

Introduction To Web Programming I (CSC 211)

Lecture Note 2

How to Structure a Web Page 2

23rd September, 2021

How to code other HTML5 Semantic Elements

- ▶ Although there are many other HTML5 semantic elements, these are the most useful ones that are currently supported by modern browsers.
- ▶ **Accessibility and SEO guideline**
- ▶ Use the HTML5 semantic elements to indicate the structure of you web pages.

Other HTML5 semantic elements

Element	Contents
time	A date or date and time that can be parsed by a browser.
figure	An illustration, diagram, photo, code listing or the like that is referred to from the main content of the document.
figcaption	The caption that identifies a figure.

The attributes of the time element

Attribute	Description
datetime	A date and time in a standard format that can be parsed by a browser.
pubdate	A Boolean attribute that indicates that the date is the publication date for the article that contains the time element.

► **A time element**

<p>Next year's admission will probably be on <time datetime="2022-3-23">Monday 23rd</time>.</p>

► **The figure and figcaption elements**

<figure>

<code>

var today = new Date();

document.writeln(today.getFullYear());

</code>

<figcaption>

JavaScript code for getting the year from the current date

</figcaption>

When and how to code div and span elements

- ▶ **div** is a block element for structuring a web page – it lets you divide a page into divisions that can be formatted and positioned with CSS.
- ▶ Before HTML5, *div* elements were used to define divisions within the body of a document. Now, the HTML5 semantic elements will be replacing *div* elements.
- ▶ **span** is an inline element for formatting text – it lets identify text that can be formatted with CSS.
- ▶ Before HTML5, *span* elements were used to identify portions of text that you could apply formatting to. Today, a better practice is to use the block elements for special types of text (pre, blockquote, address, e.t.c.) and inline elements for for formatting text (i, b, sub, sup, br, e.t.c) and inline elements for identifying content such as (abbr, cite, code, dfn, em, kbd, q, samp, strong, var, e.t.c) to identify content and use the CSS to format that content.

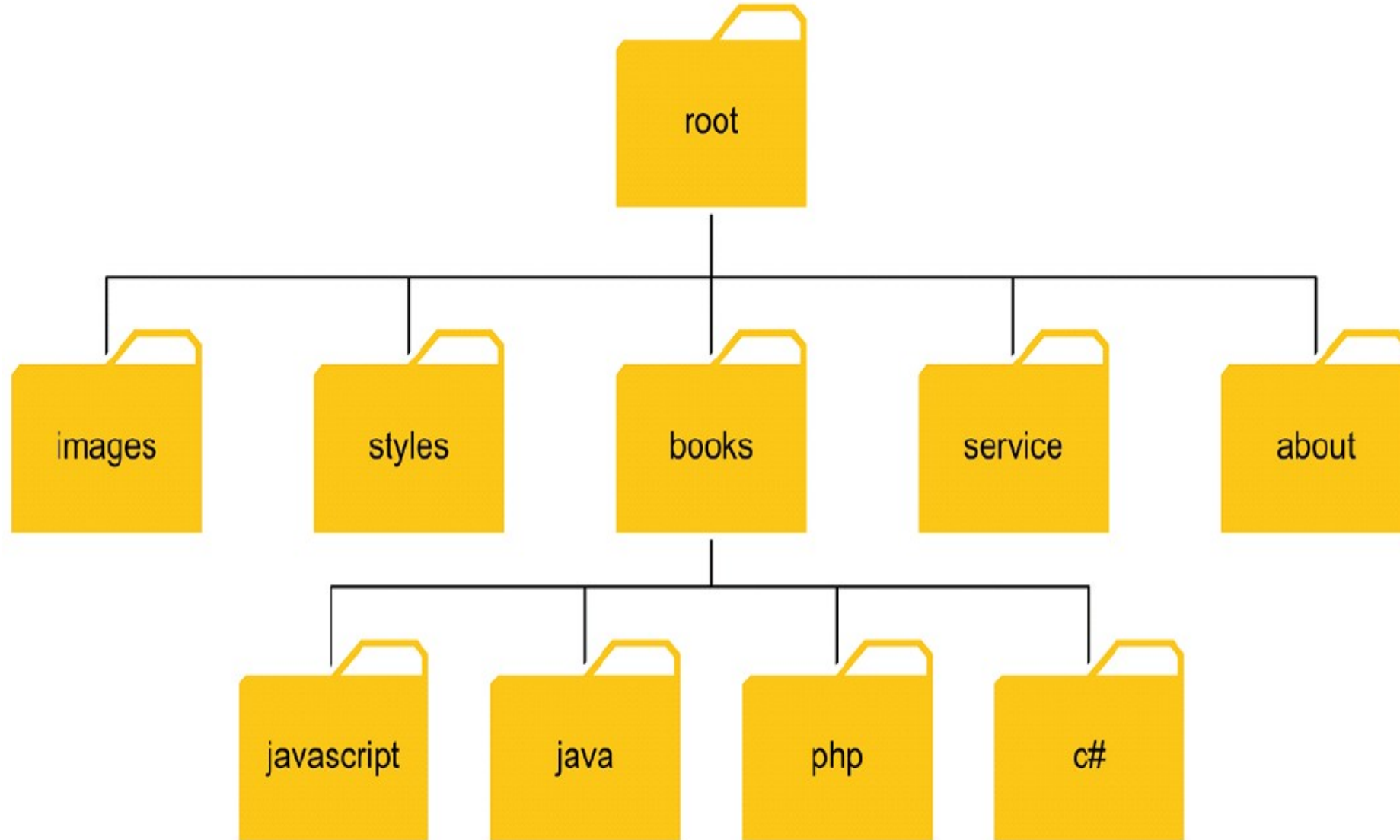
A page that's structured with div and span elements

```
<body>
  <div id="header">
    <h1>San Joaquin Valley Town Hall</h1>
  </div>
  <div id="main">
    <p><span id="welcome">Welcome to San Joaquin Valley Town Hall.</span>
      We have some fascinating speakers for you this season!</p>
  </div>
  <div id="footer">
    <p>&copy; Copyright 2021 CSC 211 – Intro to Web Programming I</p>
  </div>
</body>
```

How to links, lists, and images

- ▶ **How to code absolute and relative URLs**
- ▶ When you code an ***absolute URL***, you code the complete URL including the domain name for the site. Absolute URLs let you display pages at other websites.
- ▶ When you code ***relative URL***, you base it on the current folder, which is the folder that contains the current page.
- ▶ A *root-relative path* is relative to the root folder of the website. It always starts with a slash. Then, to go down one subfolder, you can code the subfolder name and a slash. To go down two subfolders, you can code a second subfolder name and a slash. And so on.
- ▶ A *document-relative path* is relative to the folder the current document is in. then, to go down one subfolder, you can code the subfolder name followed by a slash. To go down two subfolders, you can code a second subfolder name followed by another slash. And so on.
- ▶ You can also go up in a document-relative path. To go up one level from the current folder, you can code two periods and a slash. To go up two levels, you code two periods and a slash followed by two more periods and a slash. And so on

A simple website directory structure



Examples of absolute and relative URLs

- ▶ **Absolute URLs**

`https://www.fud.edu.ng/index.html`

`https://www.jsiit.edu.ng/hostels/booking.html`

- ▶ **Root-relative paths**

`/login.html`

(refers to root/login.html)

`/images/logo.gif`

(refers to root/images/logo.gif)

- ▶ **Document-relative paths that navigate down from the root folder**

`images/logo.gif`

(refers to root/images/logo.gif)

`books/php/overview.html`

(refers to root/books/php/overview.html)

- ▶ **Document-relative paths that navigate up from the root/books folder**

`../index.html`

(refers to root/index.html)

`../images/logo.gif`

(refers to root/images/logo.gif)

How to code links

- ▶ Most web pages contain *links* that go to other web pages or web resources.
- ▶ To code a link, you use the anchor tag element (<a>)
- ▶ The <a> is an inline element that create a link that loads another web page. The href attribute of this element identifies the page to be loaded.
- ▶ The text content of a link is underlined by default to indicate that it is clickable.
- ▶ If a link has not been visited, it is displayed in blue. If it has been visited, it is displayed as purple. You can change these defaults using CSS.
- ▶ If the mouse hovers over a link, the cursor is changed to a hand with the finger pointed.
- ▶ Examples

- ▶ **A link to a web page in the same folder**

```
<p>Go view our <a href="courses.html">Courses list</a>.</p>
```

- ▶ **A link to a web page in a subfolder of the parent folder**

```
<p>Read about the <a href="../company/services.html">services we provided</a>.</p>
```

- ▶ **A link to a web page based on the root folder**

```
<p>View your <a href="/orders/cart.html">Shooping cart</a>.</p>
```

How to code lists

- ▶ There are two basic types of lists: **ordered lists** and **unordered lists**.
- ▶ By default, *unordered lists* are displayed as bulleted list and an *ordered list* is displayed as a numbered list.
- ▶ You can code *li* (list item) element for each of the two types of the lists to create each item in the lists.
- ▶ When you work with the *li* element, you should be aware that it can contain a text, inline elements, or block elements. For example, an *li* element can contain an `<a>` element that defines a link. In fact, it is a best practice to code a series of links within an unordered list.
- ▶ **Elements that create ordered and unordered lists**

Element	Description
<code></code>	Creates an unordered list.
<code></code>	Creates an ordered list.
<code></code>	Creates a list item for an unordered or ordered list.

HTML that creates two lists

```
<p>We have books on a variety of languages, including</p>
```

```
<ul>
```

```
  <li>JavaScript</li>
```

```
  <li>PHP and MySQL</li>
```

```
  <li>Servlets and JSP</li>
```

```
  <li>ASP.NET</li>
```

```
</ul>
```

```
<p>You will need to complete the following steps:</p>
```

```
<ol>
```

```
  <li>Enter your billing information.</li>
```

```
  <li>Enter your shipping information.</li>
```

```
  <li>Confirm your order.</li>
```

```
</ol>
```

How to include images

- ▶ Images are an important part of most web pages. To display image, you can use the `img` element.
- ▶ The `img` element is an inline element that is used to display an image that is identified by the `src` attribute.
- ▶ The height and width attributes can be used to indicate the size of an image so the browser can allocate the correct amount of space on the page. These attributes can also be used to size an image, but it is usually better to use an image editor to do the sizing.
- ▶ **JPEG (Joint Photographic Experts Group)** files commonly use the JPG extension and are typically used for photographs and scan. **GIF (Graphic Interchange Format)** files are typically used for small illustrations and logos. And **PNG (Portable Network Graphics)** files combine aspects of JPEG and GIF files.
- ▶ Above image formats are supported by most browsers.
- ▶ For images with useful contents, always code an alt attribute that describes the image. For images that are used for decoration, code the alt attribute with no value (`""`).

How to code character entities

► Attributes of the element

Attribute	Description
src	Specifies the relative or absolute URL of the image to display. It is a required attribute.
alt	Specifies alternate text to display in place of the image. This text is read aloud by screen readers for users with disabilities. It is required.
height	Specifies the height of the image in pixels.
width	Specifies the width of the image in pixels.

► Example

```

```

```
<h1>Faculty of Computing, Federal University Dutse</h1>
```

SAMPLE SIMPLE WEBSITE