required>  
  <label for="bb">Banská Bystrica</label>  
<br>
```

Všimnime si, že v časti *Prevádzka* nie je popisom text *Prevádzka*, ale text pri konkrétnych prepínačoch, teda Bratislava a Banská Bystrica, pretože každý prepínač je samostatný element formulára (na rozdiel od výberovej ponuky, ktorú celú zastrešuje jeden element `select`). Tiež si musíme uvedomiť, že hoci prepínače patriace do jednej skupiny majú rovnakú hodnotu atribútu `name`, hodnota atribútu `id` musí byť pre každý prepínač iná.



Element `label` nijako nemení vzhľad formulára, jeho prvkov, ani textov, ktoré tvoria popisy. Použitím elementu `label` tiež mierne uľahčíme prácu s jednotlivými prvkami formulára. Nemusíme už totiž klikať do konkrétneho prvku formulára (na prepínač, do textového poľa atď.), ale stačí kliknúť na popis prvku a efekt bude rovnaký, ako keby sme klikli na prvok formulára.

### ÚLOHA 12.29

Upravte popisy pomocou elementu `label` aj pre všetky ostatné prvky formulára (okrem tlačidiel) v súbore `prihlaska.html`.



### ZAPAMÄTAJTE SI

- aby sme mohli odoslať formulár, musí v ňom byť tlačidlo typu `submit`,
- každý prvok formulára musí mať definovaný atribút `name`,
- hodnota atribútu `name` prvkov formulára musí byť unikátna s výnimkou skupiny prepínačov (prepínače v rámci skupiny majú rovnakú hodnotu atribútu `name`),
- popisy formulárových elementov je vhodné definovať pomocou elementu `label`.



## 12.5 Formuláre a kaskádové štýly

Pomocou kaskádových štýlov môžeme ovplyvniť aj vzhľad formulárov, resp. jednotlivých formulárových elementov. Rovnako, ako pre iné elementy, môžeme definovať štýly pre elementy `input`, `select`, `textarea`, `button`, `form`, `fieldset`, `legend`, `label`. Môžeme im nastavovať farbu textu či pozadia, orámovanie, vonkajšie či vnútorné okraje, šírku,.... Môžeme tiež definovať štýl pre konkrétny element s využitím jeho atribútu `id`.

### PRÍKLAD 12.30

Na stránke `prihlaska.html` zmeníme orámovanie všetkých textových polí (jednoriadkových aj viacriadkových) a výberových ponúk na oranžovú (`#ff9900`). Tiež im nastavíme horný a dolný vonkajší okraj, čím jednotlivé prvky od seba odsunieme. Tlačidlám zmeníme farbu pozadia na `#ff9900` a taktiež pridáme horný a dolný vonkajší okraj (*obrázok 12.20 vľavo*). Do hlavičky stránky (elementu `head`) pridáme nasledujúcu definíciu štýlov:

```
<style>
  input[type=text], select, textarea {
    border: 1px solid #ff9900;
    margin: 5px 0;
  }
  button {
    background-color: #ff9900;
    margin: 5px 0;
    border: none;      /* zruší preddefinované orámovanie */
  }
</style>
```

Kódom `input[type=text]` hovoríme, že chceme definovať štýl len pre tie elementy `input`, ktoré majú atribút `type` nastavený na hodnotu `text`, teda pre jednoriadkové textové polia.



Podobne vieme definovať štýly len pre:

- odosielacie tlačidlá `input[type=submit]`,
- resetovacie tlačidlá `input[type=reset]`,
- obyčajné tlačidlá `input[type=button]`.



### ÚLOHA 12.31

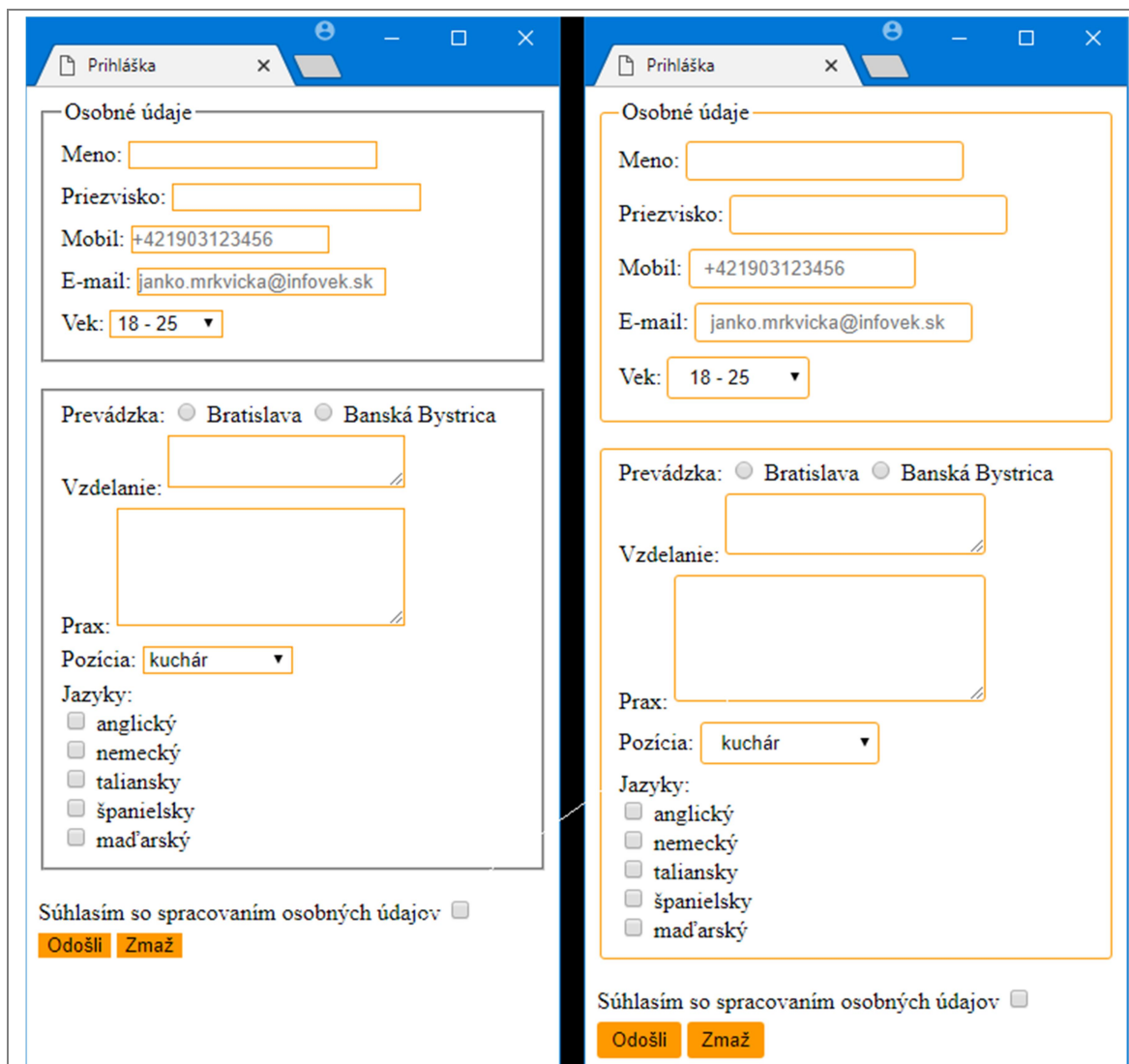
Do štýlov z predchádzajúceho príkladu:

- nastavte všetkým textovým poliam, výberovej ponuke, aj tlačidlám vnútorné okraje napr. na hodnotu `5px 10px`,
- nastavte rovnaké orámovanie, ako majú textové polia, aj pre element `fieldset`,
- experimentujte s hodnotami už definovaných vlastností,
- experimentujte s vlastnosťou `border-radius` pre element `button`, resp. pre iné elementy,
- experimentujte s inými nastaveniami, ktoré už poznáte.



### ÚLOHA 12.32

Vo formulári `rezervacia.html` upravte popisy všetkých formulárových prvkov pomocou elementu `label` a oštýlujte ju podľa vlastnej fantázie.



Obrázok 12.20 Vľavo prihláška po aplikácii štýlov z príkladu 12.30, vpravo prihláška po pridaní niektorých vlastností z úlohy 12.31.

## 12.6 Metodika pre učiteľa



### CIEĽ

Cieľom je, aby oboznámili sa s rôznymi typmi formulárových prvkov a ich spôsobom vkladania do HTML kódu.



### VÝKLAD

V príkladoch a úlohách, v ktorých sa tvorí formulár, sme nepoužívali (ale vysvetlili) atribút `method`. Ak by ste chceli testovať, aké údaje sa posielajú, môžete použiť metódu GET a vždy po odoslaní formulára sa pozrieť do riadka s adresou. Pri dlhšom formulári to však už bude dosť neprehľadné.

**K diskusii v úvode:** vyhľadávanie, prihlasovanie, registrácia, objednávanie/nákup, testy, ...

**Úloha 12.5:** Ide o to, aby si žiaci zistili, aké rôzne formulárové prvky sa vo formulároch môžu používať (textové políčka, prepínače, začiarkávacie políčka, výberová ponuka, rôzne druhy tlačidiel). Nezáleží na tom, ako žiaci tieto prvky pomenujú.

**Úloha 12.10:** Prepínače v druhej skupine musia mať iný atribút `name`, ako prepínače v prvej.

**Nadväznosť na predchádzajúce kapitoly:** Kapitola vyžaduje prebratie kapitol 1 – 8. Ak ste vôbec nevyučovali CSS, môžete vynechať časť 12.5.

#### Atribútu required

Kým neexistoval atribút `required`, musel takéto kontroly robiť vývojár buď ešte pred odoslaním formulára na strane klienta pomocou JavaScriptu, alebo až po odoslaní formulára na strane servera, v skripte, ktorý spracováva formulár.

**Obrázok 12.17:** Reakcie sa graficky líšia v závislosti od prehliadača.

Atribút `placeholder` Placeholder možno definovať aj k typom `search`, `url`, `tel`, `email` (z HTML5), ale tie sme v učebnici neuvádzali.

#### Doplňujúce úlohy

**Úloha:** K stránkam učebnice (z úlohy 11.16) pridajte stránku s objednávkovým formulárom na učebnicu. Premyslite, čo by takýto formulár mal obsahovať. Doplnite odkaz na stránku s formulárom aj do navigácie.

**Úloha:** Rozdeľte stránku IT Pizza na dve podstránky, v jednej bude ponuka a v druhej galéria. Ostatné časti (hlavička, navigácia, päta s kontaktmi) by mali zostať na oboch stránkach. Upravte navigáciu tak, aby obsahovala odkazy na stránky `ponuka`, `galéria`, `prihlaska.html` a `rezervacia.html`. Stránky `prihlaska.html` a `rezervacia.html` upravte tak, aby mali rovnaký vzhľad ako stránky IT Pizza, t.j. pridajte hlavičku, navigáciu a pätu, pripojte potrebné štýly.