HTML5

Presentation by Uplatz

Contact Us: https://training.uplatz.com/

Email: info@uplatz.com

Phone:+44 7836 212635

Table Of Contents:

- Navigation Bars
- Label Elements
- Output Elements
- Video and Media Elements
- Progress Elements
- Selection Menu Elements
- Embed

Header Element:

- The <header> element represents introductory content for its nearest ancestor sectioning content or sectioning root element.
- A <header> typically contains a group of introductory or navigational aids.
- **Note**: The header element is not sectioning content; it doesn’t introduce a new section.

Examples:

```
<header>
 <p>Welcome to...</p>
 <h1>Voidwars!</h1>
</header>
```

In this example, the <article> has a <header>.

```
<article>
 <header>
 <h1>Flexbox: The definitive guide</h1>
 </header>
 <p>The guide about Flexbox was supposed to be here,
but it turned out Wes wasn’t a Flexbox expert
either.</p>
</article>
```

Footer Element:

- The <footer> element contains the footer part of the page.
- Here is an example for <footer> element that contain p paragraph tag.

```
 <footer>
  <p>All rights reserved</p>
 </footer>
```

Section Element:

- The <section> element represents a generic section to thematically group content.
- Every section, typically, should be able to be identified with a heading element as a child of the section.
- You can use the <section> element within an <article> and vice-versa.
- Every section should have a theme (a heading element identifying this region)
- Don't use the <section> element as a general styling 'container'.
- If you need a container to apply styling, use a <div> instead.

- In the following example, we're displaying a single blog post with multiple chapters each chapter is a section (a set of thematically grouped content, which can be identified by the heading elements in each section).

```
<article>
 <header>
 <h2>Blog Post</h2>
 </header>
 <p>An introduction for the post.</p>
 <section>
 <h3>Chapter 1</h3>
 <p>...</p>
 </section>
```

```
<section>
 <h3>Chapter 2</h3>
 <p>...</p>
 </section>
 <section>
 <h3>Comments</h3> ...
 </section>
</article>
```

Navigation Bars:

Basic Navigation Bar:

- Navigation bars are essentially a list of links, so the ul and li elements are used to encase navigation links.

```
<ul>
 <li><a href="#">Home</a></li>
 <li><a href="#">About</a></li>
 <li><a href="#">Contact</a></li>
</ul>
```

HTML5 Navigation Bar:

- To make a navigation bar using the HTML5 nav element, encase the links within the nav tag.

```
<nav>
 <a href="#">Home</a>
 <a href="#">About</a>
 <a href="#">Contact</a>
</nav>
```

Label Element:

Attributes **Description**

for Reference to the target ID Element. I.e: for="surname"

Form HTML5, [Obsolete] Reference to the form containing the Target Element. Label elements are expected within a <form> Element. If the form="someFormId" is provided this allows you to place the Label anywhere in the document.

- About Label The element is used to reference a form action element.
- In the scope of User Interface it's used to ease the target / selection of elements like Type radio or checkbox.

- as wrapper It can enclose the desired action element

```
<label>
 <input type="checkbox" name="Cats">
 I like Cats!
</label>
```

- (Clicking on the text the target input will toggle it's state / value)
- <label> as reference
- Using the for attribute you don't have to place the control element as descendant of label - but the for value must match it's ID

```
<input id="cats" type="checkbox" name="Cats">
<label for="cats" >I like Cats!</label>
```

Note:

- **Don't use more than one Control Element within a <label> element**

Basic Use:

Simple form with labels...

```
<form action="/login" method="POST">
<label for="username">Username:</label>
 <input id="username" type="text" name="username" />
 <label for="pass">Password:</label>
 <input id="pass" type="password" name="pass" />
 <input type="submit" name="submit" />
</form>
<form id="my-form" action="/login" method="POST">
```

```
<input id="username" type="text" name="username" />
<label for="pass">Password:</label>
 <input id="pass" type="password" name="pass" />
 <input type="submit" name="submit" />
</form>
<label for="username" form="my-form">Username:</label>
```

Output Element:

Attribute Description

Global Attributes that are available to any HTML5 element.

Name A string representing the name of an output. As a form element, output can be referenced by it's name using the document.forms property

. This attribute is also used for collecting values on a form submit.

for A space separated list of form element ids (e.g. <inputs id="inp1"> for value is "inp1") that the output is meant to display calculations for.

Form A string representing the <form> that is associated to the output. If the output is actually outside the <form>, this attribute will ensure that the output still belongs to the <form> and subject to collections and submits of said <form>.

Output Element Using For and Form Attributes :

- The following demo features an element's use of the [for] and [form] attributes.
- Inline JavaScript is commonly used in forms as this example demonstrates. Although the elements are type="number", their values are not numbers, the text

- So if you require the values to be calculated, you must convert each value into a number using methods such as: parseInt(), parseFloat(), Number(), etc.

```
<form id="form1" name="form1" oninput="out1.value =
parseInt(in1.value, 10) + parseInt(in2.value,
10)">
 <fieldset>
 <legend>Output Example</legend>
 <input type="number" id="in1" name="in1" value="0">
 <br/>
 +
<input type="number" id="in2" name="in2" value="0">
 </fieldset>
</form>
```

```
<!--[for] attribute enables out1 to display calculations
for in1 and in2.-->
<!--[form] attribute designates form1 as the form
owner of out1 even if it isn't a descendant.-->
<output name="out1" for="in1 in2"
form="form1">0</output>
```

Output Element with Attributes:

```
<output name="out1" form="form1" for="inp1
inp2"></output>
```

Void Elements:

- **Not all HTML tags are of the same structure.**
- **While most elements require an opening tag, a closing tag, and contents, some elements - known as void elements - only require an opening tag as they themselves do not contain any elements.**

Void elements:

- HTML 4.01/XHTML 1.0 Strict includes the following void elements:

area - clickable, defined area in an image

base - specifies a base URL from which all links base

br - line break

col - column in a table [deprecated]

hr - horizontal rule (line)

img - image

input - field where users enter data

link - links an external resource to the document

meta - provides information about the document

param - defines parameters for plugins

- HTML 5 standards include all non-deprecated tags from the previous list and

command - represents a command users can invoke [obsolete]

keygen - facilitates public key generation for web certificates [deprecated]

source - specifies media sources for picture, audio, and video elements

The example below does not include void elements:

```
<div>
 <a href="http://stackoverflow.com/">
 <h3>Click here to visit <i>Stack Overflow!</i></h3>
 </a>
 <button onclick="alert('Hello!');">Say Hello!</button>
```

```
<p>My favorite language is <b>HTML</b>. Here are my others:</p>
 <ol>
 <li>CSS</li>
 <li>JavaScript</li>
 <li>PHP</li>
 </ol>
</div>
<img src="https://cdn.sstatic.net/Sites/stackoverflow/company/img/logos/so/so-icon.png" />
<br>
<hr>
<input type="number" placeholder="Enter your favorite number">
```

- With the exception of the img tag, all of these void elements have only an opening tag.
- The img tag, unlike any other tag, has a self closing / before the greater than sign of the opening tag.
- It is best practice to have a space before the slash.

Media Elements

Attribute	**Details**
width	Sets the element's width in pixels.
height	Sets the element's height in pixels.
<source>	Defines resources of the audio or video files
track	Defines the text track for media elements
controls	Displays controls

autoplay Automatically start playing the media

loop Plays the media in a repeated cycle

muted Plays the media without sound

poster Assigns an image to display until a video is loaded

Audio:

- HTML5 provides a new standard for embedding an audio file on a web page.
- You can embed an audio file to a page using the <audio> element:

```
<audio controls>
 <source src="file.mp3" type="audio/mpeg">
Your browser does not support the audio element.
</audio>
```

Video

- You can embed also a video to a webpage using the <video> element:

```
<video width="500" height="700" controls>
 <source src="video.mp4" type="video/mp4">
Your browser does not support the video tag.
</video>
```

Using `<video>` and `<audio>` element to display audio/video content:

- Use the HTML or <audio> element to embed video/audio content in a document.
- The video/audio element contains one or more video/audio sources.
- To specify a source, use either the src attribute or the <source> element; the browser will choose the most suitable one.

Audio tag example:

```
<!-- Simple video example -->
<video src="videofile.webm" autoplay
poster="posterimage.jpg">
 Sorry, your browser doesn't support embedded videos,
 but don't worry, you can <a
href="videofile.webm">download it</a>
 and watch it with your favorite video player!
</video>
<!-- Video with subtitles -->
<video src="foo.webm">
 <track kind="subtitles" src="foo.en.vtt" srclang="en"
label="English">
 <track kind="subtitles" src="foo.sv.vtt" srclang="sv"
label="Svenska">
</video>
```

```
<!-- Simple video example -->
<video width="480" controls poster="https://archive.org/download/WebmVp8Vorbis/webmvp8.gif" >
 <source src="https://archive.org/download/WebmVp8Vorbis/webmvp8.webm" type="video/webm">
 <source src="https://archive.org/download/WebmVp8Vorbis/webmvp8_512kb.mp4" type="video/mp4">
 <source src="https://archive.org/download/WebmVp8Vorbis/webmvp8.ogv" type="video/ogg">
 Your browser doesn't support HTML5 video tag.
</video>
```

Audio tag example:

```
<!-- Simple audio playback -->
<audio
src="http://developer.mozilla.org/@api/deki/files/2926/=
AudioTest_(1).ogg" autoplay>
 Your browser does not support the
<code>audio</code> element.
</audio>
<!-- Audio playback with captions -->
<audio src="foo.ogg">
 <track kind="captions" src="foo.en.vtt" srclang="en"
label="English">
 <track kind="captions" src="foo.sv.vtt" srclang="sv"
label="Svenska">
</audio>
```

Video header or background:

- Adding a video that will autoplay on a loop and has no controls or sound.
- Perfect for a video header or background.

```
<video width="1280" height="720" autoplay muted loop
poster="video.jpg" id="videobg">
 <source src="video.mp4" type="video/mp4">
 <source src="video.webm" type="video/webm">
 <source src="video.ogg" type="video/ogg">
</video>
```

- This CSS provides a fallback if the video cannot be loaded.
- Note that is it recomended to use the first frame of the video as the poster video.jpg.

```
#videobg {
 background: url(video.jpg) no-repeat;
 background-size: cover;
}
```

Progress Element:

Parameter Value

max How much work the task requires in total

value How much of the work has been accomplished already

position This attribute returns the current position of the <progress> element

labels This attribute returns a list of <progress> element labels (if any)

Progress:

- The <progress> element is new in HTML5 and is used to represent the progress of a task

```
<progress value="22" max="100"></progress>
```

This creates a bar filled 22%

Changing the color of a progress bar:

- Progress bars can be styled with the progress[value] selector.
- This example gives a progress bar a width of 250px and a height of 20px

```
progress[value] {
 width: 250px;
 height: 20px;
}
```

- Progress bars can be especially difficult to style.
- Chrome / Safari / Opera
- These browsers use the -webkit-appearance selector to style the progress tag. To override this, we can reset the appearance.

```
progress[value] {
 -webkit-appearance: none;
 appearance: none;
}
```

Now, we can style the container itself

```
progress[value]::-webkit-progress-bar {
 background-color: "green";
}
```

Firefox:

- Firefox styles the progress bar a little differently. We have to use these styles

```
progress[value] {
 -moz-appearance: none;
 appearance: none;
 border: none; /* Firefox also renders a border */
}
```

Internet Explorer:

- Internet Explorer 10+ supports the progress element. However, it does not support the background-color property.
- You'll need to use the color property instead.

```
progress[value] {
 -webkit-appearance: none;
 -moz-appearance: none;
 appearance: none;
 border: none; /* Remove border from Firefox */
 width: 250px;
 height: 20px;
 color: blue;
}
```

HTML Fallback:

- For browsers that do not support the progress element, you can use this as a workaround.

```
<progress max="100" value="20">
 <div class="progress-bar">
```

```
<span style="width: 20%;">Progress: 20%</span>
 </div>
</progress>
```

- Browsers that support the progress tag will ignore the div nested inside.
- Legacy browsers which cannot identify the

progress tag will render the div instead.

Selection Menu Controls

Select Menu:

The <select> element generates a drop-down menu from which the user can choose an option.

```
<select name="">
 <option value="1">One</option>
 <option value="2">Two</option>
 <option value="3">Three</option>
 <option value="4">Four</option> </select>
```

Changing the Size:

- You can change the size of the selection menu with the size attribute.
- A size of 0 or 1 displays the standard dropdown style menu.
- A size greater than 1 will convert the drop-down into a box displaying that many lines, with one option per line and a scrollbar in order to scroll through the available options.

<select name="" size="4"></select>

- **Multi-option Selection Menus**
- **By default, users can only select a single option. Adding the multiple attribute allows users to select multiple**
- **options at once and submit all selected options with the form.**

- Using the multiple attribute automatically converts the drop-down menu into a box as if it had a size defined.
- The default size when this occurs is determined by the specific browser you are using, and it is not possible to change it back to a drop-down style menu while allowing multiple selections.
- **<select name="" multiple></select>**
- When using the multiple attribute, there is a difference between using 0 and 1 for the size, whereas no difference exists when not using the attribute.
- Using 0 will cause the browser to behave in whatever default manner it is programmed to do.
- Using 1 will explicitly set the size of the resulting box to only one row high.

Options:

The options inside a selection menu are what the user will be selection.

The normal syntax for an option is as follows:

<option>Some Option</option>

- However, it's important to note that the text inside the <option> element itself is not always used, and essentially becomes the default value for attributes which are not specified.
- The attributes which control the actual appearance and function of the option are value and label.
- The label represents the text which will be displayed in the drop-down menu (what you're looking at and will click on to select **it**

-). The value represents the text which will be sent along with form submission.
- If either of these values is omitted,it uses the text inside the element as the value instead.
- So the example we gave above could be "expanded" to this:

<option label="Some Option" value="Some Option">

- Note the omission of the inside text and end tag, which are not required to actually construct an option inside the menu.
- If they were included, the inside text would be ignored because both attributes are already specified and the
- text is not needed.
- However, you probably won't see a lot of people writing them this way. The most common way

- it's written is with a value that will be sent to the server, along with the inside text which eventually becomes the label attribute, like so:

<option value="option1">Some Option</option>

- Selecting an option by default You can also specify a certain option to be selected in the menu by default by attaching the selected attribute to it.
- By default, if no option is specified as selected in the menu, the first option in the menu will be selected when rendered.
- If more than one option has the selected attribute attached, then the last option present in the menu

with the attribute will be the one selected by default.

<option value="option1" selected>Some option</option>

- If you're using the attribute in a multi-option selection menu, then all the options with the attribute will be selected by default, and none will be selected if no options have the attribute.

```
<select multiple>
 <option value="option1" selected>Some option</option>
 <option value="option2" selected>Some option</option>
</select>
```

Option Groups:

- You can neatly group your options within a selection menu in order to provide a more structured layout in a long list of options by using the <optgroup> element.

- The syntax is very basic, by simply using the element with a label attribute to identify the title for the group, and containing zero or more options that should be within that group.

```
<select name="">
 <option value="milk">Milk</option>
 <optgroup label="Fruits">
 <option value="banana">Bananas</option>
 <option value="strawberry">Strawberries</option>
 </optgroup>
 <optgroup label="Vegetables" disabled>
 <option value="carrot">Carrots</option>
 <option value="zucchini">Zucchini</option>
 </optgroup>
</select>
```

- When using option groups, not all options need to be contained within a group.
- As well, disabling an option group will disable all options within the group, and it is not possible to manually re-enable a single option within a

disabled group

Datalist

- The <datalist> tag specifies a list of pre-defined options for an <input> element. It provide an "autocomplete“ feature on <input> elements.
- Users will see a drop-down list of options as they write.

<input list="Languages">

<datalist id="Languages">

```
<option value="PHP">
 <option value="Perl">
 <option value="Python">
 <option value="Ruby">
 <option value="C+">
</datalist>
```

- Browser Support
- Chrome Edge Mozilla Safari Opera
- 20.0 10.0 4.0 Not Supported 9.0

Embed

Parameters	**Details**
src	Address of the resource
type	Type of embedded resource
width	Horizontal dimension
height	Vertical dimension

Basic usage

- The embed tag is new in HTML5.
- This element provides an integration point for an external (typically non-HTML)

application or interactive content.

<embed src="myflash.swf">

Defining the MIME type:

The MIME type must be defined using the type attribute.

<embed type="video/mp4" src="video.mp4" width="640" height="480">

Summary:

- Navigation Bars
- Label Elements
- Output Elements
- Video and Media Elements
- Progress Elements
- Selection Menu Elements
- Embed

Thank You……….

If you have any quries please write to info@uplatz.com".