HTML. Unit 5. Images.

1. Introduction

In the beginning, the Web was just text, and it was really quite boring. Fortunately, it wasn't too long before the ability to embed images (and other more interesting types of content) inside web pages was added. There are other types of multimedia to consider, but it is logical to start with the humble `<img>` element, used to embed a simple image in a webpage. In this unit we'll look at how to use it in depth, including the basics, annotating it with captions using `<figure>` .

2. How do we put an image on a webpage?

In order to put a simple image on a webpage, we use the `<img>` element. This is a void element (meaning that it has no text content or closing tag) that requires a minimum of one attribute to be useful: `src` (sometimes spoken as its full title, *source*). The `src` attribute contains a path pointing to the image you want to embed in the page, which can be a relative or absolute URL, in the same way as `href` attribute values in `<a>` elements.

So for example, if your image is called `dinosaur.jpg` and it sits in the same directory as your HTML page, you could embed the image like so:

```
<img src="dinosaur.jpg" />
```

If the image was in an `images` subdirectory, which was inside the same directory as the HTML page (which Google recommends for SEO/indexing purposes), then you'd embed it like this:

```
<img src="images/dinosaur.jpg" />
```

Search engines also read image filenames and count them towards SEO. Therefore, you should give your image a descriptive filename; `dinosaur.jpg` is better than `img835.jpg`.

You also could embed the image using its absolute URL, for example:

```
<img src="https://www.example.com/images/dinosaur.jpg" />
...
<img src="https://developer.mozilla.org/static/img/favicon144.png" />
```

3. Alternative text

The next attribute we'll look at is `alt`. Its value is supposed to be a textual description of the image, to be used by search engines, and also in situations where the image cannot be seen/displayed or takes a long time to render because of a slow internet connection. Keep in mind that an `alt` attribute's value should clearly and concisely describe the image's content. For example, our above code could be modified like so:

```
<img src="images/dinosaur.jpg"
```

```
    alt="The head and torso of a dinosaur skeleton;
         it has a large head with long sharp teeth" />
...
<img src="https://developer.mozilla.org/static/img/favicon144.png"
    alt="MDN logo" />
```

The easiest way to test your `alt` text is to purposely misspell your filename. If for example our image name was spelled `dinosooooor.jpg`, the browser wouldn't display the image, and would display the alt text instead.

So, why would you ever see or need alt text? It can come in handy for a number of reasons:

- The user is visually impaired, and is using a screen reader to read the web out to them. In fact, having alt text available to describe images is useful to most users.
- As described above, the spelling of the file or path name might be wrong.
- The browser doesn't support the image type. Some people still use text-only browsers, such as Lynx, which displays the alt text of images.
- You may want to provide text for search engines to utilize; for example, search engines can match alt text with search queries.
- Users have turned off images to reduce data transfer volume and distractions. This may be especially common on mobile phones, and in countries where bandwidth is limited or expensive.

What exactly should you write inside your `alt` attribute? It depends on *why* the image is there in the first place. In other words, what you lose if your image doesn't show up:

- **Content.** If your image provides significant information, provide the same information in a *brief* `alt` text – or even better, in the main text which everybody can see. Don't write redundant `alt` text. How annoying would it be for a sighted user if all paragraphs were written twice in the main content? If the image is described adequately by the main text body, you can just use `alt=""`.
- **Link.** If you put an image inside `<a>` tags, to turn an image into a link, you still must provide accessible link text. In such cases you may, either, write it inside the same `<a>`

element, or inside the image's `alt` attribute – whichever works best in your case.

- **Text.** You should not put your text into images. If your main heading needs a drop shadow, for example, use CSS for that rather than putting the text into an image. However, If you *really can't avoid doing this*, you should supply the text inside the `alt` attribute.

Essentially, the key is to deliver a usable experience, even when the images can't be seen. This ensures all users are not missing any of the content. Try turning off images in your browser and see how things look. You'll soon realize how helpful alt text is if the image cannot be seen.

4. Width and height

You can use the `width` and `height` attributes to specify the width and height of your image. We could do this:

```
<img src="images/dinosaur.jpg"
     alt="The head and torso of a dinosaur skeleton;
          it has a large head with long sharp teeth"
     width="400"
     height="340" />
```

However, you shouldn't alter the size of your images using always HTML attributes. If you set the image size too big, you'll end up with images that look grainy, fuzzy, or too small, and wasting bandwidth downloading an image that is not fitting the user's needs. The image may also end up looking distorted, if you don't maintain the correct aspect ratio. You should use an image editor to put your image at the correct size before putting it on your webpage, but if you do need to alter an image's size, you should use CSS instead.

5. Image titles

As with links, you can also add `title` attributes to images, to provide further supporting information if needed. In our example, we could do this:

```
<img src="images/dinosaur.jpg"
     alt="The head and torso of a dinosaur skeleton;
          it has a large head with long sharp teeth"
     title="A T-Rex on display in the Manchester University Museum" />
```

This gives us a tooltip on mouse hover, just like link titles. However, it should not be used as a replacement of the text inside the `alt` attribute. The `title` has a number of accessibility problems, mainly based around the fact that screen reader support is very unpredictable and most browsers won't show it unless you are hovering with a mouse (so e.g. no access to keyboard users).

6. Proposed exercise: Embedding images

> *Create a web page to display several images. You have to use the basic* `<img>` *tag and the* `src` *attribute to point to the URL of each individual image. Por example, you can use images like these ones:*
>
> *https://developer.mozilla.org/static/img/favicon144.png*
>
> *https://raw.githubusercontent.com/mdn/learning-area/master/html/multimedia-and-embedding/images-in-html/dinosaur_small.jpg*
>
> *https://validator.w3.org/images/w3c.png*
>
> *You must also add some alt text to each image, and check that they work by misspelling the image URL's. And finally experiment with different values of*

> *width and height to see what the effect is.*

7. Image link

This example builds upon the previous ones, showing how to turn the image into a link. To do so, nest the `<img>` tag inside the `<a>` . You should use the alternative text to describe the resource the link is pointing to, as if you were using a text link instead:

```
<a href="https://developer.mozilla.org">
  <img src="https://developer.mozilla.org/static/img/favicon144.png"
       alt="Visit the MDN site" />
</a>
```

8. Proposed exercise: Top ten pages

> *Create a web page showing an ordered list (`<ol>`) of the top ten web pages you like. You must use images to create links to each page, and set the* `height` *attribute to the same value for all the images so that all of them are the same height. For example:*

```
<ol>
  <li>
    <a href="https://developer.mozilla.org">
      <img src="https://developer.mozilla.org/static/img/favicon144.png"
           alt="MDN site" height="100" />
    </a>
  </li>
  <li>
    <a href="https://validator.w3.org/">
      <img src="https://validator.w3.org/images/w3c.png"
           alt="Markup Validation Service" height="100" />
    </a>
  </li>
  ...
```

```
</ol>
```

9. Annotating images with figures and figure captions

Speaking of captions, there are a number of ways that you could add a caption to go with your image. For example, there would be nothing to stop you from doing this:

```
<div class="figure">
  <img src="images/dinosaur.jpg"
       alt="The head and torso of a dinosaur skeleton;
            it has a large head with long sharp teeth"
       width="400"
       height="340">

  <p>A T-Rex on display in the Manchester University Museum.</p>
</div>
```

This is ok, since it contains the content you need, and can be nicely styled using CSS. But there is a problem here: there is nothing that semantically links the image to its caption, which can cause problems for screen readers. For example, when you have 50 images and captions, which caption goes with which image?

A better solution is to use the HTML5 `<figure>` and `<figcaption>` elements. These are created for exactly this purpose: to provide a semantic container for figures, and to clearly link the figure to the caption. Our above example could be rewritten like this:

```
<figure>
  <img src="images/dinosaur.jpg"
       alt="The head and torso of a dinosaur skeleton;
            it has a large head with long sharp teeth"
       width="400"
       height="340">

  <figcaption>A T-Rex on display in the Manchester University Museum.
</figcaption>
</figure>
```

The `<figcaption>` element tells browsers, and assistive technology that the caption describes the other content of the `<figure>` element. From an accessibility viewpoint, captions and `alt` text have distinct roles. Captions benefit even people who can see the image, whereas `alt` text provides the same functionality as an absent image. Therefore, captions and `alt` text shouldn't just say the same thing, because they both appear when the image is gone. Try turning images off in your browser and see how it looks.

Also keep in mind that a figure could be made of several images, a code snippet, audio, video, equations, a table, or something else. We will go through this in another unit.

10. Proposed exercise: Figures with captions

> *Create a web page to display several images you like, using the `<figure>` element, and add a caption to each one using the `<figcaption>` element, as it is done in the example below:*

```
<figure>
  <img src="https://developer.mozilla.org/static/img/favicon144.png"
       alt="Mozilla developer website">
  <figcaption>MDN Logo</figcaption>
</figure>

<figure>
  <img src="https://validator.w3.org/images/w3c.png"
       alt="W3C Validator">
  <figcaption>W3C Logo</figcaption>
</figure>

...
```

11. Proposed exercise: Top ten films

> *Create a new web page showing an ordered list (* `<ol>` *) of the top ten films you like. In this case you must use figures with captions to create links to each page. Set the* `height` *attribute to the same value for all the images so that all of them are the same height. For example:*

```
<ol>
  <li>
    <a href="https://en.wikipedia.org/wiki/Steve_Jobs_(film)">
      <figure>
        <img
src="https://upload.wikimedia.org/wikipedia/commons/4/49/SteveJobsMacbookA
             alt="Steve Jobs and Macbook Air" height="100" />
        <figcaption>Steve jobs</figcaption>
      </figure>
    </a>
  </li>
  <li>
    <a href="https://en.wikipedia.org/wiki/2001:_A_Space_Odyssey_(film)">
      <figure>
        <img src="https://upload.wikimedia.org/wikipedia/en/0/09/2001chil
             alt="Baby, space and Earth" height="100" />
        <figcaption>2001: A Space Odyssey</figcaption>
      </figure>
    </a>
  </li>
  ...
</ol>
```