16 WORKING WITH WEB ANIMATION AND VIDEO BONUS

Lesson overview

In this lesson, you'll learn how to incorporate web-compatible animation and video components into your webpage and do the following:

- Insert web-compatible animation
- Insert web-compatible video

This lesson will take about 60 minutes to complete. Please log in to your account on peachpit.com to download the project files for this lesson, as described in the "Getting Started" section at the beginning of this book. Follow the instructions under "Accessing the Lesson Files and Web Edition." Define a site based on the lesson16bonus folder.

Understanding web animation and video

The web can provide a variety of experiences to the average user. One second you are downloading and reading a best-selling novel. Next, you're listening to your favorite radio station or performing artist. Then, you're watching live television coverage or a feature-length movie. Before Adobe Flash, animation and video were hard to incorporate into webpages. That's because HTML was invented at a time when even static images were difficult to use on the Internet; video was a dream far off in the future.

For a time, Adobe Flash brought order to this chaos. It provided a single platform for creating both animation and video. However, with the invention and rise in popularity of smartphones and tablet devices over the last decade, Flash has fallen on hard times. For most manufacturers, the power and capability of Flash were too difficult to support on these devices and it was abandoned. Flash is not dead. It's still unmatched for its multimedia power and functionality. But today, all bets are off when it comes to animation and video.

The techniques for creating web-based media are being reinvented. As you may have guessed, this trend away from Flash is ringing in a new era of chaos on the web-media front. Half a dozen or more codecs are competing to become the "be-all, end-all" format for video distribution and playback for the web. The only ray of sunshine in this morass is that HTML5 was developed with built-in support for both animation and video.

Great strides have already been made to replace much of the capability of Flash-based animation using native HTML5 and CSS3 functionality. The status of video is not as clear. So far, a single standard has not yet emerged, which means that to support all the popular desktop and mobile browsers, you'll have to produce several different video files. In this lesson, you'll learn how to incorporate web animation and video into your site.

Previewing the completed file

To see what you'll work on in this lesson, preview the completed page in a browser. The finished page is based on the travel page you created in Lesson 10, "Adding Interactivity."

1 Launch your favorite web browser.

2 Open **travel-finished.html** from the finished folder in lesson16.
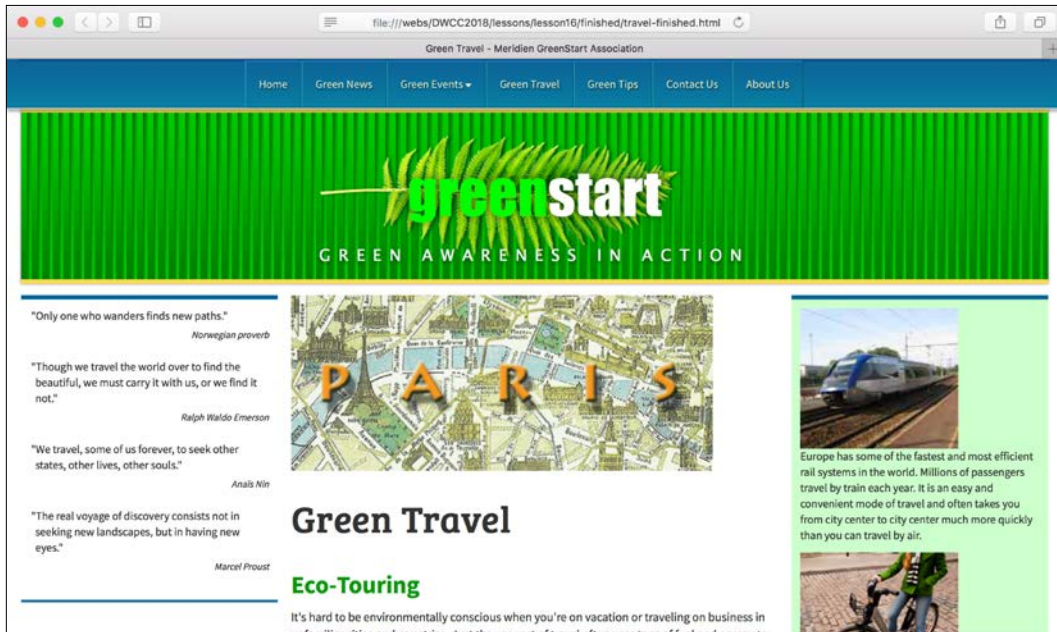
The page includes two media elements: the banner animation at the top of the `main_content` region and the video inserted below. Depending on the browser used to view the page, the video playback may be generated from one of three different formats: MP4, WebM, or Ogg.

Note that the banner ad plays when the page loads and loops after it calls you to sign up for the Eco-Tour.

3 To view the video, click the Play button. Move the cursor over the video to display the controls, and click the Play button.

Different browsers support different types of video. Depending on the video format your browser supports, you may notice that the controls fade if you move the cursor away from the video but that they return once you position the cursor over the video again.

4   When you're finished previewing the media, close your browser.

Video and animation provide powerful venues for rich web content, and Dreamweaver makes it a simple matter to incorporate this type of content in your webpages.

Adobe Animate CC

The animation used in this lesson was built in Animate CC, Adobe's replacement for Flash Professional. Animate works similarly to Flash but exports web animation and interactive content natively using HTML5, CSS3, and JavaScript, among other formats. Animate is available to all Creative Cloud subscribers.

Check out www.adobe.com/products/animate.html for more information about Animate CC.

Adding web animation to a page

Dreamweaver has a built-in and simplified workflow for inserting Animate compositions, making the process a point-and-click operation. Dreamweaver takes advantage of a feature in Animate designed to assist in deploying compositions to other programs and workflows, such as Adobe InDesign, Adobe Dreamweaver, and Apple's iBooks Author. The Publish Settings command in Animate enables you to export your compositions into a single file or folder. By defining your Publish settings appropriately, you can create a complete set of files that are compatible with these applications.

For the purposes of this exercise, I created an animated banner ad 480 pixels by 200 pixels in Animate CC and published it to an OAM file, which is an archive file format that contains all the constituent elements needed to support the animation in Dreamweaver and other Adobe applications. In this exercise, you will learn how to insert the animation into a layout and make it responsive.

1   Open **travel.html** from the lesson16 site root folder in Live view. Ensure that the program is maximized to fit the entire display and that the document window is at least 1100 pixels in width.

    The banner will be inserted above the main heading.

2   Click the heading *Green Travel*.
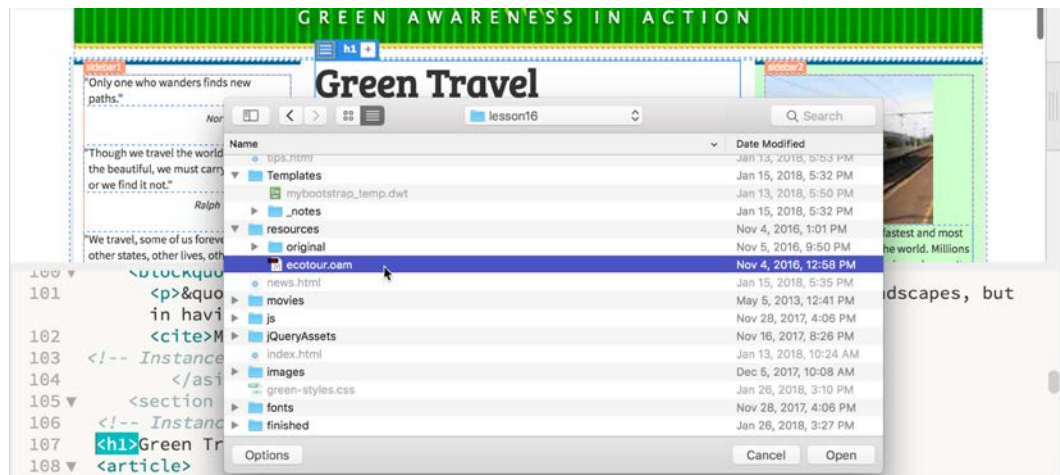
    The Element HUD appears focused on the h1 element.

3 Choose Insert > HTML > Animated Composition. You can also use the Animated Composition option in the Insert panel's HTML category.
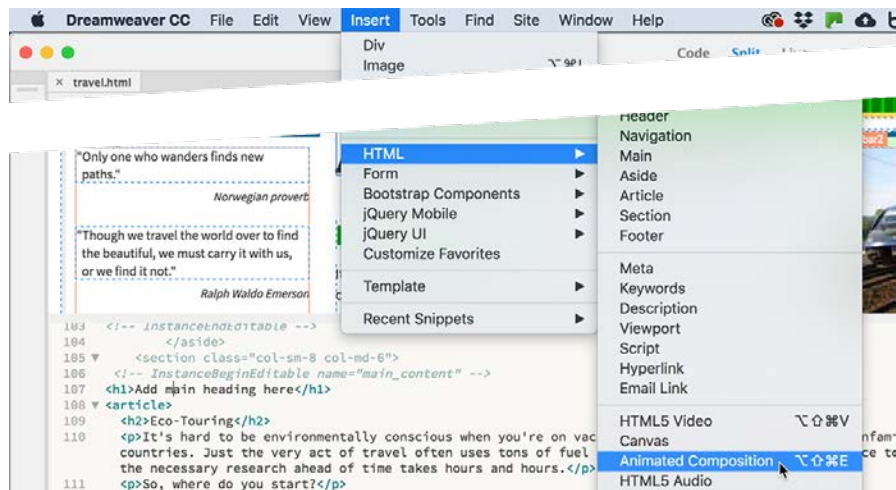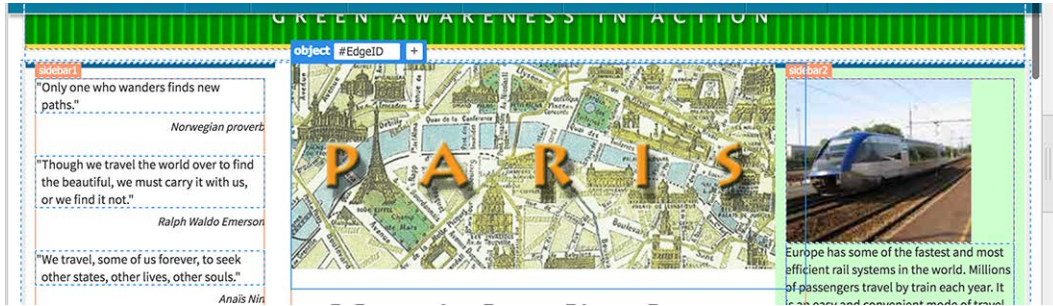

4 Click Before in the Position Assist dialog.

A file dialog opens.

5 Navigate to the lesson16/resources folder.
Select **ecotour.oam**.

6 Click OK/Open to insert the composition.

The banner should appear and begin to play. It is designed to loop over and over. Although the animation is designed to fit in the center column, there seems to be a small issue with the width of the animation if your document window is less than 1200 pixels in width. It seems to be hanging outside the right edge of the center section.

7 Save the file and switch to Code view.
Examine the HTML code.

```
105 ▼    <section class="col-sm-8 col-md-6">
106   <!-- InstanceBeginEditable name="main_content" -->
107 ▼ <object id="EdgeID" type="text/html" width="500" height="220" data-dw-
      widget="Edge" data="animation_assets/ecotour/Assets/ecotour.html">
108   </object>
109   <h1>Green Travel</h1>
```

The parent structure for the animation should still be selected. This is not the animation itself; instead, Dreamweaver has inserted an `<object>` element that points to the actual animation stored within the site.

The first issue you have to deal with is that Dreamweaver for some reason has increased the width of the animation. The actual banner is supposed to be 480 pixels by 200 pixels, but the `<object>` element shows HTML attributes set for 500 by 220.

8 In the Code view window, edit the width to say **480**.
Edit the height to say **200**.

```
105 ▼    <section class="col-sm-8 col-md-6">
106   <!-- InstanceBeginEditable name="main_content" -->
107   <object id="EdgeID" type="text/html" width="480" height="200" data-dw-
      widget="Edge" data="animation_assets/ecotour/Assets/ecotour.html">
108   </object>
109   <h1>Green Travel</h1>
```

**9** Switch to Live view. Observe the animation.

The animation now fits within the center column. The banner animation plays automatically in Live view once the code is processed.

It may seem like a miracle that this amazing banner animation was created by one simple file, but what you cannot see is that Dreamweaver just created in the site root directory a new folder that is populated by a half dozen files.

10 Open the Files panel and examine the list of folders in the site root.

● **Note:** You may need to click the Refresh icon at the bottom of the panel to see all the files in the site folder.
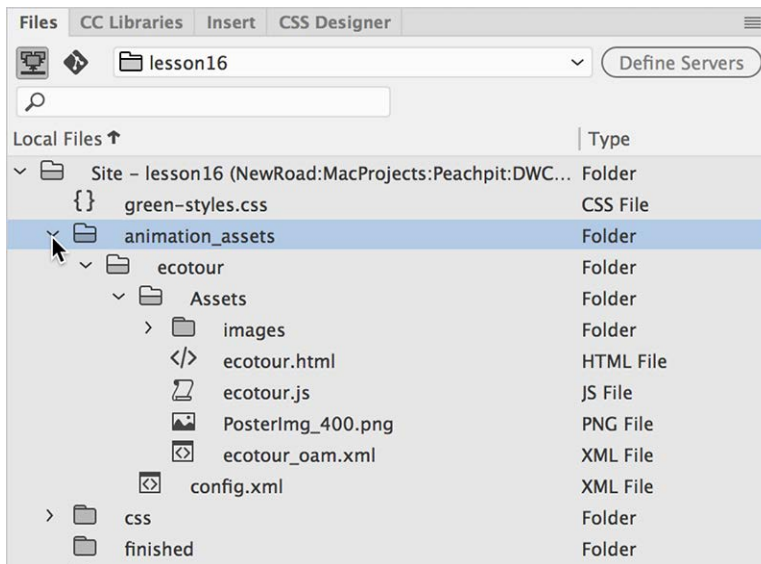
The folder named animation_assets now appears in the root directory. The folder was generated automatically and contains all the files needed to support the composition. The entire folder must be uploaded to the web host when **travel.html** is posted.

◆ **Warning:** Dreamweaver will not automatically upload all the support files needed for the Animate composition when you select dependent files. Be sure the entire contents of the **animation_assets** folder is uploaded when publishing the site to the web.

11 Save all files.

Congratulations! You've successfully incorporated an HTML5- and CSS3-based animation into your page. But you're not done yet. Whenever you insert a component in your layout, especially animations and video, you need to check to see whether they conform properly to your responsive design.

Making the animation responsive

The heading on this exercise is a bit misleading. The animation you inserted earlier is already responsive. But that doesn't mean it will respond properly in the current layout. Let's check it out.

1 Open **travel.html** from the lesson16 site root folder in Live view. Ensure that the program is maximized to fit the entire display and that the document window is at least 1100 pixels in width.

The layout should be displayed in three columns, with the animation playing at the top of the middle column.

2 Drag the Scrubber to the left until the document window is narrower than 480 pixels.


The animation appears normally in all widths until you get below 480 pixels. At smaller widths the animation protrudes from the main content section on the right side and does not scale smaller.

Bootstrap provides a method for supporting responsive animation and video, but at the time of this writing, Dreamweaver only has built-in support for embedding video. You'll have to create and modify the code for the animation yourself. The first step is to wrap the animation in a Bootstrap-responsive container.

3 Drag the scrubber back to the right side.

4 Switch to Split view. In Code view, locate the `<object>` element containing the animation (around line 107).

5 Insert the cursor anywhere in the `<object>` element. Select the `object#EdgeID` tag selector.
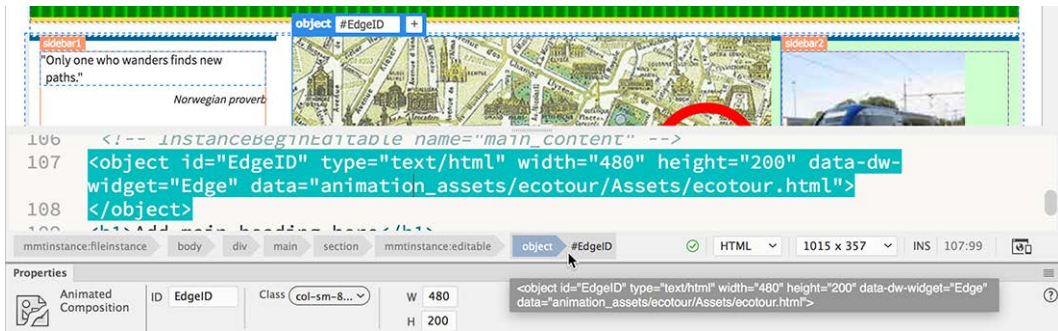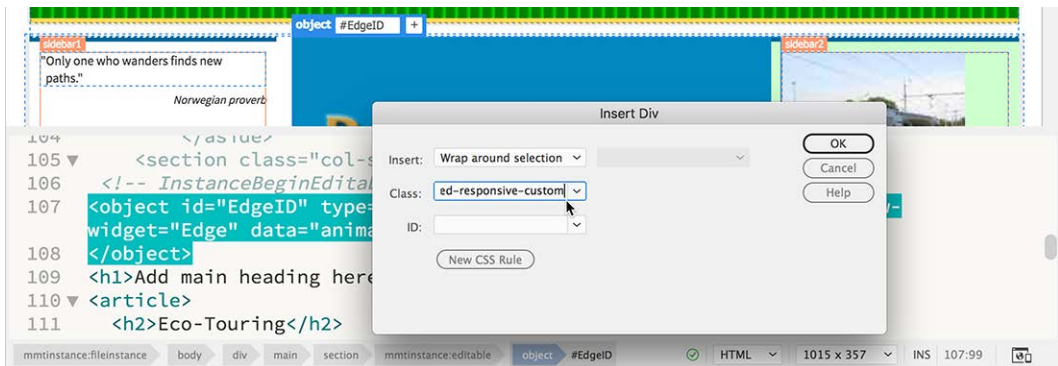

The entire element is highlighted in Code view.

6 Select Insert > Div.

The Insert Div dialog appears. You'll need to enter two class names.

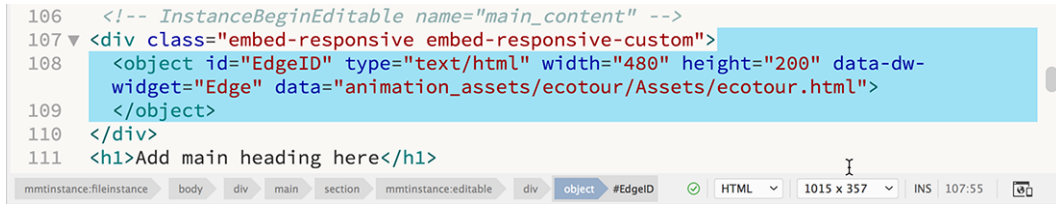● **Note:** Make sure the Insert menu shows the option Wrap Around Selection.

7 In the Class field, enter the following classes:
`embed-responsive embed-responsive-custom`

As you type, the first class may appear in the field. Feel free to select it using the mouse or keyboard. The second class name will have to be entered manually. Be sure to insert a space between the two names.
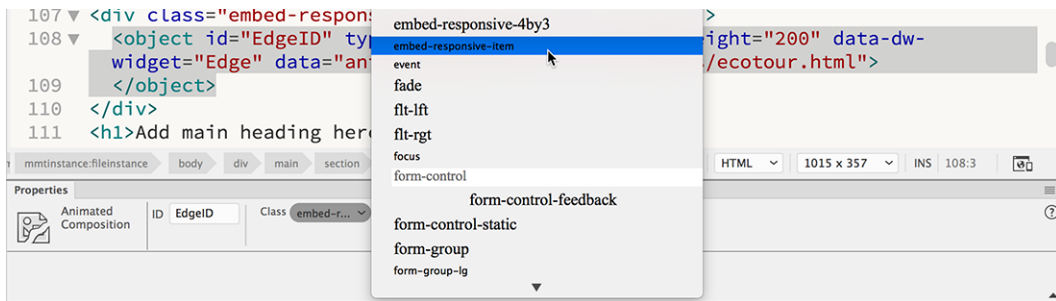
8 Click OK.

The responsive `<div>` now wraps the `<object>` element. You'll need to add a responsive class to it as well.

9 Select the `object#EdgeID` tag selector again.
In the Property inspector, click the Class field and select this class:
embed-responsive-item

The animation disappears from the layout. The class you assigned to the `<object>` element caused it to collapse visually. It's still there, as you can see from the code, but you'll have to add a new rule to make it appear properly and work within the new structure.
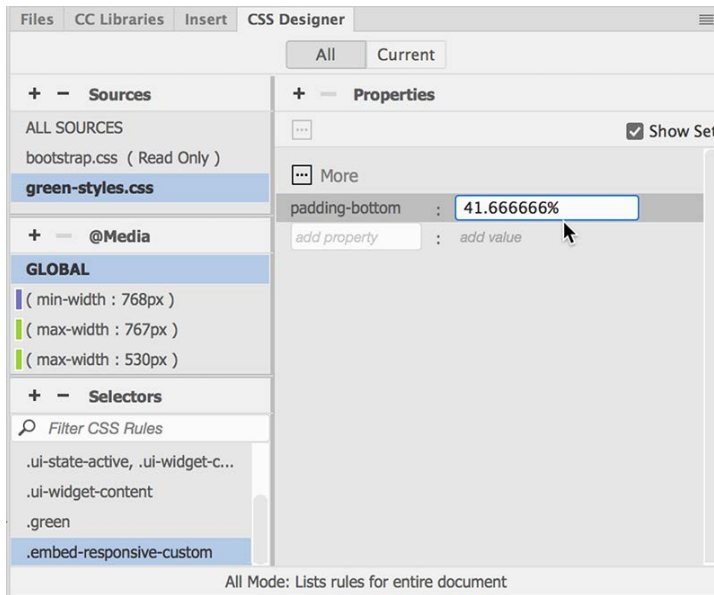
● **Note:** Remember that the All button must be enabled before you try to select the CSS source.

10 Choose **green-styles.css** > GLOBAL in the CSS Designer.

**11** Create the following selector:
.embed-responsive-custom

12 Add the following property:

`padding-bottom: 41.666666%`

As soon as the property is created, the animation reappears in the layout.

The Bootstrap framework currently supports two basic embed ratios of 4:3 and 16:9 that match the most popular video formats. If the animation matched one of these ratios, you would select an existing Bootstrap class that supports it. But it doesn't, so we had to use a custom padding property derived from a factor of its height and width. Check out http://tinyurl.com/fluid-width-animation to learn more about working with fluid width video.

13 Drag the Scrubber to the left until the document window is narrower than 400 pixels.
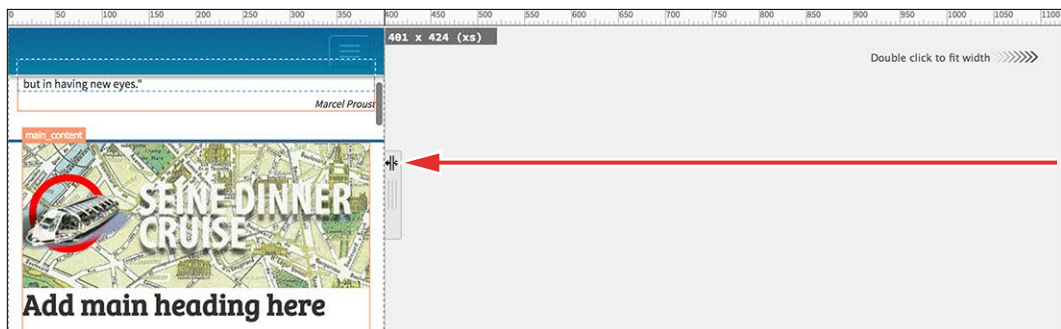
The animation scales down to match the width of the column.

14 Drag the Scubber all the way to the right, or double-click in the gray area to the right of the Scrubber. Save all files.

You've now completed an HTML5-compatible animation. Now, let's see how easy it is to work with video.

Adding web video to a page

Implementing HTML5-compatible video in your site is a bit more involved than it was when you had to insert only a single Flash-based file. Unfortunately, no single video format is supported by all browsers in use today. To make sure your video content plays everywhere, you'll have to supply several different formats. Dreamweaver provides a built-in technique to add multiple video files so you won't have to do all the coding yourself. In this exercise, you will learn how to insert HTML5-compatible video on a page in your site.

1 If necessary, open **travel.html** in Live view. Ensure that the program is maximized to fit the entire display and that the document window is at least 1100 pixels in width.

You will insert the video in the `main_content` section of the page.

2 Click the paragraph *Click here to sign up*.

The HUD appears focused on the p tag. As mentioned earlier, the Insert panel provides a Bootstrap option to insert responsive video.

3 Choose Window > Insert, if necessary.
Select the **Bootstrap Components** category.
Click the **Responsive Video Embed option**.
The Position Assist HUD appears.
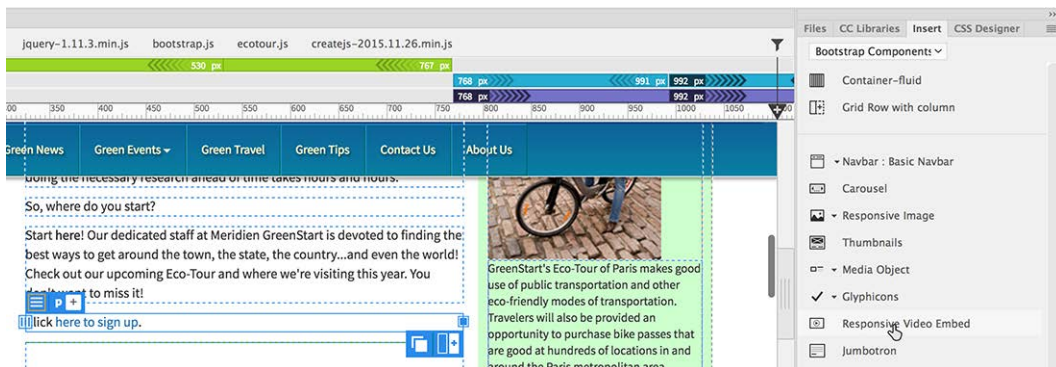
4 Click Before.
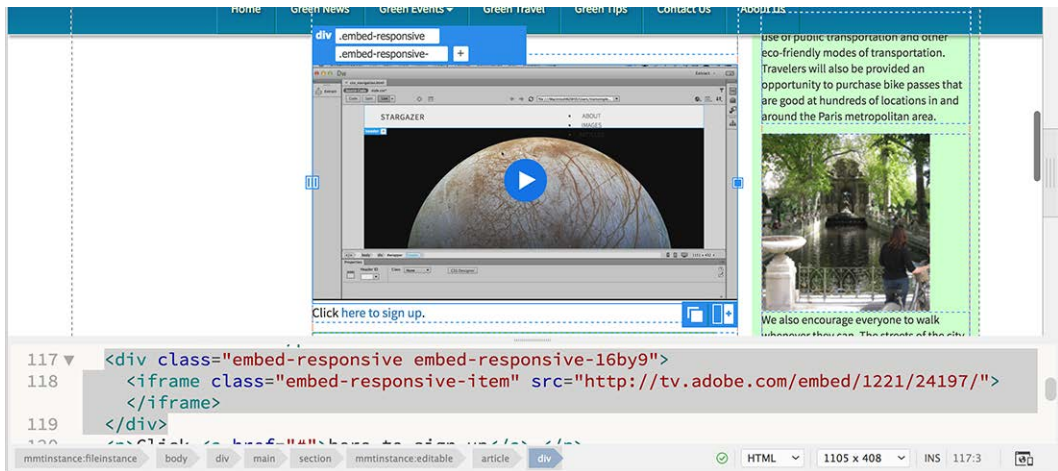
A Bootstrap-compatible video placeholder appears on the page. It contains a sample video hosted on the web. If you have a live Internet connection, you can actually play the sample video.

This placeholder uses an `<iframe>` element to host the video. An `<iframe>` element is usually employed to host web content from a third-party website, such as YouTube or Vimeo. Since you are going to host your own video, there's no need for the `<iframe>`. You're going to replace the `<iframe>` with an HTML5-compatible `<video>` element.

5 If necessary, switch to Split view. In the Code view window, insert the cursor in the `<iframe>` element.

6 Select the `iframe.embed-responsive-item` tag selector. Note the class name.

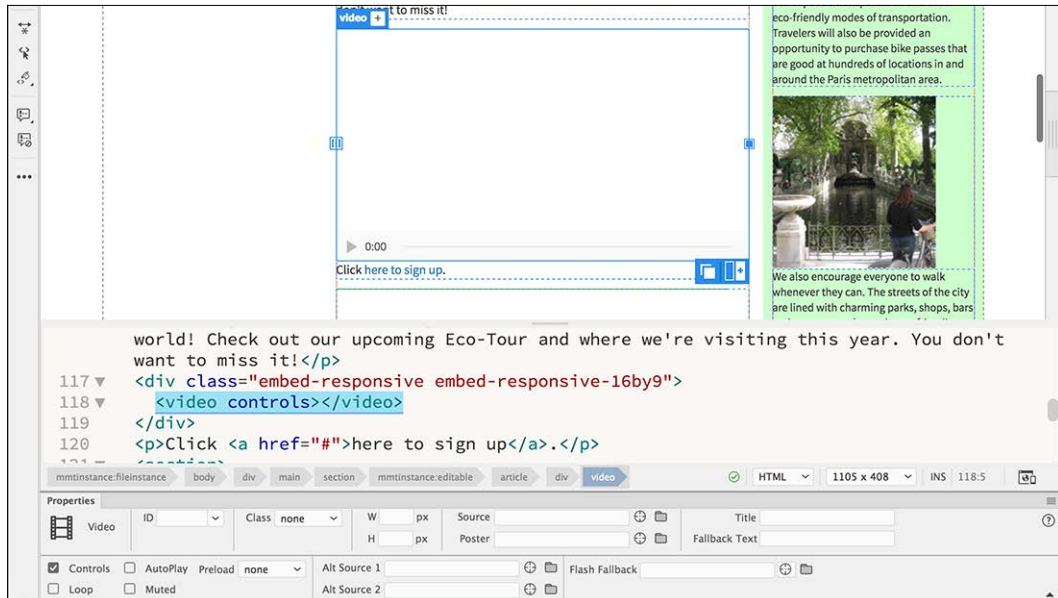The `<iframe>` is now selected. Let's replace it with the HTML5 `<video>` element.

7 Delete the entire `iframe` element.

The cursor is still in the location of the video element.

● **Note:** The video workflow requires the use of the Property inspector. If the Property inspector is not visible, select Window > Properties to display it.

8 Choose Insert > HTML > HTML5 Video.

The HTML5 `<video>` element appears. The Property inspector displays options for targeting the video source files. Note that this interface enables you to specify up to three video source files and one Flash fallback file. The first step is to select your primary video source.

9 In the Property inspector, click the Browse icon in the Source field. Navigate to the **movies** folder and select the **paris.mp4** file. Click OK/Open.
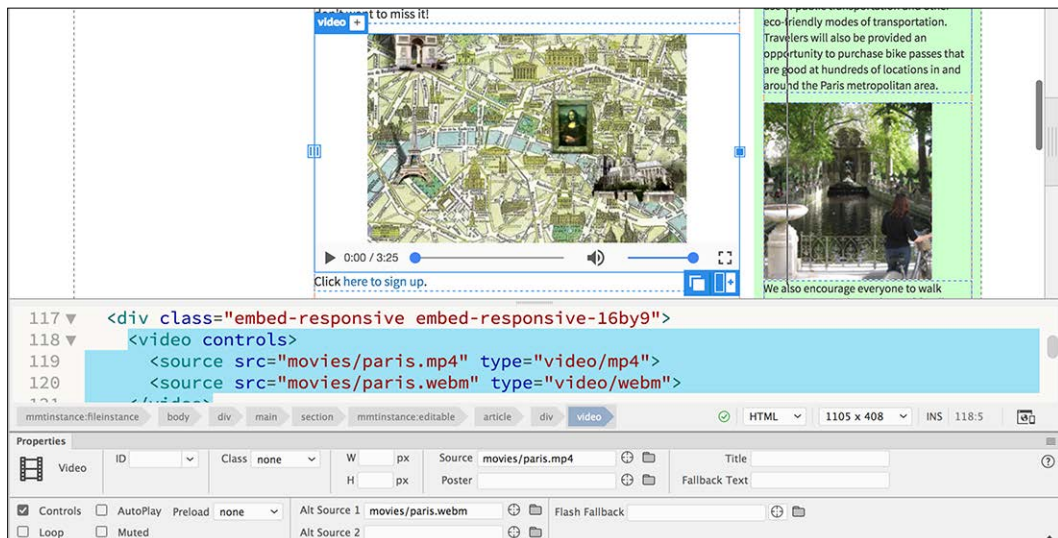
The MP4 video displays, replacing the original placeholder.

Now let's enter the width and height of the video. Although you will use CSS to control the height and width of the video, entering width and height attributes is recommended to help render your page faster in the browser.

10 In the Property inspector, enter **400** in the width (W) field and **300** in the height (H) field.
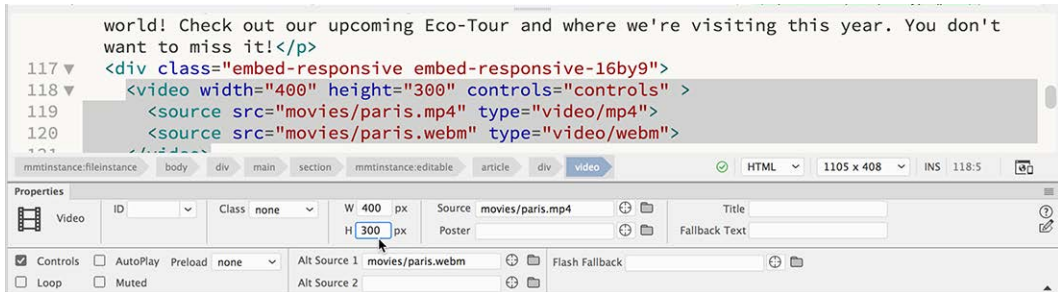
The attributes entered didn't seem to affect the size of the video; it still looks a bit small. You'll address this issue later when we make it responsive, but first you need to complete the <video> element. Best practices for HTML5 suggests you provide alternate source files.

Adding alternate video sources

The MP4 file format will be the primary video format loaded. MP4, also known as MPEG-4, is a video format based on Apple's QuickTime standard. It is supported natively by iOS-compatible devices, such as the iPhone and iPad, as well as by Apple's Safari browser. Many experts advise loading MP4 files first; otherwise, iOS devices may ignore the video element altogether.

You may have noticed that Dreamweaver automatically inserted a WebM version of the movie as Alt Source 1. WebM is an open source, royalty-free video format sponsored by Google. It is compatible with the latest versions of Firefox, Chrome, Opera, Microsoft Edge, and Internet Explorer.

To round out our HTML5 video selections, the next format you'll load is a lossy, open source multimedia format: Ogg. It is designed for the distribution of multimedia content that is free of copyright and other media restrictions.

● **Note:** Ogg is a container format. When the container contains a video, it uses the extension .ogv.

1 Click the Browse icon for the Alt Source 2 field.
Select the file **paris.theora.ogv** from the **movies** folder and click OK/Open.
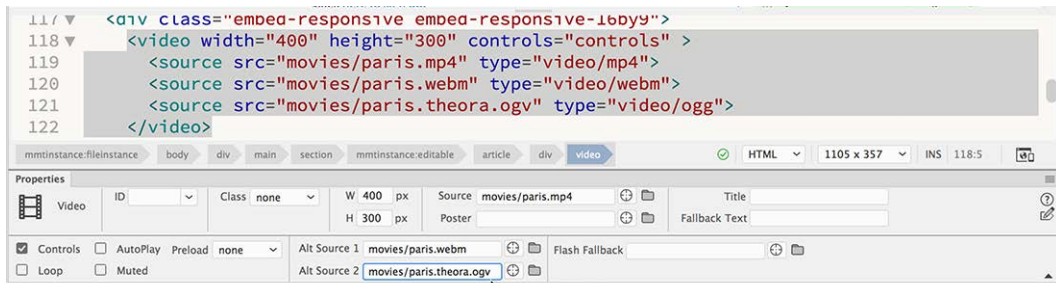
These three formats support all the modern desktop and mobile browsers.

2 Save the file.

3 If necessary, switch to Live view.

In some browsers, the `<video>` element won't generate a preview of the video content. You should add an image placeholder by using the Poster field in the Property inspector.

4 If necessary, select the `<video>` tag selector.
In the Property inspector, click Browse in the Poster field.

5 In the **movies** folder, select **paris-poster.png** and click OK/Open.
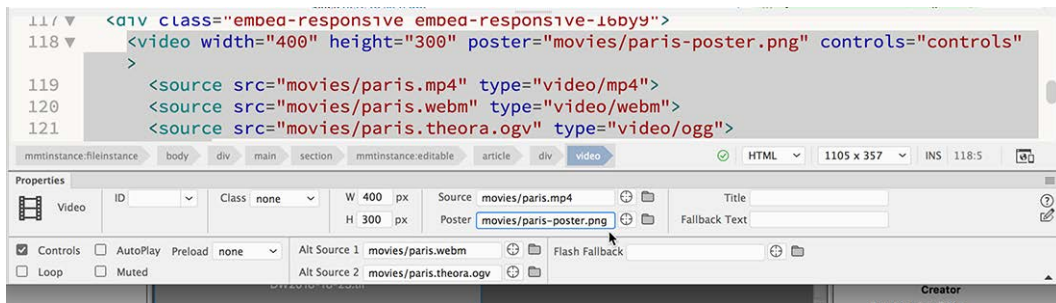
A preview image has been applied to the `<video>` element. The preview image appears only if a visitor uses a browser that doesn't support HTML5 video.

By supplying multiple video sources and a poster, you are ensuring that something will always appear in the browser whenever this file loads.

6 Save all files.

Video controls may appear below the element depending on what video format is displayed. In the next exercise, you will learn how to configure these controls and how the video will respond to the user.

Setting HTML5 video options

One of the final steps for configuring the video is to decide what other HTML5-supported options to specify. The options are displayed within the Property inspector whenever the `<video>` element is chosen. The options are selectable in all views.

1 If necessary, open **travel.html** in Live view.
Select the `<video>` tag selector.
Observe the left side of the Property inspector.

  - **Controls** displays visible video controls.

  - **AutoPlay** starts the video automatically after the webpage loads.

  - **Loop** causes the video to replay from the beginning automatically once it finishes.

  - **Muted** silences the audio.

  - **Preload** specifies the method in which the video loads.

2 If necessary, select the Controls option and deselect the AutoPlay, Loop, and Muted options. Set Preload to **None**.

● **Note:** The Preload option has to be selected manually. It appears to already be selected, but the option is not added to the element until you select it.
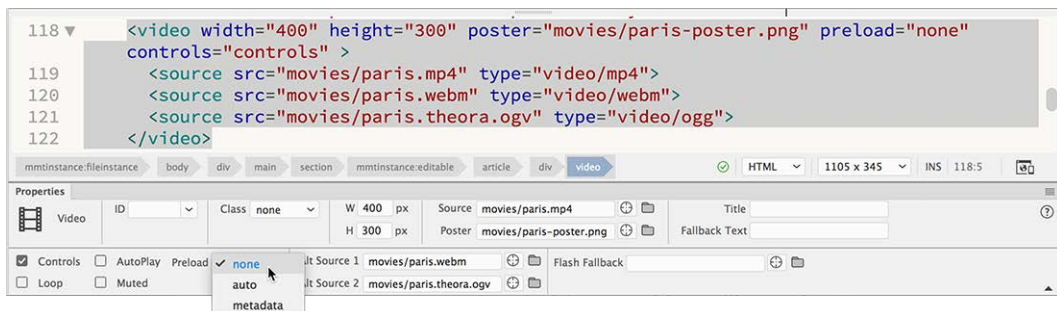
3 Save the file.

Video is very memory- and bandwidth-intensive. This is especially true for phones and tablets. Setting Preload to None prevents any video resources from downloading until the user actually clicks to play the video. It may require a few extra seconds for the video to download when launched, but your visitors will appreciate that you are respectful of their minutes and data plan.

Although these settings complete the structure of the `<video>` element, you still need to make sure the video works properly on all screens and devices.

Making the video responsive

The `<video>` element is complete, but you may have noticed that the preview doesn't seem to match the dimensions you entered earlier. That's because the CSS controlling the Bootstrap structure of the element has already taken over from the HTML attributes. Unfortunately, the default settings for the embedded video are expecting a video with an aspect ratio of 16:9. The one you're using is actually 4:3. Luckily, this is an easy fix.

1 If necessary, open **travel.html** in Split view.
Make sure that the program fills the entire display and that the document is at least 1100 pixels wide.

2 In the Code view window locate the `<video>` element (around line 118).

```
117 ▼  <div class="embed-responsive embed-responsive-16by9">
118 ▼     <video width="400" height="300" poster="movies/paris-poster.png" preload="none"
          controls="controls" >
119          <source src="movies/paris.mp4" type="video/mp4">
120          <source src="movies/paris.webm" type="video/webm">
121          <source src="movies/paris.theora.ogv" type="video/ogg">
122       </video>
123    </div>
```

mmtinstance:fileinstance body div main section mmtinstance:editable article div   ⊘ HTML ▾ 1105 x 288 ▾ INS 117:53

Note the parent `<div>` element.

This is the responsive Bootstrap embed element. It controls the size of the video on all screens and devices. Note the classes assigned to it. The second class, `embed-responsive-16by9`, designates this structure for an aspect ratio of 16:9.

3 Edit the class as highlighted: `embed-responsive-4by3`

```
117 ▼  <div class="embed-responsive embed-responsive-4by3">
118 ▼     <video width="400" height="300" poster="movies/paris-poster.png" preload="none"
          controls="controls" >
```

As soon as you complete the change, the video preview expands to fill the entire column. There's only one modification left to do. The original `<iframe>` had a responsive Bootstrap class assigned to it. To complete the responsive styling, you need to apply the same class to the new HTML5 `<video>` element.

4 Select the `video` tag selector.

5 In the Property inspector Class menu, select
`embed-responsive-item`.
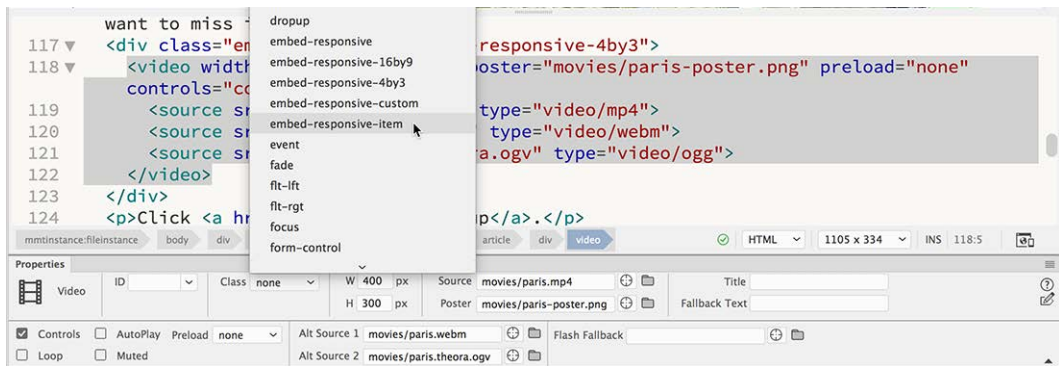
6 Save all files.

7 Preview the page in Live view or in a browser. If the video controls are not visible, move your cursor over the still image to display them. Click the Play button to view the movie.

Depending on where you preview the page, you will see one of the four video formats or the static poster image. For example, in Live view you will see the MP4-based video. The controls will also appear differently depending on what format is displayed. This movie has no sound, but the controls will often include a speaker button to adjust the volume or mute the audio.

8 In Dreamweaver, drag the Scrubber to the left to check the display of the video at various screen widths.
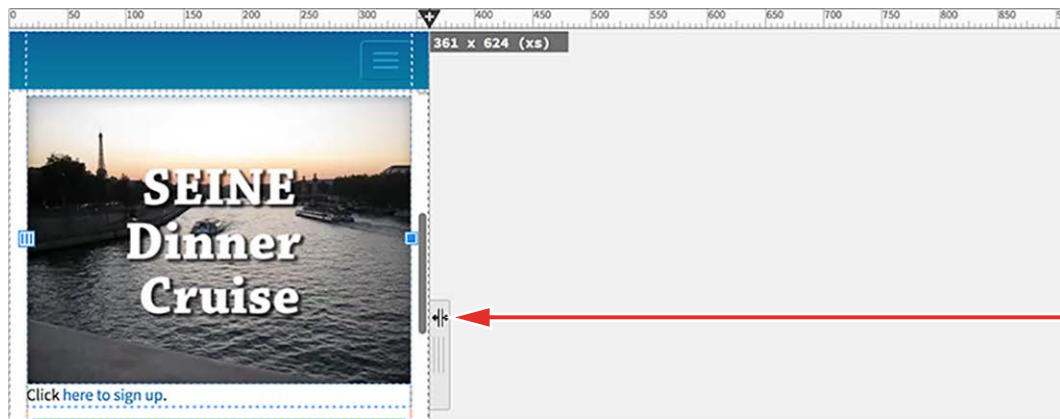
The video resizes as needed to fit the available space.

You've embedded three HTML5-compatible videos, which gives you support for most browsers and devices that can access the Internet, and a static poster for the rest. But you've learned only one possible technique for supporting this evolving standard. To learn more about HTML5 video and how to implement it, check out the following links:

• http://tinyurl.com/video-HTML5-1

• http://tinyurl.com/video-HTML5-2

• http://tinyurl.com/video-HTML5-3

To learn more about implementing video for mobile devices, check out these links:

• http://tinyurl.com/fluid-video

• http://tinyurl.com/fluid-video-1

Don't host your own videos

After showing you how to insert and host your own web-compatible videos, it's a good time to tell you that many experts think that hosting your own videos is not a good policy. For one thing, video formats have not been standardized across all browsers and devices. You can't simply upload one video format and support all your potential visitors. Millions of individuals and even large corporations have given up hosting their own videos and contracted with video-hosting services like YouTube or Vimeo.

One advantage of using a hosting service is that you have to upload only a single video format. They will handle the conversion of your video to any other formats needed. And if the standards change in the future, many services will convert your videos for you as needed. At least that's what happened at YouTube when Flash was dumped a few years ago.

For noncommercial users, you can usually host many gigabytes of video for free or for little cost. Some pay to remove advertising or the logos of the hosting services. You may even partner with these services to make money for yourself or your company.

Check out http://tinyurl.com/do-not-host-video to learn why some people think that hosting your own video is not the best plan. Check out http://tinyurl.com/video-hosting-overview for an overview of several hosting services.

Review questions

1 What advantage does HTML5 have over HTML 4 regarding web-based media?

2 What programming language(s) created the HTML5-compatible animation used in this lesson?

3 True or false: To support all web browsers, you can select a single video format.

4 In browsers or devices that do not support video, what can you do to provide some form of content to these users?

5 What advantage do video-hosting services offer over doing it yourself?

Review answers

1 HTML5 has built-in support for web animation, video, and audio.

2 The animation used in this lesson was created by Adobe Animate natively, using HTML5, CSS3, and JavaScript.

3 False. A single format supported by every browser has not emerged. Developers recommend incorporating four video formats to support the majority of browsers: MP4, WebM, and Ogg.

4 You can add a static poster image (GIF, JPEG, or PNG) via an option in the Property inspector to provide a preview of the video content in incompatible browsers and devices.

5 Most video-hosting services require you to upload only a single video format. They will convert it as needed to support all the various browsers and devices.