Web Technologies — Week 1

Mikheil Rukhaia

International Black Sea University,
Tbilisi, Georgia

October 8, 2015

Outline

1. Introduction
2. HTML5 Basics
3. Forms
4. Laboratory Work

Introduction

We will learn

- HTML5: latest version of Hyper Text Markup Language.
- CSS3: latest version of Cascading Style Sheets .
- JavaScript related techniques such as jQuery and drawing on Canvas.
- Some advanced techniques to use PHP and MySQL.

Evaluation schema

- Midterm Exam, Attendance/Lab work, and Course Project/Presentation — 20% each.
- Final Exam — 40%.
- Final grade: $(ME + AL + CP) \times 0.2 + FE \times 0.4$

Course project

- Groups of max. 2 persons (you can do it alone as well).
- Think yourself what you want to do and write 1 or 2 pages proposal (sooner you do so, more time you have to implement it).
- Each member of the group must participate in the final presentation.

Literature

Rob Larsen.
Beginning HTML and CSS.
Wiley Publishing Inc., 2013.

Matt Doyle.
Beginning PHP 5.3.
Wiley Publishing Inc., 2010.

Steve Fulton and Jeff Fulton.
HTML5 Canvas.
2nd Edition, O'Reilly Media, 2013.

HTML5 Basics

What is HTML5?

- HTML5 mainly follows rules of Strict XHTML.
- To use HTML5 it is mandatory to use `<!DOCTYPE html>` declaration at the beginning.
- Without the declaration browser switches to quirks mode, allowing some funky rules from the old versions of HTML.

Basic skeleton

- The basic skeleton of HTML document becomes:

```
<!DOCTYPE html>
<html>
 <head>
  <title> web page title </title>
 </head>
 <body>
  The body of the web page
 </body>
</html>
```

Backward compatibility

- Internet Explorer 8 and older versions cannot recognize HTML5 tags and CSS rules are not applied to them.
- Thus web page looks simply wrong in these browsers.
- To solve this problem, there is several libraries like `html5shiv` and `modernizr`.
- You can include them in the document using the following links:

```
<script
  src="http://cdnjs.cloudflare.com/ajax/libs/
               html5shiv/3.6/html5shiv.min.js">
// or
//             modernizr/2.6.1/modernizr.min.js">
</script>
```

Common elements

- Headings `<h1>-<h6>`, paragraphs `<p>`, preformatted sections `<pre>`, line breaks `<br />`, and addresses `<address>`
- Grouping elements: `<div>`, `<header>`, `<hgroup>`, `<nav>`, `<section>`, `<article>`, `<span>` and `<hr>`
- Presentational elements: `<b>`, `<i>`, `<sup>`, and `<sub>`
- Phrase elements: `<em>`, `<strong>`, `<abbr>`, `<dfn>`, `<blockquote>`, `<q>`, `<cite>`, `<code>`, `<kbd>`, `<var>`, and `<samp>`
- Lists: unordered `<ul>` and `<li>`; ordered `<ol>` and `<li>`; and definition lists `<dl>`, `<dt>`, and `<dd>`
- Editing elements: `<ins>` and `<del>`

New elements

- `<header>`: used to mark up headers for pages, articles and the like.
- `<hgroup>`: used to group together multiple levels of headings having some logical connection.
- `<footer>`: used to mark up footers; commonly used for legal copy.
- `<nav>`: used to represent a navigation section of the page, containing links to other pages or sections within the page.
- `<section>`: used to represent a section of a document.

New elements (ctd.)

- `<article>`: used to mark up "independent content"; it can be shared without the rest of the site context (e.g. a blog post, a movie review, a news article, etc.).
- `<aside>`: used to mark up related content to the surrounding one (e.g. sidebars, ads. and the like).
- `<figure>` and `<figcaption>`: used to mark up figures or illustrations and their labels (mainly used around `<canvas>`).
- `<mark>`: used to mark text like a highlighter in a paper book.

Audio and video

- The simplest way to include audio and video in your page is to use youtube or similar web services.
- If this is not an acceptable way, then you should get used to `<audio>` and `<video>` elements.
- The two tags are similar, so whatever we say about `<video>`, applies to `<audio>` as well.
- `<video>` works like the `<img>` element with `src`, `height`, and `width` attributes.
- Additionally it has the following attributes: `preload`, `autoplay`, `loop`, `muted`, `poster`, `controls`

Audio and video (ctd.)

- `preload` indicates that the browser should begin to download the video referenced in the `src` attribute even before the user presses Play (ignored on mobile devices).
- `poster` defines an image used as a placeholder until the video plays; if no poster specified, the first frame of the video is used.
- `controls` indicates whether the browser should include playback controls.

Media battle

- Audio and video elements are not well supported even in modern browsers.
- The reason is a battle between commercial and open-source worlds.
- Apple and Microsoft stand for patent-encumbered MPEG4 and MP3 formats, while Google, Opera, and Mozilla prefer free, open, and royalty-free formats such as OGG.
- Solution was to use flash, but apple refused to add its suppot on iOS.

Forms

HTML5 forms

- Forms in HTML5 are more advanced having dozens of new features.
- The problem is that not all features work in all browsers.
- Good news is that if a form control is not recognized by a browser, then it is displayed as a text input, so user can still enter some information.
- You should always test an HTML5 form in several recent browsers.

Backward compatibility

- A list of supported elements ad attributes can be found here: `http://wufoo.com/html5/`
- Polyfills are small libraries, usually written in JavaScript to provide support for new features in older browsers.
- There are many available polyfills for a variety of HTML5 features.
- The Modernizr project keeps a list at its project site: `https://github.com/Modernizr/Modernizr/wiki/HTML5-Cross-Browser-Polyfills`

New attributes of `<form>`

- There are two new attributes of the `<form>` element in HTML5.
- `novalidate` specifies that browser should not validate the form prior submission.
- `autocomplete` indicates whether or not the browser should auto-fill form values (default is `on`).

New types for `<input>`

- `color` for choosing a color by using a color wheel.
- `date` for entering a calendar date.
- `datetime` for entering a date and time with the time zone.
- `datetime-local` for entering a local date and time.
- `email` for entering either a single email address or a comma-separated list of email addresses.
- `month` for entering a year and month.
- `week` for entering a year and week number (2013-W01).

New types for <input> (ctd.)

- `number` for numerical input.
- `range` a slider that enables the user to choose a value from a range of numerical values.
- `search` for entering search terms.
- `tel` for entering telephone numbers.
- `time` for entering a time consisting of hours, minutes, seconds, and fractional seconds.
- `url` for entering website URLs.

New attributes

- `autocomplete` exists for each element in the form.
- `autofocus` indicates that the element should have focus when the page loads.
- `min` and `max` specify the minimum and maximum value.
- `pattern` specifies a regular expression that the element value is checked against.
- `placeholder` displays a short hint as the initial value in the input field.
- `required` indicates whether the input is a required element.
- `step` specifies the legal number intervals.

New attributes (ctd.)

- `list` refers to a `<datalist>` element that contains pre-defined options for the element.

- Example:

```
<input type="text" list="browsers" />

<datalist id="browsers">
  <option value="Internet Explorer" />
  <option value="Firefox" />
  <option value="Chrome" />
  <option value="Opera" />
  <option value="Safari" />
</datalist>
```

New attributes (ctd.)

- `wrap` attribute of `<textarea>` has additional values.
- `soft` (default) indicates that text is send to the server as it is typed.
- `hard` indicates that wherever the text wraps, it is transmitted to the server as a new line.

New elements

- `<label>` is used to place a label `for` the element.
- `<progress>` represents the progress through a task.
- `<meter>` is used to display a gauge (e.g. disk usage and the like).
- `<keygen>` provides a secure way to authenticate users.
- `<output>` represents the result of a calculation.

The <progress> element

- The closing tag is required.
- Should have value and max attribute in determinate mode.
- Rendering of indeterminate mode is different in that it cycles through the possible values.
- Example:

```
<progress value="14" max="20"></progress>
<progress name="indeterminate"></progress>
```

The <meter> element

- <meter> is similar to <process> but has more flexibility.
- It is possible to specify min, max values and low, high bounds of the range.
- If the value is beyond the bound, then display changes color.
- Example:

```
<meter min="0" value="95" high="90" max="100">
                                        </meter>
<meter min="0" value="50" high="90" max="100">
                                        </meter>
```

The <keygen> element

- <keygen> generates private and public keys (2048 or 1024 bit).
- Private key is stored locally, and public key is sent to the server.
- Public key could be used to generate a client certificate to authenticate the user in the future.
- Example:

```
<keygen name="certificate" />
```

The <output> element

- <output> is a place to display result of computation from a script.
- Its value is not sent to the server, when form is submitted.
- Example:

```
0
<input type="range" name="a" value="50"
  onchange="x.value=parseInt(a.value)+
                    parseInt(b.value)" />
100 +
<input type="number" name="b" value="50" />
=
<output name="x" for="a b"></output>
```

Laboratory Work

Exercises

- Create your personal web site, containing at least a homepage, study and blog pages.
- Study page must contain a table, listing the courses you have taken (categorized according to semesters), course lecturers and the grades you got.
- Your blog must contain at least two articles.
- Use `<nav>` to create a navigation bar.
- Use as much new HTML5 elements as possible.

Exercises (ctd.)

- Create a registration form containing detailed information about the user like name, surname, age, email, phone, etc.
- Use as much different form elements as possible.
- Use only HTML5 form techniques to validate all inputs.

Discussion?!