Internet and Web Principal

Week 5

HTML 2

Content

1. Adding Images
2. Table Markup
3. Forms

Adding Images

In order to be inserted inline in the content, images composed of a grid of colored pixels must be stored in the PNG, JPEG, or GIF file formats.

The SVG format is used for vector images, such as icons and graphics created with drawing software such as Adobe Illustrator.

Adding Images

If you have a source image in another popular format, such as TIFF, BMP, or EPS, you must convert it to a web format before adding it to the website. If you must maintain your graphic file in its original format for some reason (for example, a file for a CAD application), you can make it available as an external picture by making a direct connection to the image file.

Get the design

Adding Images

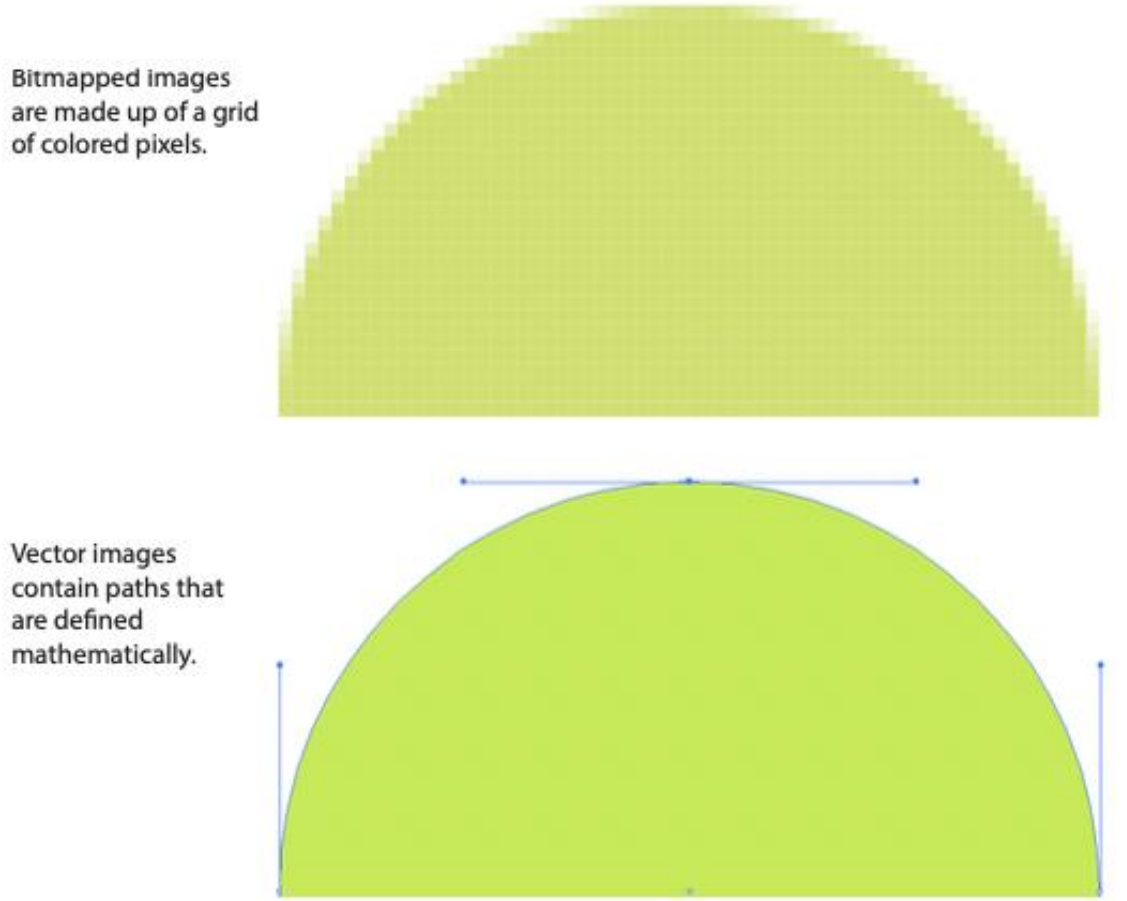


FIGURE 7-1. A comparison of circles saved in bitmapped and vector formats.

Learning Web Design, Jennifer Niederst Robbins, O'Reilly Media, Inc., 2018, Page 133

Adding Images

The img element instructs the browser to "Place an image here." You can also insert an image element directly into the flow of the text at the point where you want the image to appear. Images remain in the flow of the text, aligned with the baseline of the text, and do not generate any line breaks.

Adding Images

```
<p>This summer, try making pizza <img src="pizza.png" alt="">
 on your grill.</p>
```

This summer, try making pizza  on your grill.

FIGURE 7-2. By default, images are aligned with the baseline of the surrounding text and do not cause a line break.

Learning Web Design, Jennifer Niederst Robbins, O'Reilly Media, Inc., 2018, Page 134

Adding Images

When the browser encounters the img element, it sends a request to the server, where it downloads the image file before displaying it on the page. On a fast network with a fast computer or device, even though each image file is requested separately, the page appears to come instantly. We may be well aware of the wait for images to be fetched one at a time on mobile devices with slow network connections.

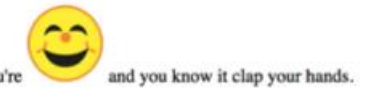
Adding Images

The src attribute's value is the image file's URL. Because the images you use on your pages will most likely be hosted on your own server, you will use relative URLs to refer to them. Images for a website are typically organized by developers into a directory called images or img (in fact, it helps search engines when you do it that way). Each component of the site may even have its own image directory.

Adding Images

```
<p>If you're <img src="happy.gif" alt="happy">
and you know it clap your hands.</p>
```

With image displayed



If you're and you know it clap your hands.

Firefox

If you're happy and you know it clap your hands.

Chrome (Mac & Windows)

If you're happy and you know it clap your hands.

MS Edge (Windows)

If you're happy and you know it clap your hands.

Safari (iOS)

If you're happy and you know it clap your hands.

Safari (Mac)

If you're and you know it clap your hands.

FIGURE 7-4. Most browsers display alternative text in place of the image if the image is not available. Safari for macOS is a notable exception. Firefox's substitution is the most seamless.

Every **img** element must also contain an **alt** attribute that provides a text alternative to the image for those who are not able to see it. Alternative text (also called alt text) should serve as a substitute for the image content—conveying the same information and function. Alternative text is used by screen readers, search engines, and graphical browsers when the image doesn't load (FIGURE 7-4). (Learning Web Design, Jennifer Niederst Robbins, O'Reilly Media, Inc., 2018, Page 136)

Learning Web Design, Jennifer Niederst Robbins, O'Reilly Media, Inc., 2018, Page 136

Adding Images

The width and height properties specify the image's dimensions in pixels. Instead of recreating the page each time a new image arrives, browsers employ the provided dimensions to hold the proper amount of space in the layout while the photos load, resulting in faster page display. If you simply specify one dimension, the image will scale appropriately.

Make sure that the pixel dimensions you specify correspond to the image's real dimensions. If the pixel values differ from the image's actual dimensions, the browser resizes it to match the supplied values. If you use width and height characteristics and your image appears distorted or even somewhat fuzzy, double-check that the values are in sync.

Adding Images

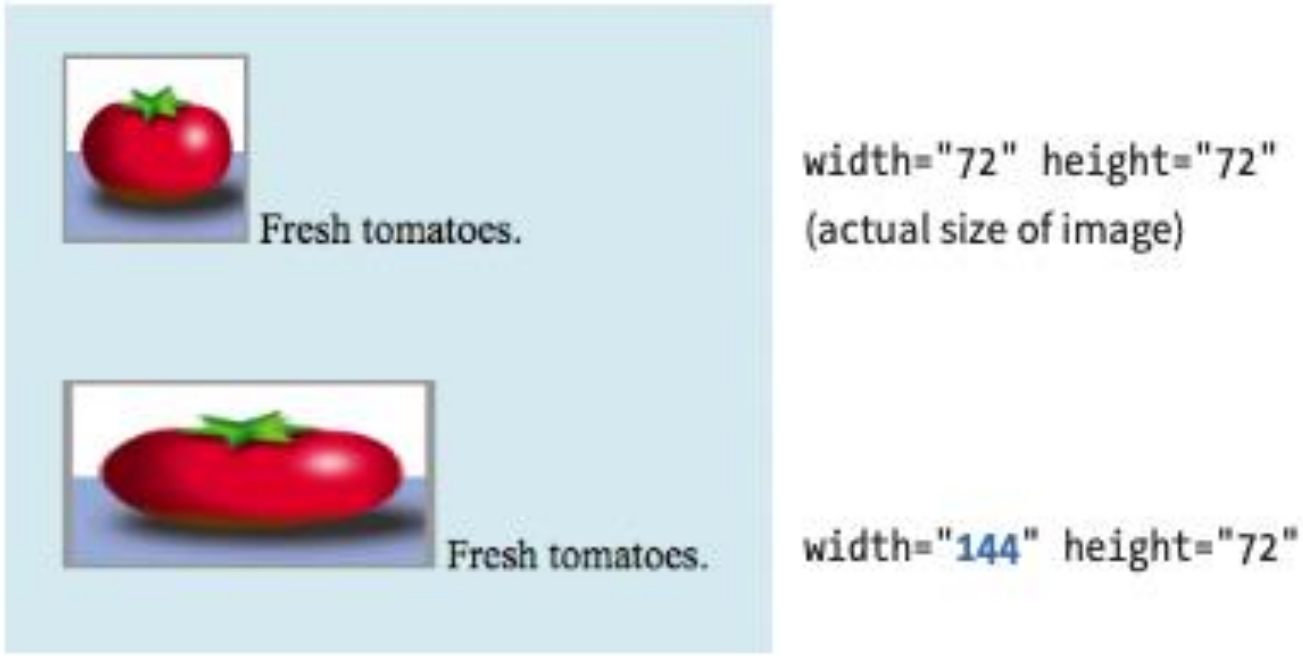


FIGURE 7-5. Browsers resize images to match the provided **width** and **height** values, but you should not resize images this way.

Learning Web Design, Jennifer Niederst Robbins, O'Reilly Media, Inc., 2018, Page 137

Table Markup

HTML tables were established for situations in which data is needed to be added to a web page in the form of rows and columns. Schedules, product comparisons, statistics, and other forms of information can all be organized using tables. It should be noted that "data" does not always imply numbers. A table cell can hold any type of data, including numbers, text components, photos, and multimedia objects.

Table Markup

In visual browsers, the presentation of data in rows and columns provides readers with an instant comprehension of the relationships between data cells and their associated header labels. Tables were the only way to create multi-column layouts or regulate alignment and spacing before style sheets. Layout tables, particularly elaborate nested table configurations that were formerly commonplace in web design, have gone extinct. If you need rows and columns for presentation, there are CSS-based solutions.

Table Markup

Minimal Table Structure

Menu item	Calories	Fat (g)
Chicken noodle soup	120	2
Caesar salad	400	26

FIGURE 8-2 reveals the structure of this table according to the HTML table model. All of the table's content goes into cells that are arranged into rows. Cells contain either header information (titles for the columns, such as "Calories") or data, which may be any sort of content.

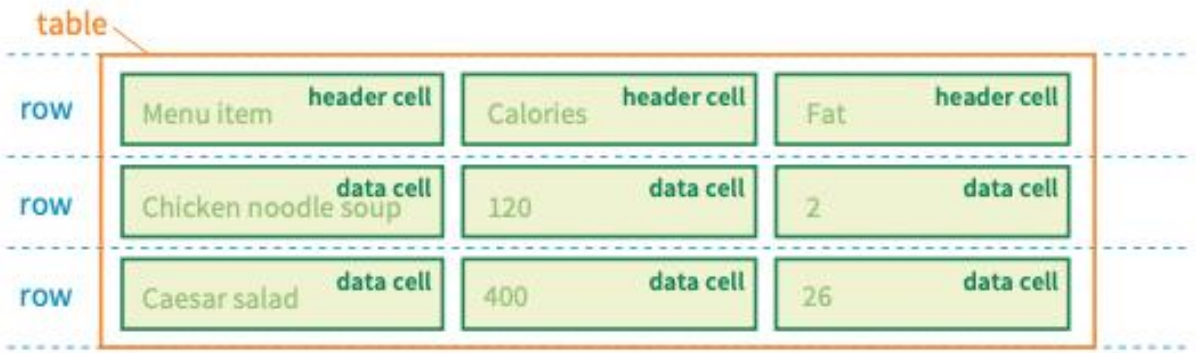


FIGURE 8-2. Tables are made up of rows that contain cells. Cells are the containers for content.

Learning Web Design, Jennifer Niederst Robbins, O'Reilly Media, Inc., 2018, Page 165

Table Markup

Minimal Table Structure

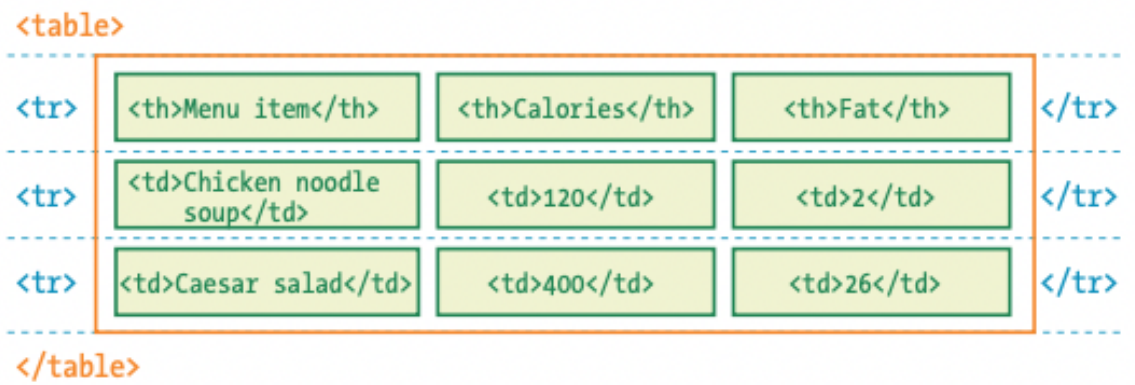


FIGURE 8-3. The elements that make up the basic structure of a table.

FIGURE 8-3 depicts the table elements (table), rows (tr, for "table row"), and cells (th, for "table headers," and td, for "table data"). Cells are the core of the table since they contain the real content. The other pieces simply hold everything together. Column elements are not visible. The number of cells in each row indicates the number of columns in a table. This is one of the factors that can making HTML tables difficult. Rows are simple—if you want three rows in your table, just add three tr components. Columns are distinct. For a four-column table, ensure that each row has four td or th components.

Learning Web Design, Jennifer Niederst Robbins, O'Reilly Media, Inc., 2018, Page 165

Table Markup

Table Headers

Table headers are important because they provide context or information about the cells in the row or column they precede. Alternative browsing devices may treat the th element differently than the tds element. As a result, headers are an important tool for making table content accessible. Make no attempt to fool them by formatting a row of td elements differently than the rest of the table. Don't, on the other hand, avoid using th elements because of their default rendering (bold and centered). Instead, semantically mark up the headers and use a style rule to adjust the display later.

Table Markup

Spanning Cells

Cell spanning, which is the stretching of a cell to cover numerous rows or columns, is a key characteristic of table structure. Spanning cells allows you to design complicated table structures, but it also makes the markup more difficult to keep track of. It may also make it more difficult for users who use screen readers to follow. By combining the colspan and rowspan attributes, you can create a header or data cell span.

Table Markup

Learning Web Design, Jennifer Niederst Robbins, O'Reilly Media, Inc., 2018, Page 168

Column spans, created with the **colspan** attribute in the **td** or **th** element, stretch a cell to the right to span over the subsequent columns (FIGURE 8-6). Here a column span is used to make a header apply to two columns (I've added a border around the cells to reveal the structure of the table in the screenshot). Notice in the first row (**tr**) that there is only one **th** element, while the sec- ond row has two **td** elements. The **th** for the column that was spanned over is no longer in the source; the cell with the **colspan** stands in for it. Every row should have the same number of cells or equivalent **colspan** values. For example, there are two **td** elements and the **colspan** value is 2, so the implied number of columns in each row is equal.

(Learning Web Design, Jennifer Niederst Robbins, O'Reilly Media, Inc., 2018, Page 168)

```
<table>
   <tr>
      <th colspan="2">Fat</th>
   </tr>
   <tr>
      <td>Saturated Fat (g)</td>
      <td>Unsaturated Fat (g)</td>
   </tr>
</table>
```

Fat	
Saturated Fat (g)	Unsaturated Fat (g)

FIGURE 8-6. The **colspan** attribute stretches a cell to the right to span the specified number of columns.

Table Markup

Row spans, created with the **rowspan** attribute, work just like column spans, but they cause the cell to span downward over several rows. In this example, the first cell in the table spans down three rows (FIGURE 8-8). Again, notice that the **td** elements for the cells that were spanned over (the first cells in the remaining rows) do not appear in the source. The **rowspan="3"** implies cells for the subsequent two rows, so no **td** elements are needed.

(Learning Web Design, Jennifer Niederst Robbins, O'Reilly Media, Inc., 2018, Page 169)

Learning Web Design, Jennifer Niederst Robbins, O'Reilly Media, Inc., 2018, Page 169

```
<table>
   <tr>
      <th rowspan="3">Serving Size</th>
      <td>Small (8oz.)</td>
   </tr>
   <tr>
      <td>Medium (16oz.)</td>
   </tr>
   <tr>
      <td>Large (24oz.)</td>
   </tr>
</table>
```

Serving Size	Small (8oz.)
	Medium (16oz.)
	Large (24oz.)

FIGURE 8-8. The **rowspan** attribute stretches a cell downward to span the specified number of rows.

Table Markup

Describing table Content

Giving your table a title or description with the caption element is the most effective technique to provide visually impaired users an overview of your table. Captions appear next to (or above) the table and can be used to describe the table's contents or to provide insights on how it is constructed.

```
<table>
        <caption>Table Content</caption>
        <tr>
                <th>Menu</th>
                <th>Display</th>
                <th>Cost</th>
        </tr>
</table>
```

Table Markup

Describing table Content

Giving your table a title or description with the caption element is the most effective technique to provide visually impaired users an overview of your table. Captions appear next to (or above) the table and can be used to describe the table's contents or to provide insights on how it is constructed.

```
<table>
        <caption>Table Content</caption>
        <tr>
                <th>Menu</th>
                <th>Display</th>
                <th>Cost</th>
        </tr>
</table>
```

Forms

A working form is made up of two elements. The first component is the form that appears on the page and is generated with HTML markup. Forms are made up of buttons, input fields, and drop-down menus (together referred to as form controls) that are used to collect information from the user. Text and other components may also be included in forms.

The other component of a web form is a server-side program or script that processes the data collected by the form and produces an appropriate response. It is what allows the form to function.

Forms

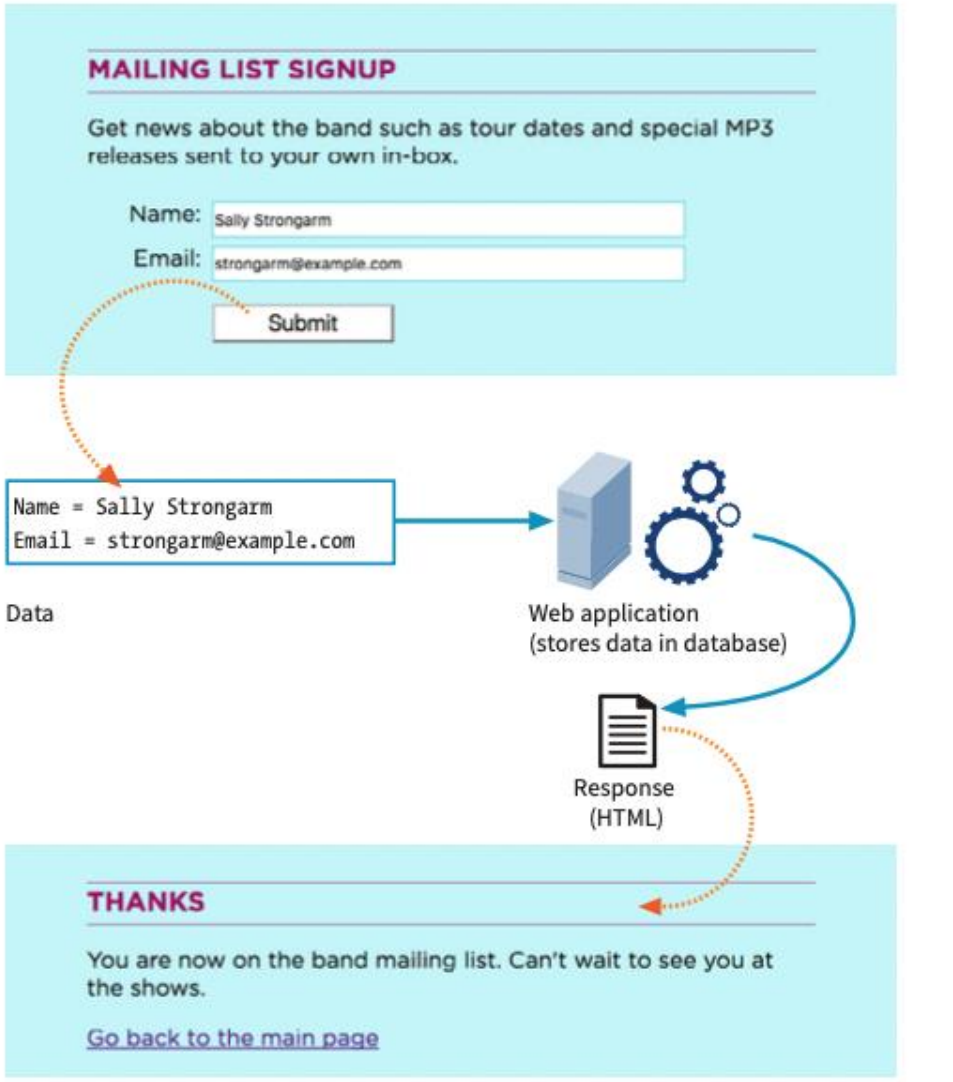


FIGURE 9-1. What happens behind the scenes when a web form is submitted.

Learning Web Design, Jennifer Niederst Robbins, O'Reilly Media, Inc., 2018, Page 179

Forms

The Form Element

The form element is a container for all of the form's content, including some form controls such as text-entry fields and buttons. It may also include block elements (h1, p, and lists, for example). It may, however, lack another form element.

Forms

The Action Attribute

The action attribute tells where the location the form will be process, it can be a application or a script The action attribute in this example sends the data to a script called process.php:

<form action="/process.php" method="POST">...</form>

The .php suffix tells that the form will be processed by a script that written in PHP scripting language, there in several technology that can processes a form such as:

1. PHP (.php)
2. Microsoft ASP (Active Server Pages; .asp)
3. Microsoft’s ASP.NET (Active Server Page; .aspx)
4. Ruby on Rails. JavaServer Pages (.jsp)
5. Python

Forms

The Method Attribute

The method attribute specifies how the information should be sent to the server. Let’s use this data as an example.

fullname = Alfred Tenggono

email = alfredteng@jiu.ac

When the browser encodes that information for its trip to the server, it looks like this:

fullname=Alfred+Tenggono&email=alfredteng%40jiu.acThere are only two methods for sending this encoded data to the server: POST or GET, indicated by the method attribute in the form element. The method is optional and will default to GET if omitted. We’ll look at the dif- ference between the two methods in the following sections.

Forms

With the GET method, the encoded form data gets tacked right onto the URL sent to the server. A question mark character separates the URL from the following data, as shown here:

get http://jiu.ac/mailinglist.php?name=Alfred+Tenggono&email=→ alfredteng%40jiu.ac

GET is inappropriate if the form submission performs an action, such as deleting something or adding data to a database, because if the user goes back, it gets submitted again.

Forms

The POST method

When the form's method is set to POST, the browser sends a separate server request containing some special headers followed by the data. In theory, only the server sees the content of this request, and thus it is the best method for sending secure information such as a home address or other personal information.

The POST method is also preferable for sending a lot of data, such as a lengthy text entry, because there is no character limit as there is for GET.

Forms

The name Attribute

The job of each form control is to collect one bit of information from a user. In the previous form example, text-entry fields collect the visitor's name and email address. To use the technical term, "fullname" and "email" are two vari- ables collected by the form. The data entered by the user ("Alfred Tenggono" and "alfredteng@jiu.ac") is the value or content of the variables.

The name attribute provides the variable name for the control. In this example, the text gathered by a textarea element is defined as the "comment" variable:

<textarea name="comment" rows="4" cols="45" placeholder="Leave us a comment."></textarea>

When a user enters a comment in the field ("This is the best band ever!"), it would be passed to the server as a name/value (variable/content) pair like this:

comment=This+is+the+best+band+ever%21 All form control elements must include a name attribute so the form process- ing application can sort the information. You may include a name attribute for submit and reset button elements, but they are not required, because they have special functions (submitting or resetting the form) not related to data collection.

Forms

Text-Entry Controls

Web form usually use for entering text information. What element you use to collect text input depends the type of the information whether it is a single line of text (input) or multiple lines (textarea).

<input type="text">

<input type="textarea">

Forms

Text-Entry Controls

Web form usually use for entering text information. What element you use to collect text input depends the type of the information whether it is a single line of text (input) or multiple lines (textarea).

Single-line text field

<input type="text” name”” Value=“” maxlenght=“”>

Multiline text-entry field

<textarea name=“” rows=“” cols=“”></textarea>

Multiline text-entry field with placeholder text

<textarea name=“” placeholder=“” rows=“” cols=“”></textarea>

Forms

Specialized Text-Entry Fields

Aside from the typical single-line text entry, there are a variety of input options for entering specialized information such as passwords, search phrases, email addresses, phone numbers, and URLs.

Forms

Specialized Text-Entry Fields

Password entry field

A password field functions similarly to a text-entry field, with the exception that the characters are masked from view by asterisk (*) or bullet (•) characters, or another character specified by the browser.

<input type="password" name=”userpass" maxlength="12" id="form-pswd">

Forms

Search, email, telephone numbers, and URLs

Prior to HTML5, the only option to collect email addresses, phone numbers, URLs, or search phrases was to add a generic text input field. The email, tel, url, and search input types in HTML5 advise the browser about the type of information to expect in the field. These input types have the same features as the basic text input type (name, maxlength, minlength, size, and value), as well as a few more.

Forms

Search, email, telephone numbers, and URLs

Search field

<input type="search">

Email address

<input type="email">

Telephone number

<input type="tel">

Location (URL)

<input type="url">

Forms

Submit and Reset Buttons

The submit button, when clicked or touched, immediately delivers the form data to the server for processing. A reset button restores the form controls to their original state when the form was first loaded. As previously stated, because these buttons perform specialized duties that do not require data entry, they are the only form control elements that do not require the name attribute.

Submits the form data to the server

<input type="submit">

Resets the form controls to their default settings

<input type="reset">

Forms

Radio buttons

Radio buttons are added to a form via the input element with the type attri- bute set to “radio.” Here is the syntax for a minimal radio button:

<input type="radio" name="variable" value="value">

Checkbox buttons

Checkboxes are added via the input element with its type set to checkbox. As with radio buttons, you create groups of checkboxes by assigning them the same name value. The difference, as we’ve already noted, is that more than one checkbox may be checked at a time.

<input type=“checkbox" name="variable" value="value">

Thank You

Alfred Tenggono, S.Kom., M.Kom.

alfred.tenggono@jiu.ac

Reference

- Learning Web Design, Jennifer Niederst Robbins, O’Reilly Media, Inc., 2018, ISBN: 978-1-491-96020-2