Attributes + Hyperlinking

Using HTML Attributes and Linking to Other HTML Pages

We'll cover the following

cssNow that you understand the context in which HTML is used, we can begin to go over more details of HTML usage.

HTML attributes

HTML **attributes** provide additional information about an HTML element. Attributes can be considered as properties of the element. An element may have a single attribute, many attributes, or no attributes at all.

Let's take a look at an example heading with a `title` attribute:

```
<h2 title="This is a subheading">Hello, World!</h2>
```

Attributes are placed in the opening tag, with a space after the element declaration (or another attribute, if there are multiple) and are defined using the following format:

```
<tagName attribute_name="attribute_value"></tagName>
```

The attribute name is always followed by an `=` sign and the attribute value is always wrapped in quotation marks. There are no spaces between the attribute name, `=` sign, and the attribute value.

Check your Understanding

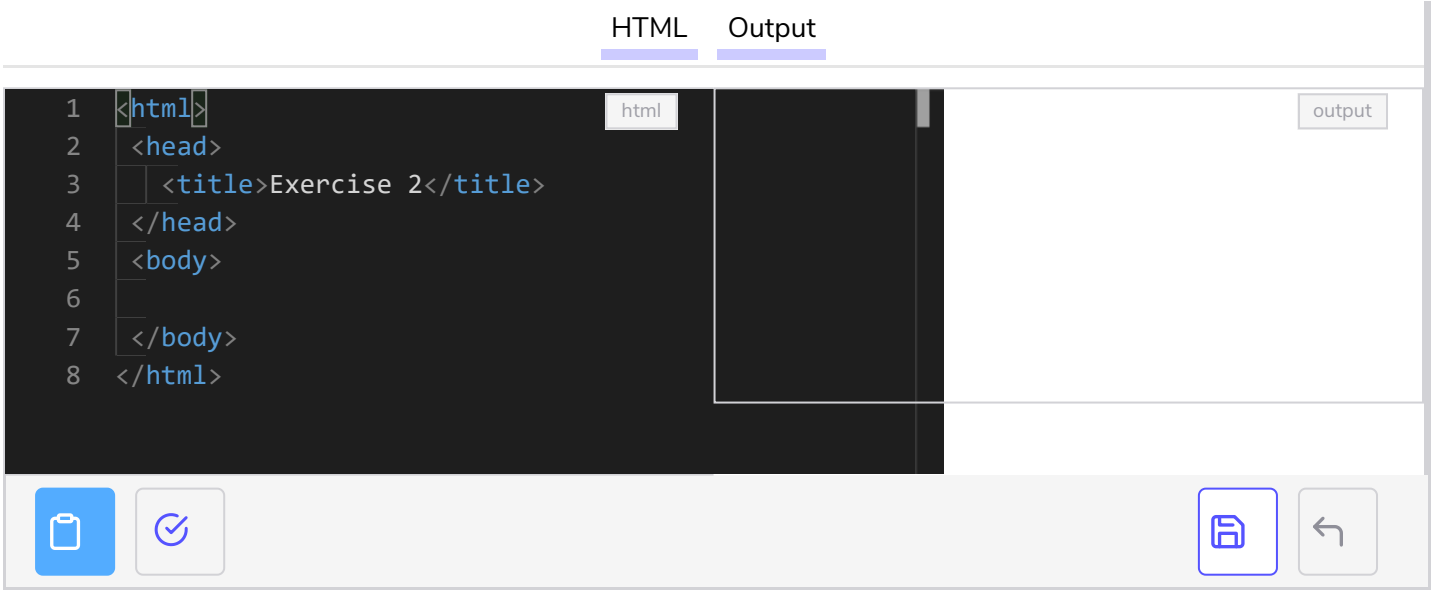
Q

Another type of HTML attribute is the “style” property, which can be used to give an element a custom style. How would we define a paragraph element with both title and style attributes?

Retake Quiz

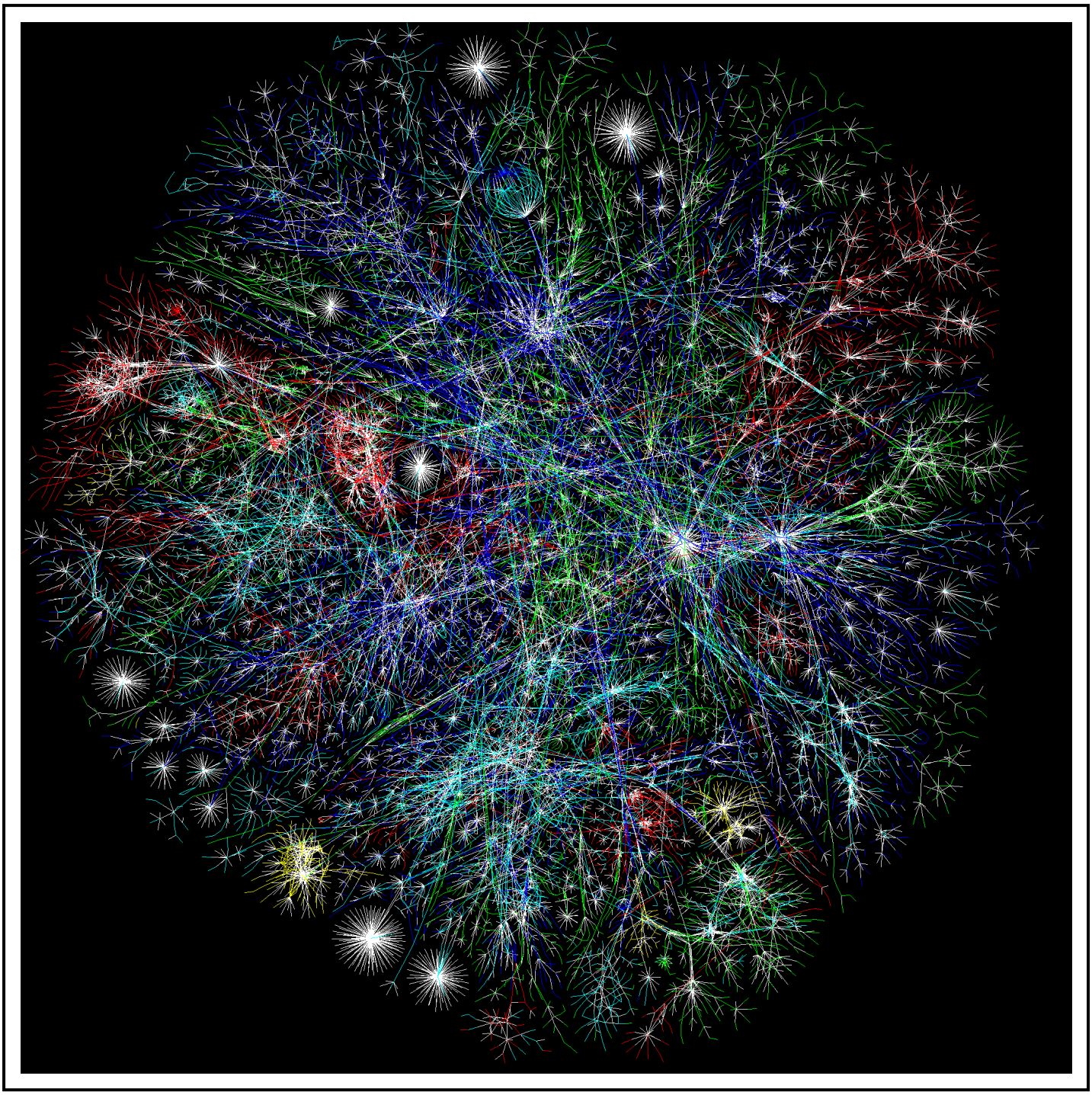
Exercise

Create a top-level header that has a `title` attribute of `Important Message!` and is colored green using the `style` attribute.



Anchor elements / hyperlinking

One of the most important aspects of the World Wide Web is the ability to *link* to other parts of the web. Without a way to redirect our HTML page to other web addresses, there really wouldn't be a "web" at all!



A network visualization of nodes on the World Wide Web

We can connect a HTML page to other web pages by creating a **hyperlink** using the *anchor* tag, like so:

```
<a href="http://www.google.com">Google</a>
```

The `href` *attribute* refers to **Hypertext Reference**, whose value is a **Uniform Resource Locator** (URL). A URL is basically fancy lingo for a web address, or the destination the link is pointing to. The `href` attribute can also refer to things like:

- Email Addresses (`mailto:someone@educative.io`)

- Phone Numbers (`tel:+18004444444`)
- Documents/Files (Give the URL of the file instead of a web page)
- Another different location on the same web page the browser is currently on

Relative vs absolute URL paths

It's important to understand how file paths play a role in how your hyperlinks will operate.

An **absolute** URL points to a single address that will direct to the same place regardless of where the original page is coming from. It looks something like this: `http://www.github.com/google`.

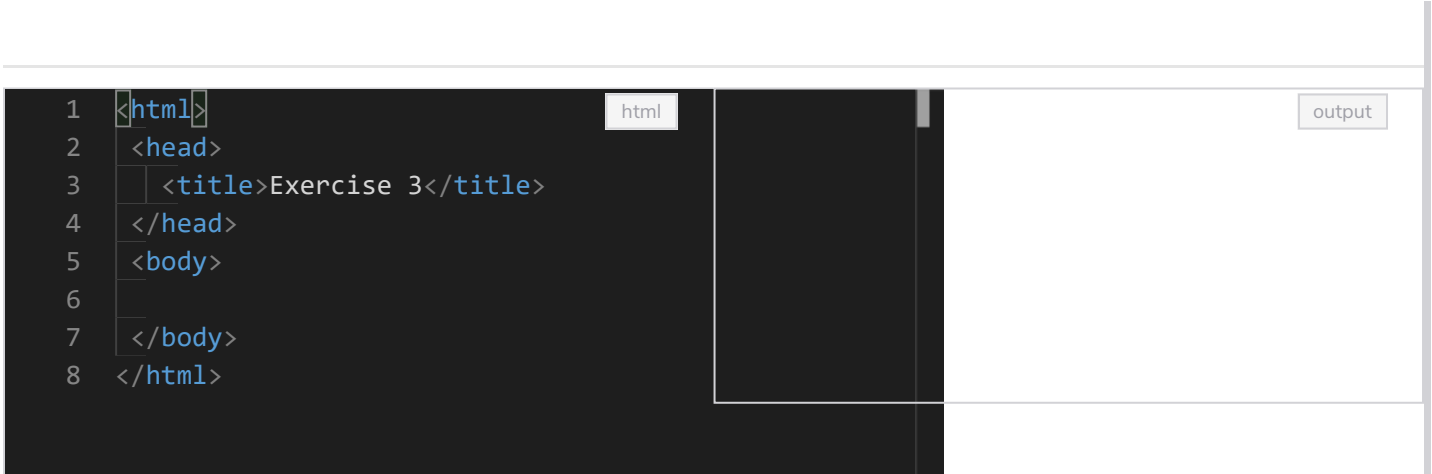
In an absolute URL path there are three main components:

1. The **Protocol**: What you most often see as `http://` or `https://` when you browse websites, but can be other things, like `file://` or `ftp://`
2. The **Domain**: The name of the website (in this example, `www.github.com`)
3. The **Path**: The directory (or folder) we wish to navigate to. This field is not always necessary, and generally allows us to navigate to a more specific portion of a domain (in this case, Google's profile on Github)

An absolute URL provides all the information necessary for a browser with an internet connection to reach the desired address. Furthermore, an absolute URL will not change its destination if used on different devices/browsers.

Exercise

Create an anchor HTML element with an *absolute* URL path using the `https` protocol to navigate to the domain name `www.buzzfeed.com`.



Relative URLs provide less information than absolute URLs and generally refer to pages on the same domain. Relative URLs are useful when you start to deal with multiple web pages on your site, and want a way to navigate between them.

Let's take a look at a quick example of a directory named `website` with:

- a main `index.html` page
- an about section, named `about.html`
- a nested directory named `blogPosts`, with three article HTML files named:
  - `article1.html`
  - `article2.html`
  - `article3.html`

If we started in the `website` directory on the `index.html` file, we could redirect to the About section by including the anchor tag:

```
<a href="about.html">About</a>
```

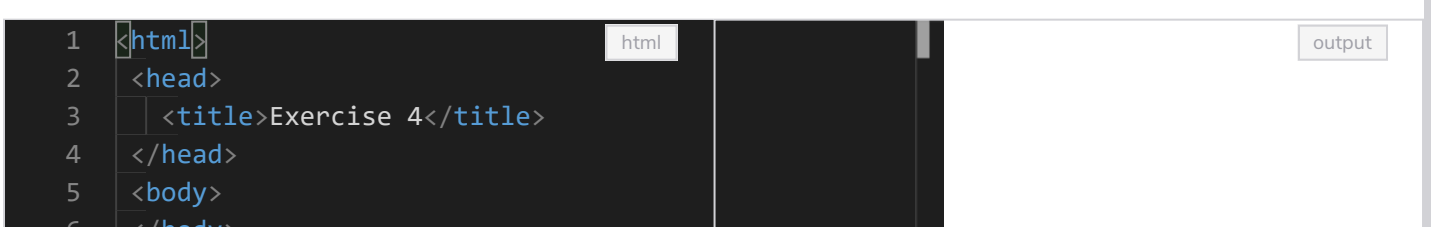
Now, say we want to navigate to an article in our `blogPost` folder. The relative URL path would then include the directory name: `blogPost/article2.html`. The entire anchor element would then be:

```
<a href="blogPost/article2.html">Article 2</a>
```

Now, how would we navigate back to the `index.html` page if we are in the `blogPost` directory? We can accomplish this by indicating the path to the file is one direct level up, like so: `../index.html`.

Exercise

Create an anchor HTML element with a *relative* URL path to the `about.html` page. Assume the page you are on is an article in the `blogPost` directory.



```
6   </body>
7   </html>
```


Check your Understanding

1. Absolute URLs do not require a protocol or domain to function correctly.

2. You can create a hyperlink to connect to a telephone number.

3. A relative URL generally consists of a file path, without a protocol or domain.

4

HTML pages must have an anchor tag in order to function properly.

5

An HTML element can have many attributes.

Retake Quiz

Now that you have learned about attributes and hyperlinking, let’s learn about adding heading and list in a html page.