CLIENT-SIDE

WEB APPS II

FORMS

WEB STORAGE

VERSION

v1.0 16 November 2021

ACKNOWLEDGMENTS

THANKS TO:

- Hamzeh Roumani, who has shaped EECS-4413 into a leading hands-on CS course at EECS and who generously shared all of his course materials and, more importantly, his teaching philosophy with me;
- Parke Godfrey, my long-suffering Master's supervisor and mentor; and
- Suprakash Datta for giving me this opportunity to teach this course.

PRINTABLE VERSION OF THE TALK

Download PDF

FORMS

HTML FORMS

An HTML form is used to collect user input. The user input is most often sent to a server for processing.

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>
```

THE <input> ELEMENT

The HTML `<input>` element is the most used form element. An `<input>` element can be displayed in many ways, depending on the type attribute. Here are some examples:

Type	Description
`<input type="text">`	Displays a single-line text input field
`<input type="radio">`	Displays a radio button (for selecting one of many choices)
`<input type="checkbox">`	Displays a checkbox (for selecting zero or more of many choices)
`<input type="submit">`	Displays a submit button (for submitting the form)
`<input type="button">`	Displays a clickable button

MORE INPUT TYPES:

- `color`
- `date`
- `datetime-local`
- `email`
- `file`
- `hidden`
- `image`
- `month`
- `number`
- `password`
- `range`
- `reset`
- `search`
- `tel`
- `text`
- `time`
- `url`
- `week`

MORE FORM ELEMENTS

Tag	Description
`<form>`	Defines an HTML form for user input
`<input>`	Defines an input control
`<textarea>`	Defines a multiline input control (text area)
`<label>`	Defines a label for an `<input>` element
`<fieldset>`	Groups related elements in a form
`<legend>`	Defines a caption for a `<fieldset>` element
`<select>`	Defines a drop-down list
`<optgroup>`	Defines a group of related options in a drop-down list
`<option>`	Defines an option in a drop-down list
`<button>`	Defines a clickable button
`<datalist>`	Specifies a list of pre-defined options for input controls
`<output>`	Defines the result of a calculation

```
<label for="cars">Choose a car:</label>
<select id="cars" name="cars">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

```
<textarea name="message" style="width:200px; height:600px;">
The cat was playing in the garden.
</textarea>
```

```
<button type="button" onclick="alert('Hello World!')">
  Click Me!
</button>
```

MORE FORM ELEMENTS

Tag	Description
`<form>`	Defines an HTML form for user input
`<input>`	Defines an input control
`<textarea>`	Defines a multiline input control (text area)
`<label>`	Defines a label for an `<input>` element
`<fieldset>`	Groups related elements in a form
`<legend>`	Defines a caption for a `<fieldset>` element
`<select>`	Defines a drop-down list
`<optgroup>`	Defines a group of related options in a drop-down list
`<option>`	Defines an option in a drop-down list
`<button>`	Defines a clickable button
`<datalist>`	Specifies a list of pre-defined options for input controls
`<output>`	Defines the result of a calculation

```
<form action="/action_page.php">
  <fieldset>
    <legend>Personalia:</legend>
    <label for="fname">First name:</label><br>
    <input type="text" id="fname" name="fname" value="John"><br>
    <label for="lname">Last name:</label><br>
    <input type="text" id="lname" name="lname" value="Doe"><br><br
    <input type="submit" value="Submit">
  </fieldset>
</form>
```

```
<form action="/action_page.php">
  <input list="browsers">
  <datalist id="browsers">
    <option value="Internet Explorer">
    <option value="Firefox">
    <option value="Chrome">
    <option value="Opera">
    <option value="Safari">
  </datalist>
</form>
```

FORM VALIDATION

DATA VALIDATION

Data validation is the process of ensuring that user input is clean, correct, and useful.

Typical validation tasks are:

- Has the user filled in all required fields?
- Has the user entered a valid date?
- Has the user entered text in a numeric field?
- Most often, the purpose of data validation is to ensure correct user input.

Validation can be defined by many different methods, and deployed in many different ways. Server side validation is performed by a web server, after input has been sent to the server. Client side validation is performed by a web browser, before input is sent to a web server.

```
<form name="myForm" method="post"
      action="/action_page.php"
      onsubmit="return validateForm()">
  <label for="fname">First name:</label><br>
  <input type="text" name="fname">
  <input type="submit" value="Submit">
</form>
```

```
function validateForm() {
  let x = document.forms["myForm"]["fname"].value;
  if (x == "") {
    alert("Name must be filled out");
    return false;
  }
}
```

FORM VALIDATION

HTML CONSTRAINT VALIDATION

HTML5 introduced a new HTML validation concept called constraint validation.

Attribute	Description
disabled	Specifies that the input element should be disabled
max	Specifies the maximum value of an input element
min	Specifies the minimum value of an input element
pattern	Specifies the value pattern of an input element
required	Specifies that the input field requires an element
type	Specifies the type of an input element

```
<form action="/action_page.php" method="post">
  <label for="fname">First name:</label><br>
  <input type="text" name="fname" required>
  <input type="submit" value="Submit">
</form>
```

WEB STORAGE

WEB STORAGE API

The Web Storage API provides mechanisms by which browsers can store key/value pairs, in a much more intuitive fashion than using cookies. The two mechanisms within Web Storage are as follows:

- `sessionStorage` maintains a separate storage area for each given origin that's available for the duration of the page session (as long as the browser is open, including page reloads and restores)
  - Stores data only for a session, meaning that the data is stored until the browser (or tab) is closed.
  - Data is never transferred to the server.
  - Storage limit is larger than a cookie (at most 5MB).
- `localStorage` does the same thing, but persists even when the browser is closed and reopened.
  - Stores data with no expiration date, and gets cleared only through JavaScript, or clearing the Browser cache / Locally Stored Data.
  - Storage limit is the maximum amongst the two.

These mechanisms are available via the `window.sessionStorage` and `window.localStorage` properties (to be more precise, in supporting browsers the `Window` object implements the `WindowLocalStorage` and `WindowSessionStorage` objects, which the `localStorage` and `sessionStorage` properties hang off) — invoking one of these will create an instance of the `Storage` object, through which data items can be set, retrieved and removed. A different `Storage` object is used for the `sessionStorage` and `localStorage` for each origin — they function and are controlled separately.

WEB STORAGE EXAMPLES

```
1  // Store
2  localStorage.setItem("lastname", "Smith");
3  localStorage.lastname = "Smith"; // alternative
4
5  // Retrieve
6  document.getElementById("result").innerHTML = localStorage.getItem("lastname");
7  document.getElementById("result").innerHTML = localStorage.lastname; // alternative
8
9  // Remove
10 localStorage.removeItem("lastname");
```

WEB STORAGE EXAMPLES

USING localStorage

```
1 if (localStorage.clickcount) {
2   localStorage.clickcount = (+localStorage.clickcount) + 1;
3 } else {
4   localStorage.clickcount = 1;
5 }
6 document.getElementById("result").innerHTML = "You have clicked the button "
7                                               + localStorage.clickcount
8                                               + " time(s).";
```

WEB STORAGE EXAMPLES

USING sessionStorage

```
1 if (sessionStorage.clickcount) {
2   sessionStorage.clickcount = (+sessionStorage.clickcount) + 1;
3 } else {
4   sessionStorage.clickcount = 1;
5 }
6 document.getElementById("result").innerHTML = "You have clicked the button "
7                                               + sessionStorage.clickcount
8                                               + " time(s) in this session.";
```

This slide is intentionally left blank.

Return to Course Page