COMP519 Web Programming
Lecture 9: HTML (HTML5 Elements: Part 3)
Handouts

Ullrich Hustadt

Department of Computer Science
School of Electrical Engineering, Electronics, and Computer Science
University of Liverpool
Contents

1 HTML Elements for the Body
   Forms
   Form Controls

2 Further Reading
Forms

• A form is an element that contains form controls, such as text fields, buttons, checkboxes, range controls, or color pickers

• Forms allow users to enter data which can then be sent to a web server for processing

• A form element has several (optional) attributes, including
  • action: URL to use for form submission
  • method: HTTP method to use for form submission (get or post)
  • enctype: encoding type to use for form submission
  • novalidate: form is not validated during submission

```
<form action="https://sam.csc.liv.ac.uk/COMP/Calendar.pl"
      method="post" enctype="text/plain">
</form>
```

• A form can be submitted and on submission the data entered via the form controls will be send to the URL in action using the HTTP request method in method with encoding type enctype
Forms

• A form is an element that contains form controls, such as text fields, buttons, checkboxes, range controls, or color pickers.

• HTML5 introduced a number of additional form controls and attributes for all form controls, but browser support is partial.

→ Always test your forms with a range of browsers (Apple Safari, Google Chrome, Mozilla Firefox, MS IE, MS Edge)
Label

- In order for a form to be usable, each form control should be accompanied by an indication of what it is for or how it should be used. Example: ‘Surname’ or ‘Enter your surname’ next to a field into which you are meant to enter your surname.

- A label element represents such an indication (a caption).

- A label element can be associated with a specific form control either using its `for` attribute or by putting the form control inside the label element itself.

```html
<label for="s1">Surname:</label>
<input name="surname" id="s1" type="text">

<label>First name(s): <input name="first" id="f1" type="text"></label>
```
**Input**

- The **input element** represents a ‘field’ that allows the user to enter data of a certain type.
- The type attribute of an **input element** determines what type of data can be entered and in what form.

<table>
<thead>
<tr>
<th>Value of type</th>
<th>Data type</th>
<th>Form control</th>
</tr>
</thead>
<tbody>
<tr>
<td>text</td>
<td>Text with no line breaks</td>
<td>Text field</td>
</tr>
<tr>
<td>tel</td>
<td>Phone number</td>
<td>Text field</td>
</tr>
<tr>
<td>date</td>
<td>Date (Year/Month/Day)</td>
<td>Date control</td>
</tr>
<tr>
<td>email</td>
<td>E-mail address</td>
<td>Text field</td>
</tr>
<tr>
<td>file</td>
<td>Zero or more files</td>
<td>Button and file selector</td>
</tr>
<tr>
<td>color</td>
<td>sRGB color</td>
<td>Button and color picker</td>
</tr>
<tr>
<td>number</td>
<td>Floating-point number</td>
<td>Text field or spinner</td>
</tr>
<tr>
<td>password</td>
<td>Password</td>
<td>Text field with obscured input</td>
</tr>
</tbody>
</table>
Input

- The **input element** represents a ‘field’ that allows the user to enter data of a certain type
- The **type** attribute of an **input element** determine what type of data can be entered and in what form

<table>
<thead>
<tr>
<th>Value of type</th>
<th>Data type</th>
<th>Form control</th>
</tr>
</thead>
<tbody>
<tr>
<td>checkbox</td>
<td>A set of zero or more values from a predefined list</td>
<td>Checkbox</td>
</tr>
<tr>
<td>radio</td>
<td>An enumerated value</td>
<td>Radio button</td>
</tr>
<tr>
<td>button</td>
<td>Initiates form submission</td>
<td>Button</td>
</tr>
<tr>
<td>submit</td>
<td>Resets form</td>
<td>Button</td>
</tr>
<tr>
<td>reset</td>
<td></td>
<td>Button</td>
</tr>
</tbody>
</table>

- Depending on the value of the type attribute, an **input element** will have additional attributes that define its behaviour
**Input**

- The **input element** represents a ‘field’ that allows the user to enter data of a certain type.

- Depending on the value of the type attribute, an **input element** will have additional attributes that define its behaviour.

- **Common attributes include**
  - **id**: unique id used to identify the element with the document.
  - **name**: (unique) name used by the form processor to access input.
  - **autofocus**: automatically focus on this form control when the page is loaded.
  - **disabled**: whether the form control is disabled.
  - **required**: whether the form control is required to have a non-empty value for form submission.

```html
<input name="studentid" id="sid" type="number"
min="190000000" max="999999999">
<input type="submit">
```
HTML Elements for the Body

**Input**

```html
<form action="process.php">
    <label for="s1">Surname: </label>
    <input name="surname" id="s1" type="text">

    <label>First name: <input name="first" id="f1" type="text"></label>

    <label>StudentID: <input name="studentid" id="sid" type="number" min="190000000" max="999999999"></label>

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

- On submission, the web browser will construct pairs `name=value` where `name` is the value of the `name` attribute of one of the form controls and `value` is the input by the user for that form control.

- A string composed of those pairs will be send to `process.php`

**Example:**
Peters, Amy Lee, and 201612345 are entered into the three fields

```
surname=Peters, first=Amy+Lee, and studentid=201612345
```

are sent to `process.php`
HTML Elements for the Body

**Input**

```html
<form action="process.php">
  <label for="s1">Surname: </label>
  <input name="surname" id="s1" type="text">
  <label>First name: <input name="first" id="f1" type="text">
  </label>
  <label>StudentID: <input name="studentid" id="sid" type="number" min="190000000" max="999999999"></label>
  <input type="submit">
</form>

• This form can be submitted by either activating the ‘Submit’ button or by pressing the return key within one of the three input fields

• Form submission is possible even with all input fields being empty, even the one for the StudentID
  ⇨ we need to add the **required attribute** to prevent empty inputs

• The StudentID field accepts floating point numbers between the specified min and max values
  ⇨ 1.9e8 is accepted and studentid=1.9e8 send to process.php

```
Names versus IDs

```html
<p>Please complete the following form if you are an undergrad:</p>
<form id="formUG" action="process-comments.php">
  <input type="hidden" name="level" value="UG">
  <label for="c1">Describe your problem:</label>
  <input type="text" name="comment" id="c1" max_length="250" required>
  <br/>
  <input type="submit">
</form>

<p>Please complete the following form if you are a postgrad:</p>
<form id="formPG" action="process-comments.php">
  <input type="hidden" name="level" value="PG">
  <label for="c2">Describe your problem:</label>
  <input type="text" name="comment" id="c2" max_length="250" required>
  <br/>
  <input type="submit">
</form>

<p>Help:<br/>
If you are an undergraduate student,<br/>complete <a href="#c1">the first form</a>.<br/>
If you are a postgraduate student,<br/>complete <a href="#c2">the second form</a>.</p>
```
Names versus IDs

Please complete the following form if you are an undergrad:

Describe your problem:  

Please complete the following form if you are a postgrad:

Describe your problem: 

Help:
If you are an undergraduate student, complete the first form.  
If you are a postgraduate student, complete the second form.

• Entering ‘None’ into the first form and activating the submit button sends level=UG and comment=None to process-comments.php

• Activating the hyperlink underlying ‘the first form’ focuses input on the form with id c1
Names versus IDs

Please complete the following form if you are an undergrad:

```
<form id="formUG" action="process-comments.php">
    <label for="c1">Describe your problem:</label>
    <input type="text" name="comment" id="c1" maxlength="250" required>
    <input type="submit">
</form>
```

Please complete the following form if you are a postgrad:

```
<form id="formPG" action="process-comments.php">
    <label for="c2">Describe your problem:</label>
    <input type="text" name="comment" id="c2" maxlength="250" required>
    <input type="submit">
</form>
```

Help:
If you are an undergraduate student, complete the first form. If you are a postgraduate student, complete the second form.

- Entering ‘None’ into the second form and activating the submit button sends level=PG and comment=None to process-comments.php
- Activating the hyperlink underlying ‘the second form’ focuses input on the form with id c2

name and id serve different purposes
still, programmers often use the same value for both unless this is not possible (as in the example above)
Input: Submit

- An **input element** with type `submit` represents a button that, when activated, submits the form it is associated with.

- Attributes include:
  - `value`: replaces the default label of the button
  - `name`: (unique) name used by the form processor to access input
  - `formaction`: Overwrite the corresponding attribute of the form
  - `formenctype`: attributes of
  - `formnovalidate`: the form

```html
<input type="submit">
<input type="submit" name="action" value="Submit essay">
<input type="submit" formaction="save.php">
```
Input: Submit (Example 1)

```
<form action="process.php" method="post">
  <label> Name:  <input name="fn" required></label>
  <label> Essay:  <textarea name="essay" required></textarea></label>
  <input type="submit" name="action" value="Submit essay">
  <input type="submit" name="action" value="Save essay" formnovalidate>
</form>
```

- On activating the ‘Submit essay’ button it will be checked that both Name and Essay have been filled in; if so, the text entered will be sent to process.php together with action=Submit+essay
- On activating the ‘Save essay’ button no check will take place; any text entered for Name and Essay will be sent to process.php together with action=Save+essay
- The script process.php can perform the appropriate operations depending on the value of action
Input: Submit (Example 2)

```html
<form>
  <label>Name:  <input name="fn" required></label>
  <label>Essay:  <textarea name="essay" required></textarea></label>
  <input type="submit" formaction="submit.php" value="Submit essay">
  <input type="submit" formaction="save.php" value="Save essay" formnovalidate>
</form>
```

- On activating the ‘Submit essay’ button it will be checked that both Name and Essay have been filled in; if so, the text entered will be send to submit.php
- On activating the ‘Save essay’ button no check will take place; any text entered for Name and Essay will be send to save.php
- The scripts submit.php and save.php have been written specifically to perform the respective function
Input: Number

- An input element with type number represents a one line plain text field for the element’s value.
- Attributes include:
  - value: shown as ‘default’ value in the text field
  - readonly: the value cannot be changed
  - min: minimum value allowed to be entered
  - max: maximum value allowed to be entered
  - step: granularity that is expected (and required) of the value
  - placeholder: a short hint describing the expected value of a text area (disappears as soon as the user enters a value)

```html
<!-- The only allowed values below are 10, 12, 14 -->
<input type="number" name="quantity" min="10" max="14" step="2" required>

<!-- The only allowed values below are 0.0 to 1.0 in steps of 0.1 -->
<input type="number" name="fraction" min="0" max="1" step="0.1" required>

<input type="number" name="age" placeholder="Enter your age">
<input type="number" name="studentid" min="190000000" max="999999999">
```
Input: Text

- An **input element** with type `text` represents a one line plain text field for the element’s value.

- Attributes include:
  - **value**: shown as ‘default’ value in the text field
  - **readonly**: the value cannot be changed
  - **size**: visible width of the field in characters
  - **minlength**: minimum number of characters allowed to be entered
  - **maxlength**: maximum number of characters allowed to be entered
  - **placeholder**: a short hint describing the expected value of a text area (disappears as soon as the user enters a value)
  - **pattern**: a regular expression that the value has to match

```html
<input type="text" name="surname" size="100" required>
<input type="text" name="department" value="Computer Science" readonly>
<input type="text" name="studentid" minlength="9" maxlength="9">
```
**Input: Password**

- An *input element* with type *password* represents a one line plain text field for the element’s value.
- Same as an *input element* with type *text*, except that the web browser should obscure the value that is being entered.
- Has the same attributes as *input element* with type *text*.
Input: Checkbox

- An **input element** with type checkbox represents a two-state control (checkbox)

- The return value of a checked checkbox is **on**, an unchecked checkbox returns nothing

- Attributes include
  - `value`: replaces the ‘default’ return value
  - `checked`: this element is selected by default

Most of the other attributes, for example, `readonly` are not available

```
<form action="process.php">
  <p>What fruits do you like?</p>
  <label>Apples <input type="checkbox" name="fruit[]" value="a"></label>
  <label>Oranges <input type="checkbox" name="fruit[]" value="o"></label>
  <label>Peaches <input type="checkbox" name="fruit[]" value="p"></label>
  <input type="submit">
</form>
```

fruit[] is an array storing a subset of {"a","o","p"} depending on which checkboxes have been checked
Input: Radio

- A set of input elements with type radio represents a radio button group in which only one form control can be selected/set to true.
- For input elements to belong to the same radio button group:
  - they must be associated with the same form and
  - their name-attributes must have the same value.
- Attributes include:
  - value: replaces the ‘default’ return value on
  - checked: this element is selected by default.

Most of the other attributes, for example, readonly are not available.

```html
<form action="process.php">
  <p>Please select your preferred contact method:</p>
  <input type="radio" id="cc1" name="contact" value="email">
  <label for="cc1">Email</label>
  <input type="radio" id="cc2" name="contact" value="phone">
  <label for="cc2">Phone</label>
  <input type="radio" id="cc3" name="contact" value="mail">
  <label for="cc3">Mail</label>
  <input type="submit">
</form>
```
Input: Radio

- If no radio button within a group is selected when the associated form is submitted, then no value for the group is send to the server
  - if the user must make a selection,
    then set the required attribute for at least one element

- If a radio button within a group is selected when the associated form is submitted, but the value-attribute of that input element is not set, then the default value on will be returned
  - make sure that all input-elements within a radio button group have a non-empty value-attribute
Input: Radio

```html
<form action="process.php">
  <p>Please select your preferred contact method:</p>
  <input type="radio" id="cc1" name="contact" value="email">
  <label for="cc1">Email</label>
  <input type="radio" id="cc2" name="contact" value="phone">
  <label for="cc2">Phone</label>
  <input type="radio" id="cc3" name="contact" value="mail">
  <label for="cc3">Mail</label>
  <input type="submit">
</form>
```

Please select your preferred contact method:

- Email
- Phone
- Mail

- No radio button is selected and no selection is required, so if we activate the submit button right away, no values will be send to process.php.
- If we want to pre-select a button, then we should set the checked attribute for one of the elements.
- If we select the first button and then activate the button, then contact=email will be send to process.php.
Textarea

- A **textarea element** represents a multi-line text input control
- A **textarea element** can have several attributes, including
  - **id**: unique id used to identify the element with the document
  - **name**: (unique) name used by the form processor to access input
  - **cols**: the visible width of a text area (number)
  - **rows**: the visible height of a text areas (number)
  - **maxlength**: maximum number of characters that can be entered
  - **required**: specifies that a text area must be filled out
  - **wrap**: specifies whether newlines are preserved
    - "hard" ↆ yes / "soft" ↆ no

```html
<textarea name="problem1" id="p1" cols="10" rows="4" maxlength="35" required wrap="hard">Describe your problem</textarea>
<textarea name="problem2" id="p2" cols="10" rows="4" wrap="soft" placeholder="Describe your problem"></textarea>
```
Select

- A **select element** represents a drop-down menu with pre-defined options between which the user must select.
- The content of a **select element** consists of a list of **option elements** that represent those options.
- A **select-element** can have several attributes, including:
  - **id**: unique id used to identify the element with the document
  - **name**: (unique) name used by the form processor to access input
  - **multiple**: allow multiple options to be selected
  - **required**: an option must be selected that has a non-empty value
  - **disabled**: the current selection can not be changed
  - **size**: number of options to show to the user
- An **option element** can have several attributes, including:
  - **label**: the label used in the drop-down menu
  - **value**: the value returned for the option
  - **selected**: the option is selected by default
  - **disabled**: the option is shown but cannot be selected
Select

- A **select element** represents a **drop-down menu** with pre-defined options between which the user must select (often preferred over a radio button group)
- The content of a **select element** consists of a list of **option elements** that represent those options

```html
<label for="module">Select a module:</label>
<select name="module">
    <option value="COMP517">
        COMP517: Object Oriented Software Development
    </option>
    <option value="COMP518">
        COMP518: Database and Information Systems
    </option>
    <option value="COMP519">
        COMP519: Web Programming
    </option>
</select>
```
Select

```html
<form action="process.php">
  <label for="module">Select a module:</label>
  <select name="module">
    <option value="COMP517">COMP517: Object Oriented Software Development</option>
    <option value="COMP518">COMP518: Database and Information Systems</option>
    <option value="COMP519">COMP519: Web Programming</option>
  </select>
  <input type="submit"/>
</form>
```

- By default, the first option is selected
- If the selection is not changed and the user activates the submit button, then module=COMP517 is sent to process.php
- In general, the value associated with the selected option will be send
Select

```html
<form action="process.php">
  <label for="module">Select a module:</label>
  <select name="module">
    <option value="COMP517">
      COMP517: Object Oriented Software Development
    </option>
    <option value="COMP518" selected>
      COMP518: Database and Information Systems
    </option>
    <option value="COMP519">
      COMP519: Web Programming
    </option>
  </select>
  <input type="submit"/>
</form>
```

- Adding the attribute `selected` to the second option, makes it the option that is selected by default.
- If the selection is not changed and the user activates the submit button, then `module=COMP518` is sent to `process.php`
• That an option with a non-empty value is pre-selected is often not desirable — the user does not need to make a conscious choice

• Adding a default option with empty value and adding the attribute `required` to the `select` element forces the user to make a conscious choice
Revision and Further Reading

Read

- Chapter 9: Forms
- Chapter 19: More CSS Techniques (Styling Forms)

of


E-book https://library.liv.ac.uk/record=b5647021