



Formuláře

Klára Pešková, Klara.Peskova@mff.cuni.cz
Katedra softwaru a výuky informatiky, MFF UK
Základy tvorby webu, ZS 2024/25

HTML formulář

- formuláře obsahují prvky uživatelského rozhraní

```
<form action="skript.php">  
  <label for="name">Name:</label>  
  <input type="text" id="name">  
  
  <label for="pwd">Password:</label>  
  <input type="password" id="pwd">  
  
  <input type="submit" value="OK">  
</form>
```

Name:

Password:

OK

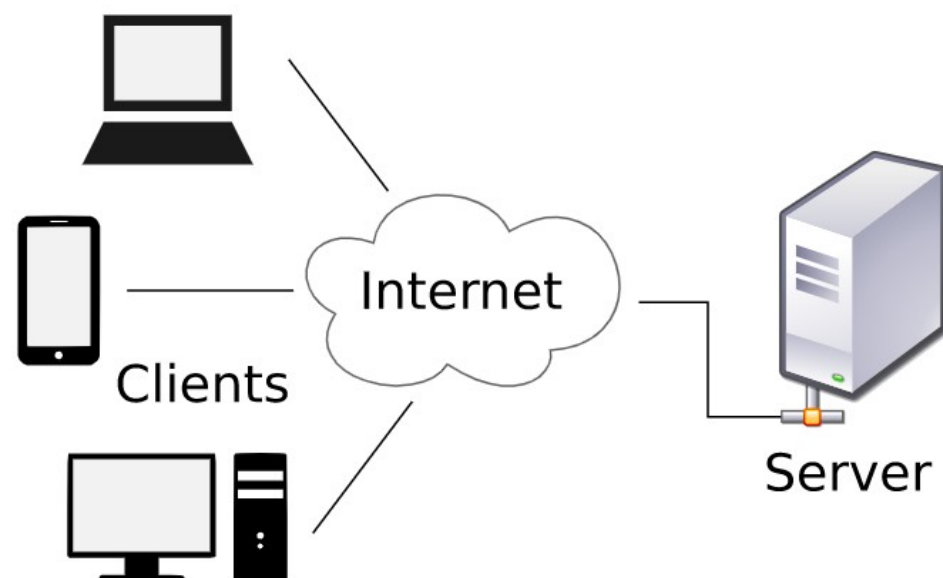
Name:

Password:

OK

Zpracování formuláře

- Uživatel vyplní formulář
- Kontrola dat (nepovinné)
- Data se odešlou na server ke zpracování
- Předzpracování a kontrola dat na serveru
- Server zpracuje data
- Server pošle prohlížeči odpověď (jako HTML)



Dnešní témata

- HTML formulářové prvky
- (CSS)
- Validace
- Cookies v javascriptu
- (Zpracování na serveru)

Prvky formuláře

- `<input>`
 - Základní prvky pro zadávání vstupu
 - Různé typy políček pro vstup (určuje se pomocí atributů)
- `<textarea>`
 - Více-řádkový text
- `<select>`
 - Výběr se seznamu `<options>`
- `<button>`
 - Tlačítko (např. odeslání, vymazání formuláře)

Základní formulářové prvky - `<input>`

```
<form>
```

```
<input type="text">
<input type="radio" name="group"
  checked>
<input type="checkbox" checked>
<input type="submit" value="OK">


<input type="hidden">
<input type="file">

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

```
</form>
```

checked, disabled, required,
maxwidth, size="počet_znaků"

Name:

Password  Please fill out this field.

- male
 female
 other

I came here to study:

- HTML
 CSS
 JavaScript
 PHP
 Flash

No file chosen

HTML5 input elementy

```
<form>
```

```
<input type="color">
```

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

```
<input type="datetime-local">
```

```
<input type="month">
```

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

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

```
<input type="range">
```

```
</form>
```

- Pokud zadaný typ (`type`) inputu není podporovaný prohlížečem, políčko se chová jako obyčejné zadávací pole typu `text`

HTML5 input elements - příklady

Date picker

Date and time picker

Month picker

December 2020							↑	↓
Mo	Tu	We	Th	Fr	Sa	Su		
30	1	2	3	4	5	6		
7	8	9	10	11	12	13		
14	15	16	17	18	19	20		
21	22	23	24	25	26	27		
28	29	30	31	1	2	3		
4	5	6	7	8	9	10		

Today

Color picker

Color picker interface showing a color gradient and RGB values. The selected color is yellow, with RGB values 255, 226, and 36.

E-mail

Please include an '@' in the email address. 'klara.cz' is missing an '@'.

Number

Range

OK

Další formulářové prvky

```
<select>
  <option>Ford</option>
  <option selected>Škoda</option>
  <option>Mercedes</option>
</select>
```

```
<textarea rows="3" cols="10">
</textarea>
```

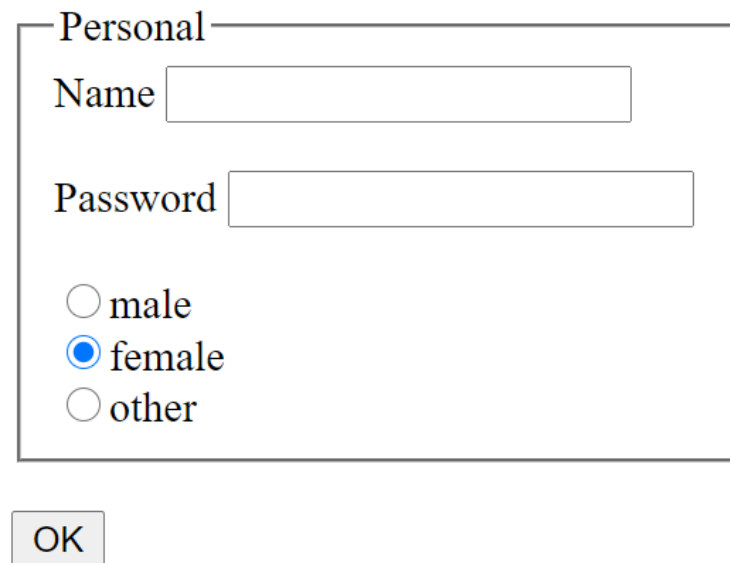
```
<input list="cars">
<datalist id="cars">
  <option>Ford</option>
  <option>Škoda</option>
  <option>Mercedes</option>
</datalist>
```

- Ford
- Škoda
- Mercedes

Další vylepšení formuláře

```
<label for="id_prvku">Popisek</label>
```

```
<fieldset>  
  <legend>Název oddílu</legend>  
</fieldset>
```



Personal

Name

Password

male
 female
 other

OK

- Atributy formulářových prvků:

- `autocomplete="on"/"off"`

- pro celý formulář nebo pro jednotlivé prvky

- `placeholder`

- `title` - zobrazí "bublínku" s hodnotou tohoto atributu (popisek)

Formuláře – CSS styly

- Selektory jako pro ostatní tagy
- Rozlišení typů pro `input`

- selektor `input[type=text]`

- `width, padding (box-sizing: border-box)`

- Aktivní prvek

- `input[type=text]:focus`

- Ikonka ve vyhledávacím poli:

- `background-image: url('icon.png');`

- `background-position: 10px 10px;`

- `background-repeat: no-repeat;`

Jméno

Klára

Příjmení



Search..

CSS - pseudoselektory

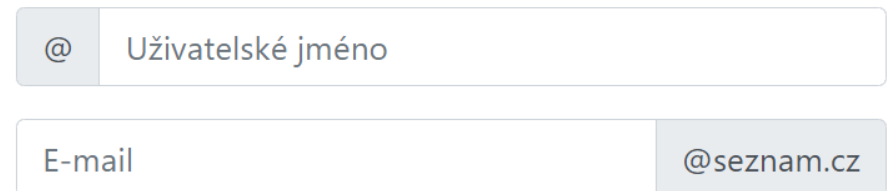
- `:disabled` input elementy s atributem `"disabled"`
- `:valid` input elementy s validními hodnotami
- `:invalid` input elementy s neplatnými hodnotami
- `:required` input elementy s atributem `"required"`
- `:optional` input elementy, které nemají atribut `"required"`

Responsivní formulář

- např. pomocí flex boxů...

Formuláře a Bootstrap

- Základní rozložení
 - **stacked** (základní) – prvky naskládáné nad sebou
 - "tabulkový" model
- Obsahuje styly pro input, textarea, checkbox, radio, select
 - Třídy pro prvky formuláře
 - `form-control`, `form-check-input` (checkbox, radio)
 - Input groups
- Třídy pro validaci formulářů



The image shows two examples of Bootstrap form input groups. The first example is a single input field with a light gray background and rounded corners. On the left side, there is a small gray square containing an '@' symbol, followed by the text 'Uživatelské jméno'. The second example is a similar input field, but it is split into two parts. The left part contains the text 'E-mail', and the right part is a small gray square containing '@seznam.cz'.



Validace formuláře

Kontrola (validace) formuláře

- Jsou vyplněna všechna povinná pole?
- Jsou v nich validní data?
 - např. číslo tam, kde má být číslo
- Klient-side
 - HTML5 - automatická kontrola, speciální atributy
 - vzhled pomocí CSS stylů
 - JavaScript
 - (Bootstrap)
- Server-side
 - kontrola při zpracování na serveru (např. PHP, Python...)

Kontrola formuláře HTML5

- atribut `required` - formulář není možné odeslat bez vyplnění pole
- HTML5 typy inputu kontrolují svůj obsah,
 - např. `type="email"` musí obsahovat zavináč a tečku
 - `type="url"` musí obsahovat protokol a tečku
 - do `type="number"` nejde zadat nic jiného než číslo
- atribut `pattern` - používá JavaScriptové regulární výrazy
 - např. `pattern="[A-Za-z]{3}"` - tři písmena

CSS styly - validace formuláře

- selektory pro povinná pole
 - `input:required:valid` / `input:required:invalid`
- př. obrázek v input poli

```
input:required{
    background-size: contain;
    background-position: right top;
    background-repeat: no-repeat;
}
input:required:invalid {
    background-image: url(invalid.png);
}

input:required:valid {
    background-image: url(valid.png);
}
```

Kontrola formuláře pomocí JavaScriptu

- atributu formuláře `onsubmit` přiřadíme javascriptovou funkci

```
onsubmit="return zkontrolujFormular()"
```

- pokud funkce vrátí `false`, formulář se neodešle
- v případě chyby typicky zobrazíme `alert`

```
function zkontrolujFormular() {  
  let x = document.forms["formular"]["jm"].value;  
  if (x == "") {  
    alert("Vyplňte, prosím, jméno.");  
    return false;  
  }  
}
```

Kontrola formuláře - Bootstrap

- třídy formuláře:
 - `.was-validated` - validace před odesláním formuláře
 - `.needs-validation` - po odeslání formuláře
- `.valid-feedback`, `.invalid-feedback`

```
<div>
```

```
  <input type="text" class="form-control" required>
```

```
  <div class="valid-feedback">OK</div>
```

```
  <div class="invalid-feedback">Vyplňte toto pole.</div>
```

```
</div>
```



Zpracování formuláře

HTML formuláře - `<form>`

- Atributy
 - `method` – HTTP metoda, která se použije při přenosu dat (GET/POST)
 - `post` – data jsou předána v těle HTTP requestu
 - `get` – data jsou zakódována v URL (v části za otazníkem) – jsou viditelná pro všechny
 - `action` – URL, na které se data odesílají (skript, který zpracuje formulář)

```
<form action="skript.php" method="get">  
  . . .  
</form>
```

GET vs. POST

```
<form action="skript.php" method="get/post">
```

- GET
 - Data v url, páry name=value oddělené &
 - Posílaná data jsou vidět (včetně hesel)
 - Délka url je omezená na 2048 znaků
 - Uživatel si může uložit výsledek
 - Hodí se pro ladění
- POST
 - Data z formuláře jsou uvnitř HTTP požadavku
=> data nejsou vidět v url
 - Nemá omezenou velikost dat

Odesílání formulářů

- Po odeslání formuláře jsou data zakódována do HTTP požadavku s daným URL a zvolenou metodou (GET/POST)
- GET: Odesílací tlačítko funguje jako odkaz s nově vytvořeným URL (odpovídajícím hodnotám vyplněných ve formuláři)

```
<form action="script.php" method="get">  
  <input type="text" name="name">  
  <input type="password" name="pwd">  
</form>
```

name

value (nebo to, co uživatel vyplní)

odeslání dat na server

=> URL: `script.php?name=John&pwd=sokol`

Odesílání formulářů - poznámky

- Speciální znaky, jako např. mezery se v URL zakódují.
- Pokud je formulář odeslán metodou GET, všechna vyplněná data (včetně hesla) jsou v URL viditelná.
- Jako tlačítko pro odeslání formuláře je místo elementu `<input>` typu `submit` možné použít párový element `<button>`, obsah tagu je potom použit jako text na tlačítku (výhoda: oddělení hodnoty atributu `value` a textu tlačítka).
- V HTML5 může být element `<button>` umístěn i mimo formulář, potom je s formulářem propojen pomocí atributu `form`.

Zpracování dat na serveru (pomocí php)

- php skript na serveru dostane data v asociativním poli `$_GET`, `$_POST` nebo `$_REQUEST`
- Např.
 - `$_GET["jmeno"] = "Jan"`
 - `$_GET["heslo"] = "45ak!"`(v php záleží na malých a velkých písmenech)
- Data je možné použít např. pro
 - sestavení obsahu výsledné stránky
 - sestavení dotazu a vyhledání informací v databázi
 - uložení dat do databáze
 - odeslání e-mailu

Validace formulářů na serveru

- Kontrolu dat je možné provádět až na serveru
 - Prázdná pole
 - Typ dat
 - Speciální znaky, lomítka

Zpracování dat na serveru – přesměrování

- Formulář – skript, který zpracuje formulář – stránka s výsledkem

