

# 3.1 HTML containers

Every HTML element that holds some part of the page content contains that content. A **container** is any part of a web document body that has opening and closing tags. Web developers typically create many containers as a convenience to assist in organizing and formatting content. For example, containers can be formatted by applying styles to adjust margins, padding, horizontal and vertical alignment, and other visual presentation attributes.

Table 3.1.1: Common HTML containers.

Container	Description
<header>	Container for introductory content
<footer>	Container for content descriptive information about the web page like author, copyright, or date modified
<section>	Container for distinct parts of a document, such as a chapter
<article>	Container for self-contained content that can be reused independently, such as a news article
<nav>	Container for content relating to website navigation
<aside>	Container for content not directly related to the main topic of a document
<div>	Generic tag for creating block containers
<span>	Generic tag for creating inline containers

A **parent container** is any container within which other containers are located.

PARTICIPATION  
ACTIVITY

3.1.1: Containers and parent containers.

©zyBooks 02/17/19 22:08 458368  
jingya xun



SANDIEGOCOMP494GappySpring2019

## Animation captions:

1. The <p> tag contains the text "The top-selling board games of all time are:".
2. The <ol> tag is a parent container for all list items.
3. Each <li> tag is the container for one list item.

A **block** tag's content tries to fill the width of the tag's parent container and can contain other block tags, inline tags, and text. Block tags include headings, tables, and paragraphs.

Some block tags cannot be contained within certain other block tags when the semantics are unclear. Ex: The `<p>` tag cannot contain another `<p>` tag.

A block tag is typically displayed starting and ending on new lines. Ex: The `<ol>` tag is a block tag that fills the entire width of the parent container, and each ordered list starts on a new line separate from previous and following blocks.

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019

#### PARTICIPATION ACTIVITY

#### 3.1.2: Parent containers.



Given the following HTML :

```
<ol>
  <li></li>
  <li></li>
  <li></li>
</ol>
```

1) The `<ol>` tag is a parent container.

- True
- False



2) The `<li>` tag is a parent container.

- True
- False



3) The `<img>` tag is a parent container.

- True
- False



A `<div>` tag is a generic tag for creating block containers to facilitate managing page content, and is the only block tag with no semantic meaning, unlike other block tags such as `<p>` and `<table>`.

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019

#### Note

*By default, web browsers do not display container borders. For clarity, the container borders are visible in the following examples.*

#### Example 3.1.1: Block tags with visible containers

<https://learn.zybooks.com/zybook/SANDIEGOCOMP494GappySpring2019/chapter/3/print>

**EXAMPLE 3.1.1. BLOCK TAGS WITH VISIBLE CONTAINERS.**

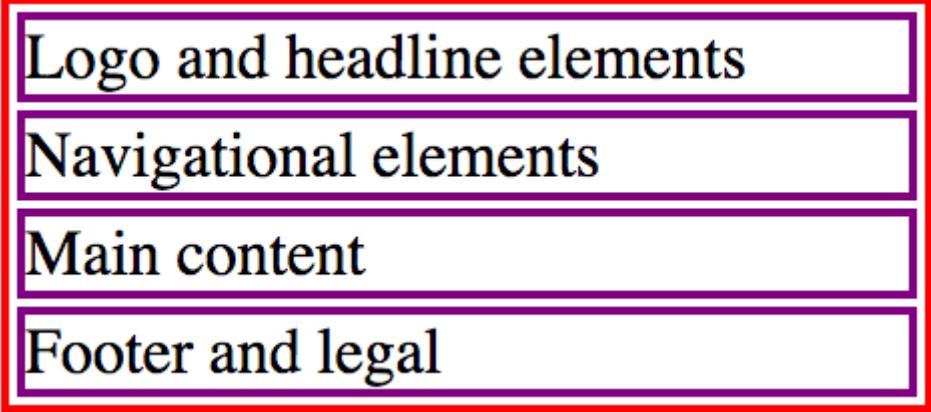
The following HTML breaks up a document into logical components using a <div> tag for each component. The corresponding image displays the border for each container.

```
<body>
  <div>Logo and headline elements</div>
  <div>Navigational elements</div>
  <div>Main content</div>
  <div>Footer and legal</div>
</body>
```

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019



**Logo and headline elements**

**Navigational elements**

**Main content**

**Footer and legal**

**PARTICIPATION ACTIVITY**

3.1.3: Block tags.



- 1) Which tag is a generic block tag without any implied meaning?

- <block>
- <div>
- <section>



- 2) What is the width of a block tag?

- The minimum width necessary to hold the block's contents.
- The full width of the web page.
- The full width of the block's parent container.



©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019

An **inline** tag's content tries to fill the minimum space possible in the tag's parent container and can only contain text or other inline tags. For example, the <a> tag is an inline tag that creates a hyperlink container as big as the link's internal content; a hyperlink does not fill the width or height of the link's parent paragraph.

The **<span>** tag is the generic tag for creating inline containers to facilitate managing content on the page. Unlike other inline tags, such as **<a>** and **<em>**, the **<span>** tag has no semantic meaning.

### Example 3.1.2: Span tags with visible containers.

The following HTML breaks up a the second paragraph from Lincoln's Gettysburg Address into sentences using a **<span>** tag for each component. The corresponding image displays the border for each container.

```
<p>
<span>Now we are engaged in a great civil war testing whether that
nation, or any nation so conceived and so dedicated, can long
endure.</span>
<span>We are met on a great battle-field of that war.</span>
<span>We have come to dedicate a portion of that field, as a final
resting place for those who here gave their lives that that nation might
live.</span>
<span>It is altogether fitting and proper that we should do this.</span>
</p>
```

Now we are engaged in a great civil war,  
 testing whether that nation, or any nation so  
 conceived and so dedicated, can long endure.  
 We are met on a great battle-field of that  
 war. We have come to dedicate a portion of  
 that field, as a final resting place for those  
 who here gave their lives that that nation  
 might live. It is altogether fitting and proper  
 that we should do this.

#### PARTICIPATION ACTIVITY

#### 3.1.4: Block and inline tags.

Given the following HTML :

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019

```
<p>Click on the "swoosh" to go to Nike Headquarters. <br>
<a href="http://nike.com"></a></p>
```

- 1) The **<p>** tag is an inline tag.

- True

False

2) The &lt;img&gt; tag is an inline tag.

 True False

3) The &lt;a&gt; tag is an inline tag.

 True False

4) The &lt;br&gt; tag is an inline tag.

 True False

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019

**PARTICIPATION ACTIVITY**

## 3.1.5: Hierarchy of block and inline tags.

1) What types of tags can be inside a block tag?

 Only inline tags Only block tags Both inline and block tags.

2) What types of tags can be inside an inline tag?

 Only inline tags Only block tags Both inline and block tags

Since <div> and <span> do not have semantic meaning, these tags are used primarily for presentation and interaction purposes. Good practice is to use tags such as <address> and <article> that convey semantic meaning when creating containers, and use <div> and <span> only when no other tags are appropriate.

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019

**PARTICIPATION ACTIVITY**

## 3.1.6: Block vs. inline.

Change the <div> tags to <span> tags in the following HTML fragment to see the difference between block and inline layout.

[HTML](#)[CSS](#)

```
1 <section>
2   <div>According to Wikipedia, J.R.R. Tolkien's <u>The Lord of the Rings</u> is the s
3   <div>Wikipedia states that "only A Tale of Two Cities by Charles Dickens has sold m
4   <div>Tolkien's <u>The Hobbit</u> is the fourth best-selling novel of all time.</div>
5 </section>
```

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019

[Render web page](#)[Reset code](#)

### Your web page

According to Wikipedia, J.R.R. Tolkien's The Lord of the Rings is the second best-selling novel ever written.

Wikipedia states that "only A Tale of Two Cities by Charles Dickens has sold more copies worldwide (over 200 million)".

Tolkien's The Hobbit is the fourth best-selling novel of all time.

PARTICIPATION  
ACTIVITY

3.1.7: Container structure.

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019

Use the following visible containers created with the provided HTML below to answer the questions.

```

<body>
  <header>
    <h1>The White House</h1>
  </header>
  <section>
    <p>
      The White House is the official residence of the
      President of the United States.
    </p>
    
    <address>
      White House<br>
      1600 Pennsylvania Avenue Northwest<br>
      Washington, DC 20500
    </address>
  </section>
</body>

```

©zyBooks 02/17/19 22:08 458368  
jingya xun  
SANDIEGOCOMP494GappySpring2019

# The White House

The White House is the official residence  
of the President of the United States.



**White House**  
**1600 Pennsylvania Avenue Northwest**  
**Washington, DC 20500**

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019



- 1) Which tag creates the container that holds the entire web page?

- <body>
- <header>
- <section>

2) Which tag creates the container that holds the picture?

- <body>
- <section>
- <img>

3) Which tag creates the container that holds the White House address?

- <body>
- <section>
- <address>

4) Which tag creates the parent container that holds the paragraph's container?

- <body>
- <section>
- <p>

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019



**CHALLENGE  
ACTIVITY**

3.1.1: HTML containers.

How was this section?



[Provide feedback](#)

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019

## 3.2 Forms

The **<form>** tag allows the web browser to send information from the user to the server. The **<form>** tag has two primary attributes:

1. The **action** attribute indicates the URL where the form data should be sent.
2. The **method** attribute indicates the HTTP request type the browser will use to communicate with the server. The method is either GET or POST. GET is the default method and is used if no method is specified.

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019

Figure 3.2.1: Partial HTML form sending data to the Twitter home page using a secure HTTP POST request.

```
<form action="https://twitter.com/" method="POST">
...
</form>
```

#### PARTICIPATION ACTIVITY

3.2.1: Submitting form data to a server.



### Animation captions:

1. The user enters information in the form.
2. The user clicks on the submit button.
3. The browser collects the data and send the data to the server.

When a user clicks the submit button in a form that uses the GET method to contact the server, the browser performs the following steps:

1. Collect all data from the form fields into a query string. The **query string** is a set of name=value pairs separated by the ampersand character (&). Each name is specified as an attribute of the HTML field, and the value is the user-entered data. Ex: The last and first field names and values in the above animation become the string:  
**last=Conner&first=Sarah**
2. Create a URL with the server page and name=value pairs. The URL is composed of the action attribute specified in the form, the question mark character (?) and the query string. Ex: **http://example.com/apply?last=Conner&first=Sarah**
3. Use the newly created URL to contact the server using an HTTP GET request.
4. Display or update the web page using the response received from the server.

#### PARTICIPATION ACTIVITY

3.2.2: Using the GET method to submit form data to a server.



### Animation captions:

1. The user enters information in the form.
2. The user clicks on the submit button.
3. Browser creates a query string of name=value pairs for each form field, separated by &.
4. The query string is appended to the URL from the form action.
5. The new URL is used to connect to the web server.

When the user clicks the submit button in a form that uses the POST method to contact the server, the browser creates the query string from the form data, uses the URL from the form's action attribute to contact the server using an HTTP POST request, and sends the query string in the HTTP request message body.

If a form field contains binary data such as an image, the normal format of the query string is not sufficient to encode the binary data. Ex: The binary data may include a byte that is the same as an ampersand. The server would interpret that byte as a separator between two fields, but the browser meant the byte to be part of the binary data. To accommodate binary data, a POST request can be split into multiple parts. An **enctype** attribute of "multipart/form-data" indicates the web browser should split a POST request into multiple parts, where each input field is sent as a separate part of the HTTP request message. If enctype is "multipart/form-data", the form data is not sent to the server as a query string.

**PARTICIPATION ACTIVITY**

3.2.3: Using the POST method to submit form data to a server.

**Animation captions:**

1. The user enters information in the form, including a picture to upload to the server.
2. The user clicks the submit button.
3. The browser collects the form data into multiple parts, and adds each part to the POST request.
4. The POST request is then sent to the server.

**PARTICIPATION ACTIVITY**

3.2.4: <form> tags.



Refer to the HTML below.

```
<form action="https://google.com/">  
</form>
```

©zyBooks 02/17/19 22:08 458368  
jingya xun

SANDIEGOCOMP494GappySpring2019



- 1) To which server will the browser send the form data?

**Check**

**Show answer**



- 2) Which method will the browser use to communicate with the server?

**Check****Show answer**

©zyBooks 02/17/19 22:08 458368

A **widget** is an interactive component (usually graphical) that the browser uses to interact with a user. Ex: Buttons, drop-down menus, and data entry fields.

SANDIEGOCOMP494GappySpring2019

The **<input>** tag allows the user to enter information into a web page. The **<input>** tag cannot enclose any additional page content, and thus does not have a closing tag. The **<input>** tag has five primary attributes: **type**, **name**, **id**, **placeholder**, and **value**. The **type** attribute indicates what type of widget to display. The default type is "text" if **type** is not specified. A **text box** widget is an **input** element with the **type** attribute of "text" that allows users to enter a single line of text.

The web browser displays a **submit button** widget for an **<input>** tag with the **type** attribute of "submit", which sends the associated form's data to the server when clicked. A submit button uses the **value** attribute to specify the button's text.

The HTML below expands the previous example to ask for a message to tweet. The text box widget does not use the **value** attribute because no default tweet message would make sense. The submit button does not use the **id**, **name**, or **placeholder** attributes because neither the browser nor the server need to keep track of the submit button.

Figure 3.2.2: Complete HTML form sending status to the Twitter home page using a secure HTTP POST request.

```
<form action="https://twitter.com/" method="POST">
  <input type="text" name="status" id="status" placeholder="Your status">
  <input type="submit" value="Send status">
</form>
```

Your status	Send status
-------------	-------------

**PARTICIPATION ACTIVITY**
**3.2.5: <input> attributes.**

©zyBooks 02/17/19 22:08 458368

jingya.xun



SANDIEGOCOMP494GappySpring2019

Match each **<input>** attribute to the corresponding effect.

<b>type</b>	<b>name</b>	<b>value</b>	<b>placeholder</b>	<b>id</b>
-------------	-------------	--------------	--------------------	-----------

Indicates which kind of widget is displayed by the browser.

Allows the server to identify which data came from which widget.

Uniquely identifies the specific input tag to the browser.

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019

Allows the input to start with a default value.

Provides a hint to the user about the information being requested.

**Reset**

The **<label>** tag displays descriptive text associated with a specific widget. A label has a **for** attribute whose value should match the **id** attribute for the widget being labeled.

Figure 3.2.3: HTML for a label associated with a text box.

```
<label for="username">Username:</label>
<input type="text" id="username">
```

Username:

A **text area** widget is an input element specified by **<textarea>** opening and closing tags that allows users to enter multiple lines of text. A **<textarea>** tag has optional **rows** and **cols** attributes to specify the initial size of the text area.

**PARTICIPATION**  
**ACTIVITY**

3.2.6: Text inputs.



The following HTML form contains a text box, text area, and submit button. Pressing the submit button submits the form data to `form-viewer.php`, which displays the submitted form data.

1. Try adding another text box with a **name** of `last`. Also change the size of the text area so that the text area has 7 rows and 50 columns of text. Make your web page match the expected web page.

2. Type some data into the form and press Submit. Note that the form data appears in the query string of the resulting web page because the form uses the GET method.
3. Change the form's **method** from "GET" to "POST". Render the web page, type some data into the form, and press Submit. The form data no longer appears in the query string of the resulting web page although the submitted data is present.

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019



**PARTICIPATION  
ACTIVITY**

3.2.7: Text inputs.



1) Which tag creates a text box widget?



- <input>
- <text>
- <textarea>

2) Which attribute must be set to create a text box widget?



- type
- no attribute
- name
- id

3) Which tag creates a text area widget?



- <input>
- <text>
- <textarea>

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019

4) Which attribute inside the <label> tag is used to associate the label with a widget?



- id
-

for

 name**CHALLENGE  
ACTIVITY**

3.2.1: Building forms.

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019



Exploring further:

- [enctype attribute](#) from W3Schools
- [Sending form data](#) from Mozilla Developer Network

How was this section?

[Provide feedback](#)

## 3.3 Common widgets

A **checkbox** is a widget for input elements with the `type` attribute of "checkbox", which allows users to check, or select, a value. A checkbox initially appears selected if the "checked" attribute is set. For each checkbox selected, the browser sends the checkbox's name and value to the server. If the value attribute is not specified, the default value of "on" is sent. If a checkbox is not selected, the browser does not send anything to the server. A [common error](#) on the server is failing to record all checkboxes in the form as the browser doesn't report any values for checkboxes not selected by the user.

Good practice is to use label elements with checkboxes so the user can click the label to check and uncheck the associated checkbox.

**PARTICIPATION  
ACTIVITY**

3.3.1: Submitting checkboxes to the server.



## Animation captions:

1. Each checkbox is displayed with a corresponding label.
2. The user checks Item 1 and Item 3 checkboxes, and clicks the submit button.
3. For each selected checkbox, the checkbox's name and the value "on" are sent to the server.

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019



PARTICIPATION  
ACTIVITY

3.3.2: Checkbox inputs.

The following HTML fragment contains a checkbox which is selected. Click the submit button to view the form data sent to server. Try adding another checkbox with the name second.



A **radio button** is a widget for input elements with the `type` attribute of "radio", which allows users to select exactly one value from possibly many values. The web browser groups radio buttons together with the same `name` attribute, where each possible value in a group has an associated input. When submitting a form, the browser sends the selected radio button's `name` and `value` attribute. Ex: If the radio button

`<input type="radio" name="movie" value="ET">` is selected, "movie=ET" will be sent in the HTTP request.

The main difference between a radio button and checkbox widget is that only one radio button in a group can be selected, while any number of checkboxes can be selected.

PARTICIPATION  
ACTIVITY

3.3.3: Radio buttons.

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019



Try adding another radio option for the restaurants group with a value of Pizza Hut. Click the submit button to view the form data sent to server. Then, try adding another set of radio buttons to match the expected rendered HTML.

**PARTICIPATION  
ACTIVITY****3.3.4: Checkboxes and radio buttons.**

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019



Refer to the HTML below.

```
<form>
  <input type="checkbox" name="check1" checked>
  <input type="checkbox" name="check2">
  <input type="radio" name="radio1" value="1" checked>
  <input type="radio" name="radio2" value="2">
  <input type="radio" name="radio3" value="3">
  <input type="submit">
</form>
```



- 1) If the user clicks submit without changing anything in the form, how many items will be sent to the server?

**Check****Show answer**

- 2) What is the maximum number of items that can be sent to the server?

**Check****Show answer**

- 3) What is the minimum number of items that will be sent to the server?

**Check****Show answer**

- 4) Which attribute should be added for each widget to add labels to the form?

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019



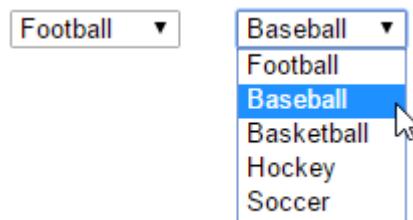
[Check](#)[Show answer](#)

The **<select>** opening and closing tags create a **drop-down menu** (or **drop-down list**), which allows users to select one of several predefined values. The **<option>** opening and closing tags create a value, or option, the user can select within a drop-down menu. When the user is not interacting with the menu, the drop-down menu usually displays the selected option.

The difference between a drop-down menu and a radio button widget is that the drop-down menu only displays the options when interacting with the user, while a radio button widget always displays all options.

Figure 3.3.1: Drop-down menu's default appearance (left) and when selecting an option (right).

```
<select name="sport">
  <option value="football">Football</option>
  <option value="baseball">Baseball</option>
  <option value="basketball">Basketball</option>
  <option value="hockey">Hockey</option>
  <option value="soccer">Soccer</option>
</select>
```



A **list box** widget is created by specifying a size with the select element's **size** attribute. Ex: `<select size="4">` creates a list box that shows four options at a time. If the list box contains more than **size** options, the browser adds a vertical scrollbar so the user can scroll through the list of options.

The **multiple** attribute allows the user to select multiple options. On Windows, the user must hold down the control (Ctrl) button to select multiple options, and on a Mac, the user must hold down the command button. Many users are unaware of how to choose multiple options from a list box, so good practice is to use checkboxes instead.

Figure 3.3.2: List box that allows multiple options to be selected.

```
<select name="flagcolors" size="4" multiple>
<option value="red">Red</option>
<option value="orange">Orange</option>
<option value="yellow">Yellow</option>
<option value="green">Green</option>
<option value="blue">Blue</option>
<option value="purple">Purple</option>
<option value="white">White</option>
<option value="black">Black</option>
</select>
```



©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019

A **button** widget can be created using the `<button>` opening and closing tags or with `<input type="button">`. The `<button>` element allows text and images to be displayed in a button, but an `<input>` button only allows text.

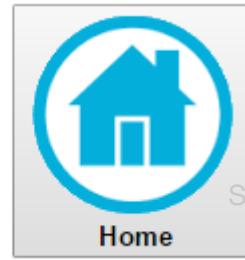
The `<button>` element has a `type` attribute that can be set to various values like "button" or "submit". The "button" type is typically used with JavaScript to perform an action when clicked. The "submit" type creates a submit button for a form. If the `type` attribute is not specified, different browsers may choose different default types, so good practice is to always specify the type.

Figure 3.3.3: HTML buttons.

```
<input type="button" value="Home">

<button type="button">
  <br>
  <strong>Home</strong>
</button>
```





©zyBooks 02/17/19 22:08 458368

jingya xun

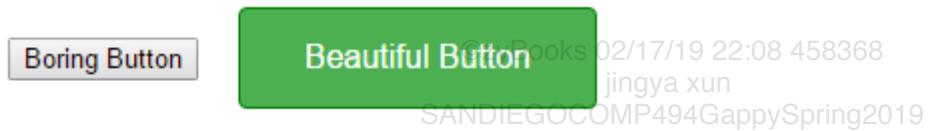
SANDIEGOCOMP494GappySpring2019

## Styling widgets

*The default look of a form's widgets may differ depending on the browser and operating*

system. Developers often use CSS to style form widgets to give the widgets a more uniform look or to increase the widgets' visual appeal.

The image on the left shows the default rendering of a button in Chrome. The button on the right has been styled with CSS.



A **password field** is a widget for input elements with the `type` attribute of "password", which allows users to enter a password without the password contents being displayed on-screen. Web browsers usually provide facilities to remember passwords at various websites to help users.

The HTML below uses the `size` attribute to limit the password field's width and uses the `maxlength` attribute to limit the maximum number of characters the user can enter. The `size` and `maxlength` attributes can be used on text boxes as well.

Figure 3.3.4: Password field that limits the number of characters to 10.

```
<input type="password" name="secret" size="10" maxlength="10">
.....
```

**PARTICIPATION ACTIVITY** 3.3.5: Menus, buttons, and passwords.

- 1) A drop-down menu only allows one option to be selected.

- True
- False

- 2) An <option> tag must have a `value` attribute.

- True
- False

- 3) Which element can create a button with an image?

- <button>
- <input>

©zyBooks 02/17/19 22:08 458368  
jingya xun  
SANDIEGOCOMP494GappySpring2019



4) Buttons always submit form data to a server.

True

False

5) Passwords from the password widget that are sent to the server using HTTP are safe from prying eyes.

True

False

6) Which input element attribute limits the number of characters the user can type in a text box or password field?

size

maxlength

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019

**CHALLENGE  
ACTIVITY**

3.3.1: Building common widgets.



Exploring further:

- [HTML <input> Tag](#) from W3Schools
- [HTML <select> Tag](#) from W3Schools
- [HTML <button> Tag](#) from W3Schools

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019

How was this section?



[Provide feedback](#)

## 3.4 HTML5 widgets

### Introduction to HTML5 widgets

©zyBooks 02/17/19 22:08 458368

HTML5 introduces many new features and helps standardize input by adding new widgets and attributes. These new widgets and attributes reduce web page development time and improve user experience, especially for mobile device browsers.

Before HTML5, many web developers built their own custom solutions for handling special types of input such as dates. Because HTML5 provides standard solutions for the most commonly needed special input types, developers save significant time by using the standard built-in widgets. New widget attributes and built-in field validation for many input types further reduce work for developers.

The traditional diversity of special input type solutions made the user experience inconsistent across websites. HTML5 input standardization improves the quality and consistency of user experience across websites and gives browsers a standard way of identifying and supporting specific types of input (Ex: display customized keyboard for entering telephone numbers).

#### Example 3.4.1: HTML5 telephone input using desktop and mobile browsers.

The HTML5 telephone input (type="tel") does not appear to provide any special benefit when viewed using a traditional desktop browser. Because there is such a wide range of possible input values worldwide, HTML5 does not attempt input validation.

Your mobile number:

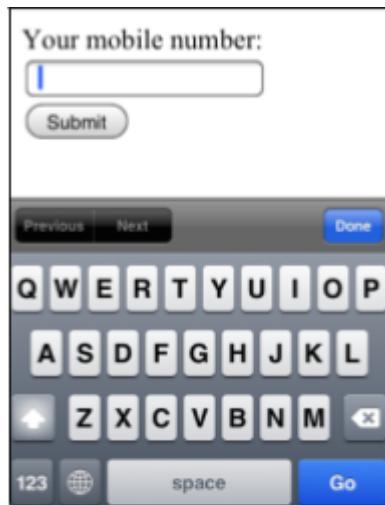
Desktop browser:  
Text input

Your mobile number:

Desktop browser:  
Telephone input

However, the HTML5 telephone input shows real value in a mobile device browser. A keyboard customized for telephone number input is displayed when the user touches the input area.



Mobile browser:  
Text input  
Displays generic keyboard



Mobile browser:  
Telephone input  
Displays telephone keyboard

## Input pickers

The new widgets include input pickers. An input **picker** is a widget that allows the user to interactively pick a choice using a popup or other guided selection method.

### The date input picker

The date widget is a frequently used input picker. The date input box displays a date input template and lets the user specify the date in one or more ways, including a popup date selector.

The basic syntax for the date input is `<input type="date">`. Two attributes for the date input are min (the earliest date permitted) and max (the latest date permitted).

Example 3.4.2: HTML5 date picker.

©zyBooks 02/17/19 22:08 458368  
jingya xun  
SANDIEGOCOMP494GappySpring2019

Select an appointment date:

mm / dd / yyyy

November 2016

Sun	Mon	Tue	Wed	Thu	Fri	Sat
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3

## The color input picker

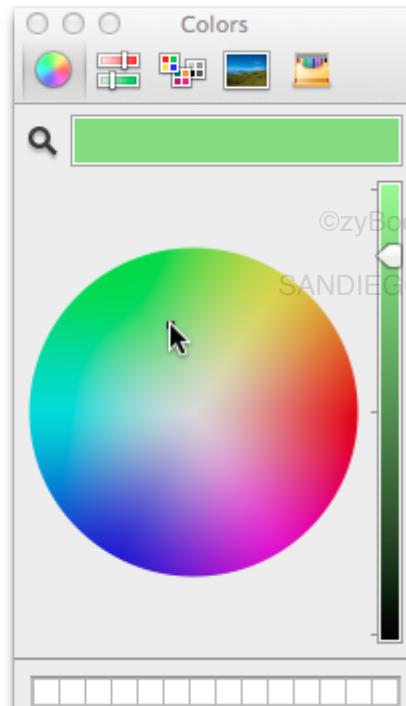
Clicking on the `color` input picker creates a color selector pop up that helps the user explore and choose a color. Requiring the user to enter a hex string such as `#4268D3` into a text input field is confusing for most users, so the color picker introduces a large improvement.

The basic syntax for the color input is `<input type="color">`.

Example 3.4.3: HTML5 color picker input.

©zyBooks 02/17/19 22:08 458368  
jingya xun  
SANDIEGOCOMP494GappySpring2019

Select a color:



## The number input

The `number` input ensures user input is a valid number.

The basic syntax for the number input is `<input type="number">`.

Example 3.4.4: HTML5 number input.

Enter a value  $\geq 0$  and  $\leq 212$ .

! Value must be less than or equal to 212.

The number widget attributes are used to automatically validate input.

## The range input

The `range` input control allows the user to select a value by dragging a sliding control along the length of a line.

The basic syntax for the range input is `<input type="range">`. Three commonly used attributes for the range input are the min, max and currently selected values.

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019

### Example 3.4.5: HTML5 range input.

Select a value using the slider



**PARTICIPATION  
ACTIVITY**

3.4.1: Some additional HTML5 widgets.

©zyBooks 02/17/19 22:08 458368

jingya xun



SANDIEGOCOMP494GappySpring2019

**time      month      url      email      datetime-local**

Allows input of a date and time.

Allows input of an email address.

Allows selecting a month and year.

Allows entering a time more easily.

Allows entering URLs.

**Reset**

**PARTICIPATION  
ACTIVITY**

3.4.2: Enter the missing HTML5 type names.



- 1) Allow the user to enter a day, month, and year.

<input type="\_\_\_\_\_">

**Check**

[Show answer](#)

©zyBooks 02/17/19 22:08 458368  
jingya xun



SANDIEGOCOMP494GappySpring2019

- 2) Allow the user to enter a month, and year.

<input type="\_\_\_\_\_">

**Check**

[Show answer](#)

3)



Allow the user to select some combination of red, blue, and green.

```
<input type="text">
```

[Check](#) [Show answer](#)

- 4) Allow the user to use a slider to select a number between 0 and 10.

```
<input type="range"
min="0" max="10">
```

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019



[Check](#) [Show answer](#)

- 5) Ensure the user types a valid number between 0 and 10.

```
<input type="number"
min="0" max="10">
```

[Check](#) [Show answer](#)



## Common HTML5 attributes

Attributes introduced in HTML5 include the examples listed in the table below.

Table 3.4.1: HTML5 input attributes.

Attribute Name	Description	Example
max	Sets the maximum value that the input can have. E.g. <code>max="212"</code> limits the input value to a maximum of 212.	<code>&lt;input type="number" max="212"&gt;</code> ©zyBooks 02/17/19 22:08 458368 jingya xun
min	Sets the minimum value that the input can have. E.g. <code>min="1976-07-04"</code> limits the date input value to no earlier than the U.S. bicentennial date, July 4, 1976.	<code>&lt;input type="date" min="1976-07-04"&gt;</code> SANDIEGOCOMP494GappySpring2019

pattern	<p>Provides a pattern (called a regular expression) that input must match. E.g.</p> <p><b>pattern=" [ABC ] [ 0-9 ] "</b> only permits entering a two-character value that first has A, B, or C, and is then followed by a single digit.</p>	<pre>&lt;input type="text" pattern=" [ABC ] [ 0-9 ] "&gt;</pre> <p>©zyBooks 02/17/19 22:08 458368 jingya xun SANDIEGOCOMP494GappySpring2019</p>
required	<p>States that input is required and the field must not be left empty. E.g. <b>required</b></p>	<pre>&lt;input type="password" required&gt;</pre>
step	<p>Sets amount by which the value can change. E.g., <b>step="5"</b> means that the value can only change by multiples of 5.</p>	<pre>&lt;input type="range" step="5"&gt;</pre>

**PARTICIPATION ACTIVITY**

3.4.3: Enter the missing HTML5 attribute names.



- 1) Ensure that the value will change by multiples of 100.

```
<input type="range"
      = "100">
```

**Check****Show answer**

- 2) Ensure that the user types only one letter in the range A through F.

```
<input type="text"
      = "[A-F]">
```

©zyBooks 02/17/19 22:08 458368  
jingya xun

SANDIEGOCOMP494GappySpring2019



- 3) Ensure that the temperature entered will not be less than -273.16.

```
<input type="number"
      = "-273.16">
```



[Check](#)[Show answer](#)

- 4) Ensure that the date entered will not be greater than the end of 2025.

```
<input type="date"  
      value="2025-12-31">
```

[Check](#)[Show answer](#)

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019



- 5) Ensure the user enters a value in the input field.

```
<input type="number"  
      value=">
```

[Check](#)[Show answer](#)

## Warning

*Even though the HTML5 standard was published in late 2014 and the new widgets and attributes introduced in HTML5 section are very useful, these widgets and attributes are not fully supported by all browsers.*

## Fallbacks and polyfills

A **fallback** is a mechanism that allows a web page element to function correctly even if the browser does not support a particular element. Good practice is to implement a fallback mechanism if a particular HTML5 widget is not widely supported by browsers at the time.

A **polyfill** is a fallback using JavaScript code that makes certain features of HTML5 (Ex: the date picker) work on browsers that do not natively support those HTML5 features. Developers often use a JavaScript library such as Modernizr to detect which features the browser does not support, and then load one or more polyfills to provide fallback mechanisms for the non-supported HTML5 features.

©zyBooks 02/17/19 22:08 458368  
jingya xun  
SANDIEGOCOMP494GappySpring2019

### Example 3.4.6: An HTML5 date polyfill.

When the date input element is not supported by a particular browser, the browser treats the date input box like an ordinary text input field.

```
<form>
  Select an appointment date:<br>
  <input type="date" name="appointmentDate">
  <br>
  <input type="submit">
</form>
```

Select an appointment date:

**Submit**

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019

[Unsupported date input](#)

The date picker problem can be solved by detecting the problem using a library called Modernizr and adding a date picker from one of the widely-used JavaScript libraries such as jQuery.

The highlighted lines are responsible for loading the necessary Modernizr and jQuery files and for activating the polyfill if the browser does not support the date input. Notice that the original HTML form contents are not modified.

```
<link href="https://cdnjs.cloudflare.com/ajax/libs/jqueryui/1.12.0/jquery-ui.min.css" rel="stylesheet">
<script src="https://cdnjs.cloudflare.com/ajax/libs/modernizr/2.8.3/modernizr.min.js"></script>
<script src="https://code.jquery.com/jquery-3.1.0.min.js"></script>
<script src="https://code.jquery.com/ui/1.12.0/jquery-ui.min.js"></script>

<script>
$(function() {
  if (!Modernizr.inputtypes['date']) {
    $('input[type=date]').datepicker({
      dateFormat: 'mm-dd-yy'
    });
  }
});
</script>

<form>
  Select an appointment date:<br>
  <input type="date" name="appointmentDate">
  <br>
  <input type="submit">
</form>
```

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019

**Select an appointment date:**

The image shows a date picker for November 2016. The days of the week are labeled Su, Mo, Tu, We, Th, Fr, Sa. The dates are arranged in four rows: Row 1 (1-5), Row 2 (6-12), Row 3 (13-19), and Row 4 (20-26). Row 5 contains the dates 27, 28, 29, and 30. A tooltip at the bottom right of the calendar area contains the text: "©zyBooks 02/17/19 22:08 458368 jingya xun SANDIEGO COMP494GappySpring2019".

Fallback date input: uses Modernizr, and  
the jQuery date picker

**PARTICIPATION ACTIVITY**

3.4.4: Questions about HTML5 widgets and attributes.



- 1) The range and number widgets are interchangeable choices for entering numbers.
  - True
  - False
  
- 2) A polyfill is code that creates a browser fallback for a feature if the web browser does not support that feature.
  - True
  - False
  
- 3) Browser fallbacks are only used to make new features work on old browsers.
  - True
  - False
  
- 4) Having users enter dates, colors, email, and numbers using the specialized widgets is preferable when an appropriate widget is available.
  - True



©zyBooks 02/17/19 22:08 458368  
jingya xun  
SANDIEGO COMP494GappySpring2019



False

- 5) It is better to avoid new HTML5 elements and attributes because of incomplete support for some new HTML5 features.

True

False

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019

Exploring further:

- W3 Consortium official date input element description
- Wikipedia article on Modernizr
- Wikipedia article on polyfills
- Check browser feature support at caniuse.com

How was this section?



[Provide feedback](#)

## 3.5 Audio and video

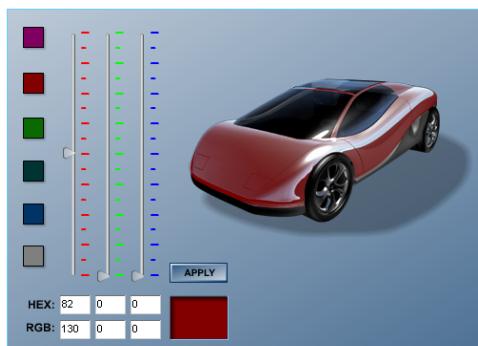
### Browser plug-ins

A website may play media files consisting of audio or video for website visitors. Prior to HTML5, developers used **<embed>** and **<object>** elements to embed audio or video in a web page. The embedded audio or video required browser plug-ins to play the media. A browser **plug-in** is software that can properly read and interpret a file format that the browser cannot. Ex: The Flash Player plug-in allows a browser to play Flash files. Plug-ins have traditionally been a source of security risks for web browsers, and some browsers have stopped supporting some types of plug-ins.

jingya xun  
SANDIEGOCOMP494GappySpring2019

Figure 3.5.1: Flash example (left) and the same example when the Flash plug-in is unavailable (right).

The color of a display object can be changed dynamically at runtime using the ColorTransform object and the built-in transform property. Applying color dynamically is useful for situations where you need to change the state of a graphic but don't want to use multiple images to do so.



The color of a display object can be changed dynamically at runtime using the ColorTransform object and the built-in transform property. Applying color dynamically is useful for situations where you need to change the state of a graphic but don't want to use multiple images to do so.



### Drawing sample from Adobe

```
<!-- Embedded Flash -->
<object type="application/x-shockwave-flash" data="4_colorb1.swf" width="520" height="358">
  <param name="play" value="true">
  <param name="menu" value="true">
  <param name="scale" value="showall">
</object>
```

#### PARTICIPATION ACTIVITY

#### 3.5.1: Browser plug-ins.



- 1) A plug-in is often required to play media embedded in a web page using `<embed>` or `<object>`.

- True
- False



- 2) Some plug-ins are not available for all web browsers.

- True
- False



## Audio element

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019

The HTML5 `<audio>` element plays an audio file in a web page. The `<source>` element is used inside the `<audio>` tag to specify an audio file to play.

#### PARTICIPATION ACTIVITY

#### 3.5.2: Audio element.



The `<audio>` tag loads an MP3 file using the `<source>` tag. The `controls` attribute displays audio controls for the user. The text "Your browser does not support the audio element" message only displays on older browsers that do not support the `<audio>` tag.

1. Press the play button to play the audio and the pause button to pause the audio.
2. Add the `autoplay` attribute to the `<audio>` tag and render the web page. Note how the MP3 plays automatically.
3. Add the `loop` attribute to the `<audio>` tag and render the web page. The audio will restart upon reaching the end of the audio.
4. Add the `muted` attribute to the `<audio>` tag and render the web page. Note how the audio is initially muted.

Vivaldi meets hip hop by typos: Creative Commons Attribution License



Different web browsers support different audio formats, so multiple `<source>` tags can be used to supply alternate file formats. The MP3 and AAC formats have wide browser support.

Table 3.5.1: Common audio formats.

Format	File extension	Description
AAC (Advanced Audio Coding)	.aac	Designed to be the successor of the MP3 format with better sound quality.
MIDI (Musical Instrument Digital Interface)	.midi .flv	Main format for electronic music devices that store notes; does not contain sound.
MP3	.mp3	The sound portion of an MPEG file. Most popular format for music players.
MP4	.m4a .mp4	Developed by the Moving Pictures Expert Group. m4a is audio only, but mp4 may contain video.

Ogg	.ogg	Open container format developed by Xiph.Org Foundation.
Wave (Waveform Audio File Format)	.wav	Developed by Microsoft and IBM, mainly for storing uncompressed audio on Windows.
WMA (Windows Media Audio)	.wma	Developed by Microsoft for Windows music players. ©zyBooks 02/17/19 22:08 458368 jingya xun

SANDIEGOCOMP494GappySpring2019

**PARTICIPATION**  
**ACTIVITY**

## 3.5.3: Audio element.



- 1) The audio element's audio can always be paused by the user.

True  
 False

- 2) The audio element can play audio as soon as the audio file has downloaded to the browser.

True  
 False

- 3) The AAC and MP3 formats are supported by nearly all web browsers.

True  
 False

- 4) A browser that is only capable of playing MP3 audio will download which file?

```
<audio controls>
  <source src="sound.ogg">
  <source src="sound.mp3">
</audio>
```

sound.mp3  
 sound.ogg

©zyBooks 02/17/19 22:08 458368  
jingya xun

SANDIEGOCOMP494GappySpring2019

**Video element**

The **<video>** element displays a video in a web page. The **<source>** element is used in a **<video>** tag to specify the name of the video file to play.

**PARTICIPATION  
ACTIVITY**

3.5.4: Playing a video.



The **<video>** tag uses the **controls** attribute to display controls to play and pause a video and change the volume.

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019

1. Press the play button to play the video and the pause button to pause the video.
2. Change the **width** attribute to "300" and render the web page. The video's display area is now 300 pixels wide. The **height** attribute can be used to set the height of the video's display area in pixels.
3. Add the **autoplay** attribute to the **<video>** tag and render the web page. Note how the video plays automatically.
4. Add the **muted** attribute to the **<video>** tag and render the web page. Note how the video is initially muted.

New York City Subway Arriving: CC0 License



Different web browsers support different video formats, mainly because of patent issues. Multiple **<source>** tags can be used to supply alternate file formats. MP4, WebM, and Ogg are ideal video formats for playing video on the web.

Table 3.5.2: Common video formats.

Format	File extension	Description
Flash	.swf .flv	Developed by Macromedia. Requires a plug-in to play.
MPEG-4 or MP4	.m4a .mp4	Developed by the Moving Pictures Expert Group. m4a is audio only, but mp4 may contain video. Supported by HTML5.

Ogg	.ogg	Developed by Xiph.Org Foundation. Supports audio and video.
QuickTime	.mov	Developed by Apple. Plays in Macs, but plug-in support for Windows has been dropped.
WebM	.webm	Open media format developed by Mozilla, Opera, Adobe, and Google for the web.
WMV (Windows Media Video)	.wmv	Developed by Microsoft for storing video on Windows.

**PARTICIPATION ACTIVITY**

3.5.5: Video element.

- 1) The size of the video displayed by a video element is configurable.
- True
- False
- 2) Multiple `<source>` tags can be used to specify different video formats.
- True
- False
- 3) Playing video uses significantly more network bandwidth than playing just audio.
- True
- False

## Controlling media playback with JavaScript

The audio and video elements can be controlled with JavaScript. Ex: JavaScript can start and stop audio, change the volume, and jump to a specific location in a video. See the [HTMLMediaElement link](#) in Exploring further below for a list of JavaScript properties and methods to control audio and video.

## Showing YouTube videos

Developers may host a video on YouTube to display in a web page. Including a YouTube video in a web page requires the `<iframe>` element. The `<iframe>` element allows a web page to be embedded in a rectangular area of the current web page. The `<iframe>` element uses the `src` attribute to specify the URL of the web page to display and the `width` and `height` attributes to define the width and height in pixels of the rectangular iframe.

The YouTube Help pages give instructions for [uploading a video](#) to YouTube and [embedding a video](#) into a web page.

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019

PARTICIPATION  
ACTIVITY

3.5.6: Embedding a YouTube video.



The web page below uses the `<iframe>` tag to embed a YouTube video in the web page. The URL of the YouTube web page ends with "7g7kP\_Trp0g?rel=0". `7g7kP_Trp0g` is the ID assigned to the video when the video was first uploaded to YouTube, and `?rel=0` disables showing related videos when the video finishes.

1. Click the video to play the video and the pause button to pause the video.
2. Add `&autoplay=1` to the end of the YouTube URL and render the web page. The video should play automatically.
3. Change the video ID in the URL to `g4hvUvBmoaA` and render the web page. A different video should display in the iframe.



PARTICIPATION  
ACTIVITY

3.5.7: YouTube videos and `<iframe>`.



- 1) What `<iframe>` attribute indicates the URL of the web page to display in the iframe?

- link
- src
- href

- 2) What URL plays a YouTube video automatically?

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019



- <https://www.youtube.com/embed/g4hvUvBmoaA>
- <https://www.youtube.com/embed/g4hvUvBmoaA?autoplay>
- <https://www.youtube.com/embed/g4hvUvBmoaA?autoplay=1>

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019

Exploring further:

- [HTML5 Video from W3Schools](#)
- [HTML5 Audio from W3Schools](#)
- [HTML5 Plug-ins from W3Schools](#)
- [Media formats supported by the HTML audio and video elements from MDN](#)
- [HTMLMediaElement from MDN](#)

How was this section?

[Provide feedback](#)

## 3.6 <script> and <style>

The **<script>** tag allows a web page to include interactive content, which the browser assumes to be JavaScript unless indicated otherwise. The optional type attribute is used to indicate the content type when the content is not JavaScript. The src attribute provides the URL of an external document containing the interactive content. If a <script> tag does not have the src attribute, then the interactive content is contained directly within the tag. The HTML below shows two ways the <script> tag can be used.

### Example 3.6.1: Two ways of using <script> tags.

```
<script src="https://example.com/my_interactive_content.js"></script>
<script type="text/javascript">
    alert('Hello, World!');
</script>
```

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019

A common error is to forget the closing </script> tag when using the src attribute. Even when the interactive content is located in a separate external document, the closing </script> tag must be included.

Good practice is to use the src attribute to separate content and functionality and promote modularity. An external JavaScript file can be edited separately from an HTML file, which allows a web page's interactive content to be updated even if the content is unchanged. A separate JavaScript file can also be reused on many web pages to provide the same functionality for different pieces of content. Additionally, when the JavaScript file is separate, a browser that doesn't understand JavaScript, such as a screenreader for blind users, can avoid downloading the JavaScript file that will not be used.

©zyBooks 02/17/19 22:08 458368

The **<style>** tag allows the web page to introduce presentational directives, usually CSS. A **<style>** tag is placed in an HTML document prior to the **<body>** tag, because the style section is designed to describe the presentation of the entire document. Although only needed for non-CSS content and rarely used, the **<style>** tag has an optional type attribute that describes the content inside the tag.

SANDIEGOCOMP494GappySpring2019

### Example 3.6.2: Using the **<style>** tag.

```
<html>
  <style>
    p {
      margin: 1em;
    }
  </style>
  <body>
    <p>The White House is the official residence of the
       President of the United States.
    </p>
  </body>
</html>
```

### Note

*Unlike all other HTML tags, the contents within the **<script>** and **<style>** tags are not displayed by the browser. The **<script>** and **<style>** tags' purpose is to provide interactive functionality and presentational styling.*

#### PARTICIPATION ACTIVITY

#### 3.6.1: **<script>** and **<style>** tags.

text/css

text/javascript

<script>

<style>

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019

Tag that surrounds interactive content  
in an HTML document.

Tag that surrounds presentational content in an HTML document.

Default type for the <script> tag.

Default type for the <style> tag.

©zyBooks 02/17/19 22:08 458368

Reset

jingya xun

SANDIEGOCOMP494GappySpring2019

How was this section?



[Provide feedback](#)

## 3.7 HTML developer guidelines

The following developer guidelines are helpful best practices for web development. Although the guidelines and suggestions are not mandatory, good practice is to consistently follow the guidelines, which results in maintainable web pages and helps avoid errors. Cleanly-organized HTML is also easier to read and understand by other developers.

### Use closing tags

For tags that have a closing tag, always use the closing tag to mark the end of the tag content. Ex: HTML5 does not require closing </p> tags because the web browser can infer the ending of a paragraph, but always including the </p> closing tag to end the paragraph explicitly is safer. A common error is to expect all web browsers to add a missing closing tag at the same location in the HTML.

### Avoid self-closing tags

HTML5 does not require or promote using self-closing tags for empty elements. Ex: Use <br> not <br />, and use <img ... > not <img ... />.

PARTICIPATION  
ACTIVITY

3.7.1: Use closing tags.

©zyBooks 02/17/19 22:08 458368  
jingya xun

SANDIEGOCOMP494GappySpring2019



Does the HTML follow the developer guidelines to always use closing tags and avoid self-closing tags?

1)



```
<ol>
  <li>Avatar
  <li>Titanic
  <li>Jurassic World
</ol>
```

- Yes  
 No

2) `<p>We have updated our company logo to  
  
from the older logo of  
</p>`

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019



- Yes  
 No

3) `<div class="footer">  
<p>This site is copyright 2016.</p>`



- Yes  
 No

4) `<address>  
John Smith <br />  
123 Main St. <br />  
Anytown, USA  
</address>`



- Yes  
 No

## Use quotes for attributes values

Always use quotes around attributes values. Ex: Use `value="Start"` instead of `value=Start`. While HTML5 does not require quotes around an attribute value that does not contain a space, a common error is forgetting the quotes for attribute values that contain spaces. Ex: `value=Start animation` is illegal and results in an error.

## Use double quotes

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019

Although HTML5 accepts either double or single quotes, good practice is to use double quotes, which in general results in more readable HTML. Ex: Use `value="Start"` instead of `value='Start'`. Inconsistent use of quotes can also lead to problems when using some web development frameworks, which assume the developer uses double quotes.

If a double quote (") needs to be placed within a string, the internal double quote must be escaped. A special character such as the double quote can be **escaped** within a string by placing

the backslash character (\) before the special character, causing that character to lose any special meaning. Ex: "Dwight D. \"Ike\" Eisenhower".

**PARTICIPATION ACTIVITY**

3.7.2: Identify correct use of quotes around attribute values.



Which quote guideline is violated ?

1) `<p id="preamble">We the people ...</p>`

- Use quotes around attribute values
- Use double quotes
- No guideline violations

2) `<input type="checkbox" checked>`



- Use quotes around attribute values
- Use double quotes
- No guideline violations

3) `<input type="checkbox" checked>`



- Use quotes around attribute values
- Use double quotes
- No guideline violations

4) `<input type='text' name='zip'>`



- Use quotes around attribute values
- Use double quotes
- No guideline violations

5) `<a href="http://example.com" rel='nofollow'>`

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019



- Use quotes around attribute values
- Use double quotes
- No guideline violations

6)



```
<p class="assertion">He says he is  
'innocent'.</p>
```

- Use quotes around attribute values
- Use double quotes
- No guideline violations

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019

## Use boolean attributes concisely

A **boolean attribute** is an attribute that is true when present false when absent. Ex: The checked attribute for a checkbox widget is a boolean attribute. If the checked attribute is set, the checkbox will start in the selected state; otherwise, the checkbox will start in the unselected state. Older versions of HTML required all attributes to have values, so developers would use `checked="checked"` to indicate the checked attribute was true. browsers continue to support the older syntax, but good practice is to only use the attribute name for a boolean attribute without specifying a value. Ex:

`<input type="checkbox" name="foodPreference" value="vegetarian" checked>`  
is the preferred way to use the checked boolean attribute.

PARTICIPATION  
ACTIVITY

3.7.3: Using without a value.



Does the HTML follow the developer guidelines to use boolean attributes concisely?

1) `<select>`  
`<option selected>Pick an option`  
`below</option>`  
`<option>Lions</option>`  
`<option>Tigers</option>`  
`<option>Bears</option>`  
`<option>Oh, my!</option>`  
`</select>`



- Yes
- No

2) `<div hidden="hidden">`



- Yes
- No

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019

3) `<a href="#"nofollow>Don't spam me!`  
`</a>`



- Yes
- No

## Use lowercase for all tags and attributes

Good practice is to use lowercase for all tags and attributes. Ex: Use `<p>` and `value="Start"` instead of `<P>` and `VALUE="Start"`.

**PARTICIPATION**  
**ACTIVITY**

3.7.4: Use lowercase.

©zyBooks 02/17/19 22:08 458368

jingya xun



Does the HTML follow the developer guidelines to use lowercase for tags and attributes?

 1) `<H1>Table of Contents</H1>`

- Yes
- No


 2) `<img Src="https://apple.com/apple.jpg">`

- Yes
- No



## Start block-level elements on new lines

Each block-level element should start on a new line. Ex: A `<p>`, `<table>`, or `<ol>` element should always begin on a new line. To highlight the nature of block-level elements, inline elements should only start on a new line for readability purposes. Ex: A `<a>`, `<span>`, or `<q>` element may start anywhere on a line.

**PARTICIPATION**  
**ACTIVITY**

3.7.5: Start block-level elements on a new line.



Does the HTML follow the developer guidelines to start block-level elements on new lines?

 1) `<h1>Olympics</h1><p>The IOC organises the modern summer and winter Olympic Games and Youth Olympic Games held every four years.</p>`

- Yes
- No

©zyBooks 02/17/19 22:08 458368

jingya xun



SANDIEGOCOMP494GappySpring2019

 2) `<h1>Olympics</h1><p>The IOC organises the modern Olympic Games and Youth Olympic Games, held in summer and winter, every four years. The first Summer Olympics organised by the IOC was held in Athens, Greece, in 1896.</p>`


Yes No

- 3) `<label for="city">City:</label>  
<input type="text" name="city"  
id="name">`

 Yes No

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019

## Indent nested elements consistently

Nested elements should be indented at least two spaces. Indentation should be consistent throughout a file.

Table 3.7.1: Good and bad examples of consistent indentation.

| Guideline                 | Good example   | Bad example  |
|---------------------------|--|--|
| Indenting nested elements | <pre>&lt;ol&gt;   &lt;li&gt;Item 1&lt;/li&gt;   &lt;li&gt;Item 2&lt;/li&gt; &lt;/ol&gt;</pre>  | <pre>&lt;ol&gt; &lt;li&gt;Item 1&lt;/li&gt; &lt;li&gt;Item 2&lt;/li&gt; &lt;/ol&gt;</pre>  |
| Consistent indenting      | <pre>&lt;html&gt;   &lt;title&gt;My Title&lt;/title&gt;   &lt;p&gt;First paragraph&lt;/p&gt;   &lt;p&gt;Second paragraph&lt;/p&gt; &lt;/html&gt;</pre> | <pre>&lt;html&gt;   &lt;title&gt;My Title&lt;/title&gt;     &lt;p&gt;First paragraph&lt;/p&gt;     &lt;p&gt;Second paragraph&lt;/p&gt; &lt;/html&gt;</pre> |

### PARTICIPATION ACTIVITY

3.7.6: Indent nested elements.

Does the HTML follow the developer guidelines to consistently indent nested elements?

- 1) The all-time top grossing movies domestically<ol>
 <li>Avatar</li>
 <li>Titanic</li>
 <li>Jurassic World</li>
</ol>

 Yes No

- 2)

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019

```
<tr>
  <td>Hematocrit</td>
  <td>34.9–44.5%</td>
  <td>38.8–50.0%</td>
</tr>
```

- Yes  
 No

3) `<form>
 <label for="name">Name:</label>
 <input type="text" id="name"
 name="name">
 <label for="address">Address:
 </label>
 <input type="text" id="address"
 name="address">
 <label for="city">City:</label>
 <input type="text" id="city"
 name="city">
</form>`

- Yes  
 No

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019



## Provide labels and placeholders

Always provide labels and/or placeholders for widgets. Ex: Use

`<input type="text" name="firstName" placeholder="First name">`.

PARTICIPATION  
ACTIVITY

3.7.7: Providing labels and placeholders.



Does the HTML follow the developer guidelines to provide labels and placeholders?

1) `<input type="text"
 name="favoriteSport" id="sport">`

- Yes  
 No



2) May we add you to our mailing list?

```
<br>
<label for="optInYes">Yes</label>
<input placeholder="Add me"
type="radio" name="optIn"
id="optInYes" value="true">
<label for="optInNo">No</label>
<input placeholder="Do not add me"
type="radio" name="optIn"
id="optInNo" value="false">
```

- Yes  
 No

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019



## Separate content from presentation and functionality

Good practice is to not use embedded or inline CSS and JavaScript. Ex: Instead of using `<p style="color:red;">No!</p>`, use `<p class="attention">No!</p>` and apply a rule in the CSS file. Best practice is to use separate files for CSS and JavaScript. Ex: Instead of using the `<style>` tags for specifying CSS rules within the HTML document, specify the CSS rules in a separate CSS file, such as theme.css, and import the file using `<link rel="stylesheet" href="theme.css">`.

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019

**PARTICIPATION  
ACTIVITY**

3.7.8: Separate content from presentation and functionality.



Does the HTML follow the developer guidelines to separate content from presentation and functionality?

1) `<html>
 <link rel="stylesheet"
 href="main_style.css">
 <body>
 <p>A page with style!</p>
 </body>
</html>`

- Yes
- No



2) `<button onclick="return
process_data()">Process Data</button>`

- Yes
- No



3) `<html>
 <style>p { color: blue }</style>
 <body>
 <p>Once in a blue moon!</p>
 </body>
</html>`

- Yes
- No



©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019

## Use CSS for layout

CSS is designed for layout; tables are designed for holding tabular data. Good practice is to avoid using tables to manage page layout and to avoid using CSS to manage tabular data. Using tables to manage layout obscures the content and meaning of the web page and can cause problems when trying to control layout correctly using CSS. Using CSS to build tables requires more complex JavaScript to manage data tables in more sophisticated web applications.

## Example 3.7.1: Example HTML5 web page.

The following HTML5 code illustrates the style guidelines above.

```
<!DOCTYPE html>
<html>
  <title>Example usage of HTML style rules</title>
  <meta charset="UTF-8">
  <link rel="stylesheet" href="layout.css">
  <script src="functions.js"></script>
  <body>
    <p>According to the <a href="https://usatoday.com">USA Today</a>,
       the three busiest travel days are:</p>
    <ol>
      <li>Summer weekends</li>
      <li>Thanksgiving</li>
      <li>Christmas</li>
    </ol>
    <p>According to the <a href="https://www.mayoclinic.org">Mayo Clinic</a>,
       the following values are typical complete blood count results:</p>
    <table>
      <caption>Normal complete blood count results for adults.</caption>
      <tr>
        <th>Count</th>
        <th>Female</th>
        <th>Male</th>
      </tr>
      <tr>
        <td>Red blood cell</td>
        <td>3.90–5.03 trillion cells/L</td>
        <td>4.32–5.72 trillion cells/L</td>
      </tr>
      <tr>
        <td>Hematocrit</td>
        <td>34.9–44.5%</td>
        <td>38.8–50.0%</td>
      </tr>
    </table>
    <form action="ignored" method="post">
      <label for="surname">Surname:</label>
      <input type="text" name="surname" id="surname" placeholder="Enter your last name.">
      <label for="promise">I promise to follow the HTML style rules.</label>
      <input type="checkbox" name="promise" id="promise" checked>
      <input type="submit">
    </form>
  </body>
</html>
```

©zyBooks 02/17/19 22:08 458368  
jingya xun  
SANDIEGOCOMP494GappySpring2019

## Validate HTML

Always validate HTML using an HTML validator and revise the HTML to eliminate any errors or warnings. One of the big challenges in web development is to make sure that the web pages are processed correctly by as many browsers as possible. Strict adherence to the HTML5 standard generally reduces browser errors. The W3C official validator at <https://validator.w3.org> is free, works well, and allows validation by providing the URL, by uploading an HTML file, or by directly entering HTML.





Does the following HTML pass the [HTML validator](#) with no errors or warnings? Copy the HTML, click on the validator link, paste, and click the check button on the validator.

1) 

```
<!DOCTYPE html>
<html lang="en">
<title>Test page</title>
<p>Does this page pass the validator
test?</p>
</html>
```



©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019

- Yes  
 No

2) 

```
<!DOCTYPE html>
<html lang="en">
<h1>Test page</h1>
<p>Does this page pass the validator
test?</p>
</html>
```



- Yes  
 No

**PARTICIPATION**  
**ACTIVITY**

3.7.10: Consider all developer guidelines.



©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019

```

( 0) <!DOCTYPE html>
( 1) <html>
( 2) <title>Movies based on books</title>
( 3)   <body>
( 4)     <p>The following movies were based on books that
( 5)       had been written previously.
( 6)     <ol>
( 7)       <li>The Wizard of Oz</li>
( 8)       <li>Harry Potter and the Sorcerer's Stone</li>
( 9)       <li>Willy Wonka and the Chocolate Factory</li>
(10)     </ol>                                         ©zyBooks 02/17/19 22:08 458368
(11)     <p>Indicate which medium you think is better.</p>      jingya xun
(12)     <form action="http://example.org/bookVSmovie.html"        SANDIEGOCOMP494GappySpring2019
(13)       method="post" ><p>The Wizard of Oz</p>
(14)       <label for="wizOzBook">Book:</label>
(15)       <input type="radio" id="wizOzBook" name="wizOz"
(16)         value="wizBook" />
(17)       <label for="wizOzMovie">Movie:</label>
(18)       <input type="radio" id="wizOzMovie" name="wizOz"
(19)         value="wizMovie" />
(20)     <p>Harry Potter and the Sorcerer's Stone:</p>
(21)     <label for="PotterBook">Book:</label>
(22)     <input type="radio" id="PotterBook" name="potter"
(23)       value="PotterBook" />
(24)     <label for="PotterMovie">Movie:</label>
(25)     <input type="radio" id="PotterMovie" name="potter"
(26)       value="PotterMovie" />
(27)     <p>Willy Wonka and the Chocolate Factory</p>
(28)     <label for="WonkaBook">Book:</label>
(29)     <input type="radio" id="WonkaBook" name="wonka"
(30)       value="WonkaBook" checked="checked" />
(31)     <label for="WonkaMovie">Movie:</label>
(32)     <input type="radio" id="WonkaMovie" name="wonka"
(33)       value="WonkaMovie" ><input type='submit'>
(34)   </form>
(35) </body>
(36) </html>

```

1) Which guideline is violated on line 2?



- Use closing tags
- Start block-level elements on new lines
- Indent nested elements consistently
- Use CSS for layout

2) Which line does not use double quotes correctly?



- 8
- 20
- 33

3) Which guideline is violated on line 30?



- Use closing tags
- Use boolean attributes concisely

- Start block-level elements on new lines
- Indent nested elements consistently

4) Which guideline is violated on line 16?



- Avoid self-closing tags
- Use double quotes
- Indent nested elements consistently
- Provide labels and placeholders

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019

5) Which line did not start a block-level element on a new line?



- 6
- 12
- 13
- 33

6) Which guideline is violated in line 4?



- Use closing tags
- Start a block-level element on a new line
- Use lowercase for all tags and attributes

7) Which guideline is violated in lines 7-9?



- Start block-level elements on new lines
- Indent nested elements consistently
- Separate content from presentation and functionality
- Validate HTML

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019

Exploring further:

- Google's style guide for HTML and CSS.

- W3Schools' style guide for HTML, CSS, and JavaScript.

How was this section?



[Provide feedback](#)

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019

©zyBooks 02/17/19 22:08 458368

jingya xun

SANDIEGOCOMP494GappySpring2019