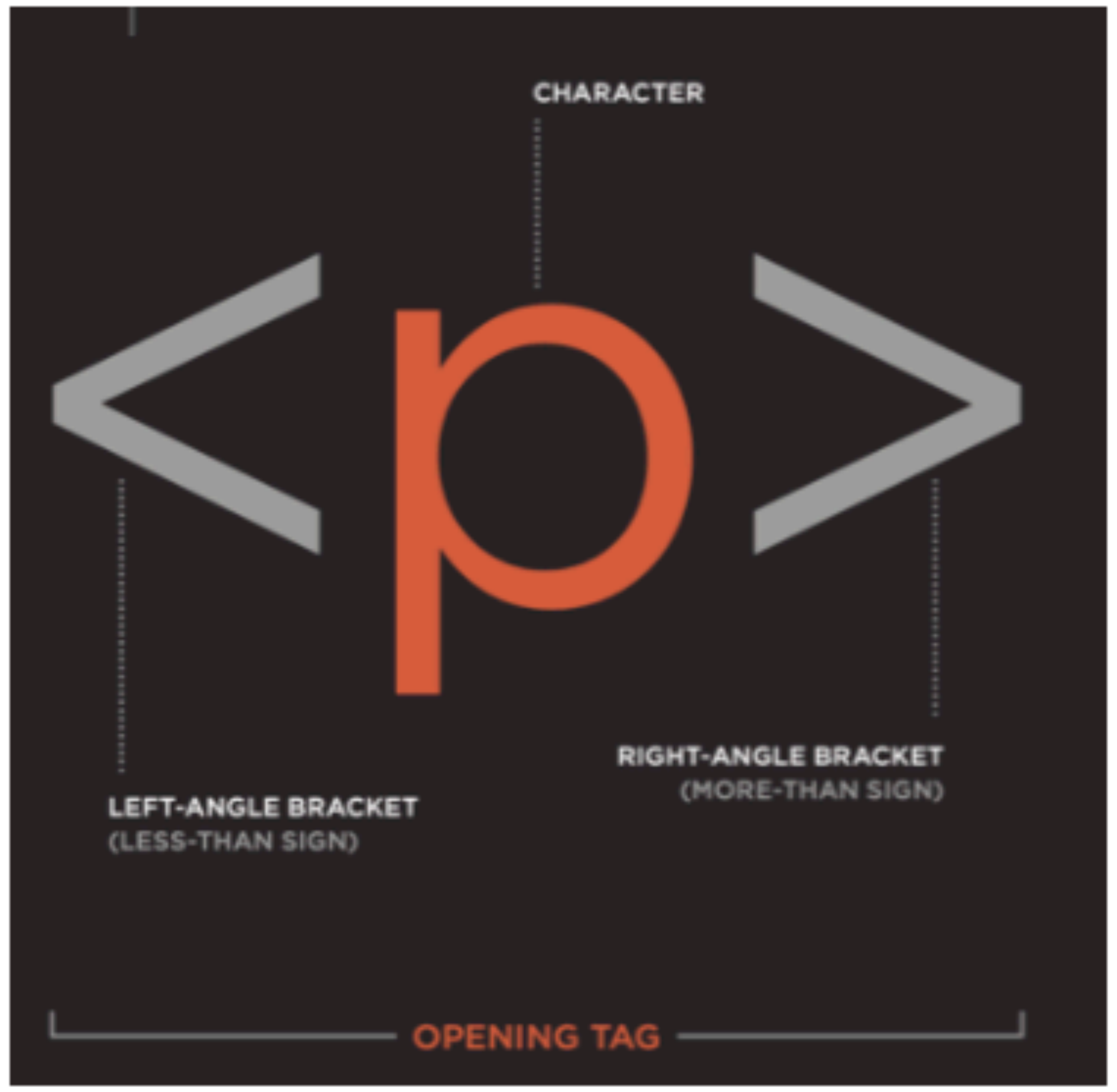
HTML : Elements & Linking



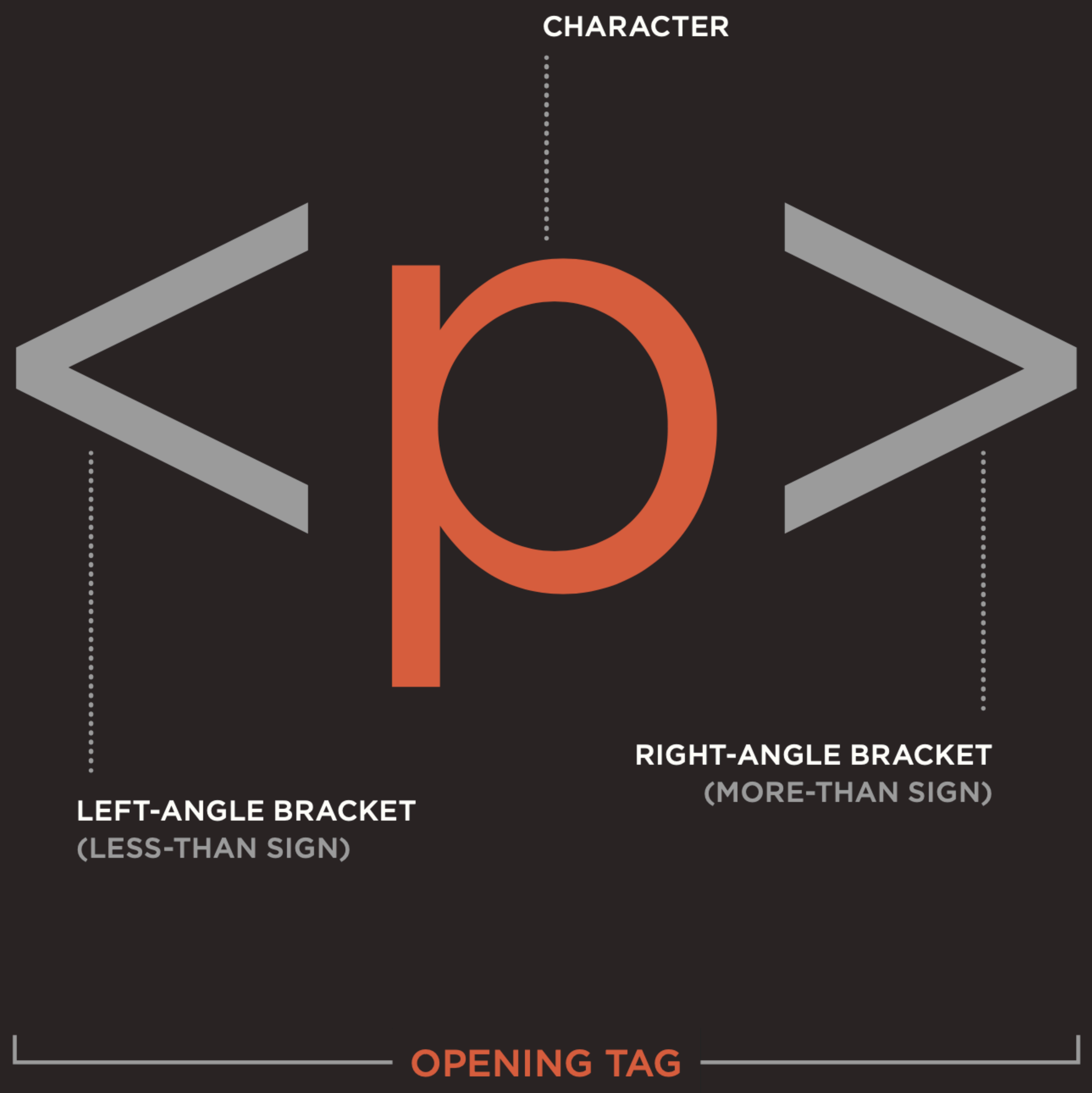
Element tags and attributes. The structure of links

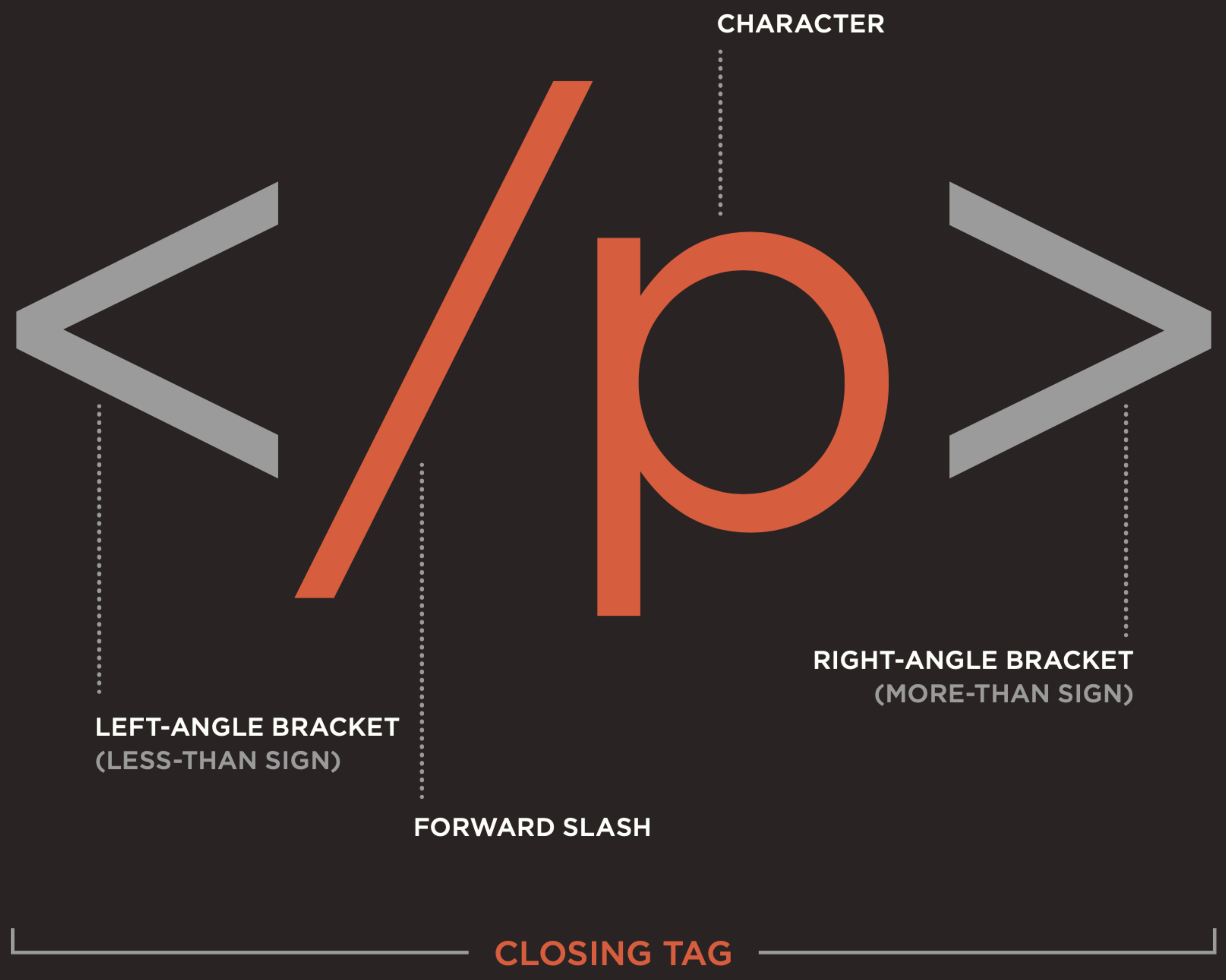
Some Key Concepts

- The structure of an ***HTML Element*** its variants.
- Nature of a ***relative path***, the differences from an ***absolute path***.

Agenda

- Elements, Attributes, & Documents
- Linking
- Nesting
- Line break, Block & Inline Elements





The characters in the brackets indicate the tag's purpose.

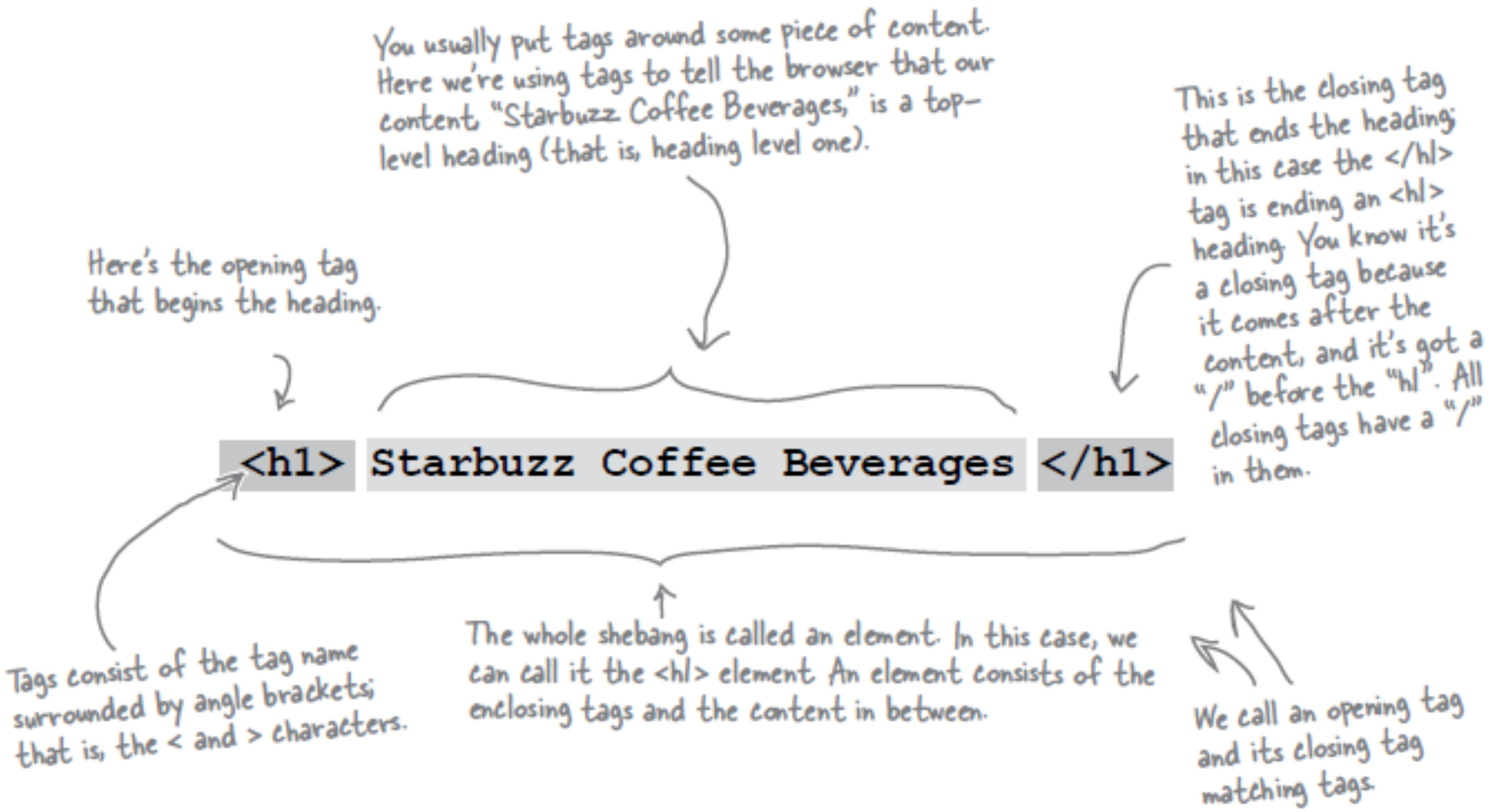
For example, in the tags above the p stands for paragraph.

The closing tag has a forward slash after the the < symbol.

The terms "tag" and "element" are often used interchangeably.

Strictly speaking, however, an element comprises the opening tag *and* the closing tag *and* any content that lies between them.

Components of an HTML Element



You usually put tags around some piece of content. Here we're using tags to tell the browser that our content, "Starbuzz Coffee Beverages," is a top-level heading (that is, heading level one).

Here's the opening tag that begins the heading.

This is the closing tag that ends the heading; in this case the </h1> tag is ending an <h1> heading. You know it's a closing tag because it comes after the content, and it's got a "/" before the "h1". All closing tags have a "/" in them.

Tags consist of the tag name surrounded by angle brackets; that is, the < and > characters.

The whole shebang is called an element. In this case, we can call it the <h1> element. An element consists of the enclosing tags and the content in between.

We call an opening tag and its closing tag matching tags.

Components of an HTML Element

<ElementName >	Start Tag
Content	
</ElementName>	End Tag

<title>

<title> My App Store </title>

ElementName: *<title>*

Content: *My App Store*

ElementName: *</title>*

<p>

```
<p>
  This store brings you great app bundles
  week after week. We select the best power
  user apps from a broad range of suppliers
  and combine them into great deals. These
  are the highest quality apps
  from the best publishers, at great
  prices.
</p>
```

ElementName:	*<p>*
Content:	`This store brings you great app bundles week after week. We select the best power user apps from a broad range of suppliers and combine them into great deals. These are the highest quality apps from the best publishers, at great prices.`
ElementName:	*</p>*

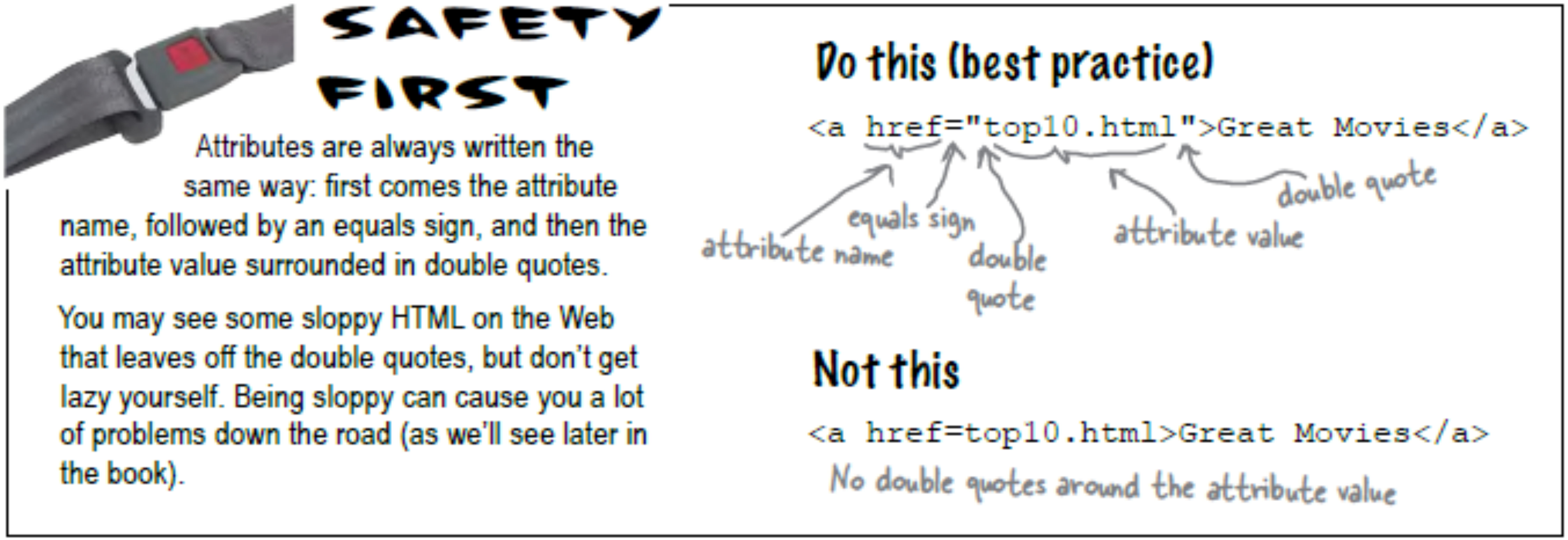
<a>

```
<a href="apps.html"> App Store </a>
```

ElementName:	*<a>*
AttributeName:	`href`
`AttributeValue:`	`“apps.html”`
Content:	*App Store*
ElementName:	*</a>*

Attributes

- Attributes give you a way to specify additional information about an element.



SAFETY FIRST

Attributes are always written the same way: first comes the attribute name, followed by an equals sign, and then the attribute value surrounded in double quotes.

You may see some sloppy HTML on the Web that leaves off the double quotes, but don't get lazy yourself. Being sloppy can cause you a lot of problems down the road (as we'll see later in the book).

Do this (best practice)

```
<a href="top10.html">Great Movies</a>
```

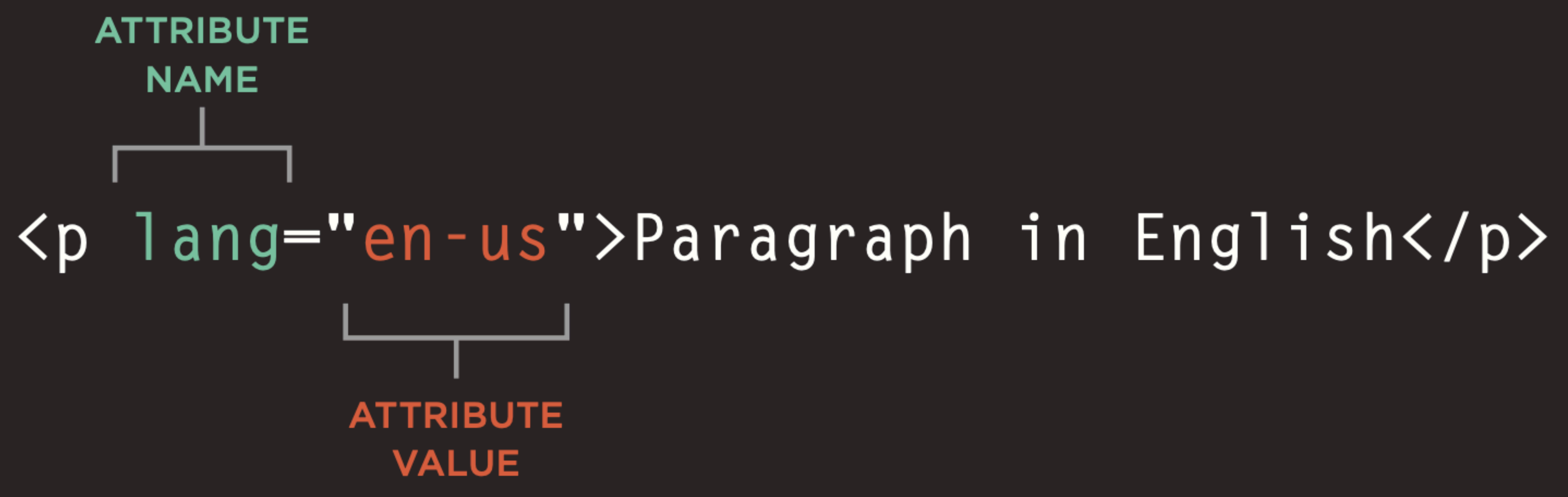
Not this

```
<a href=top10.html>Great Movies</a>
```

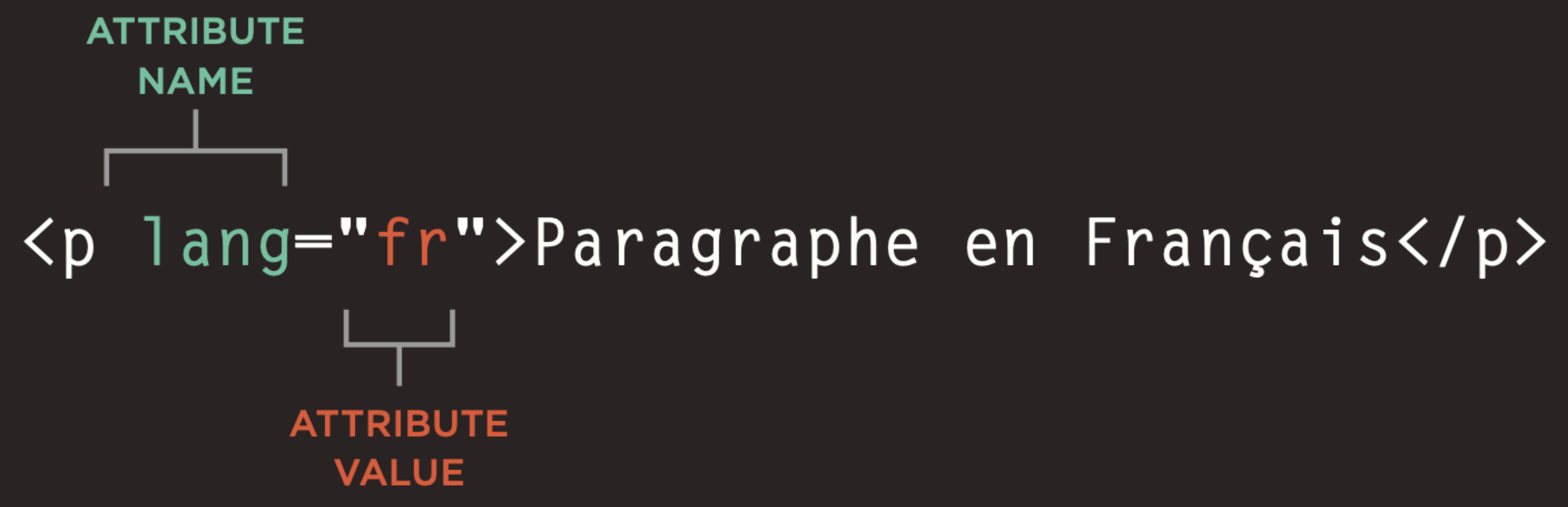
No double quotes around the attribute value

Attributes provide additional information about the contents of an element. They appear on the opening tag of the element and are made up of two parts: a name and a value, separated by an equals sign.

HTML5 allows you to use uppercase attribute names and omit the quotemarks, but this is not recommended.



```
<p lang="en-us">Paragraph in English</p>
```



```
<p lang="fr">Paragraphe en Français</p>
```

The attribute **name** indicates what kind of extra information you are supplying about the element's content. It should be written in lowercase.

The **value** is the information or setting for the attribute. It should be placed in double quotes. Different attributes can have different values.

Here an attribute called `lang` is used to indicate the language used in this element. The value of this attribute on this page specifies it is in US English.

The majority of attributes can only be used on certain elements, although a few attributes (such as `lang`) can appear on any element.

Most attribute values are either pre-defined or follow a stipulated format. We will look at the permitted values as we introduce each new attribute.

The value of the `lang` attribute is an abbreviated way of specifying which language is used inside the element that all browsers understand.

<img>

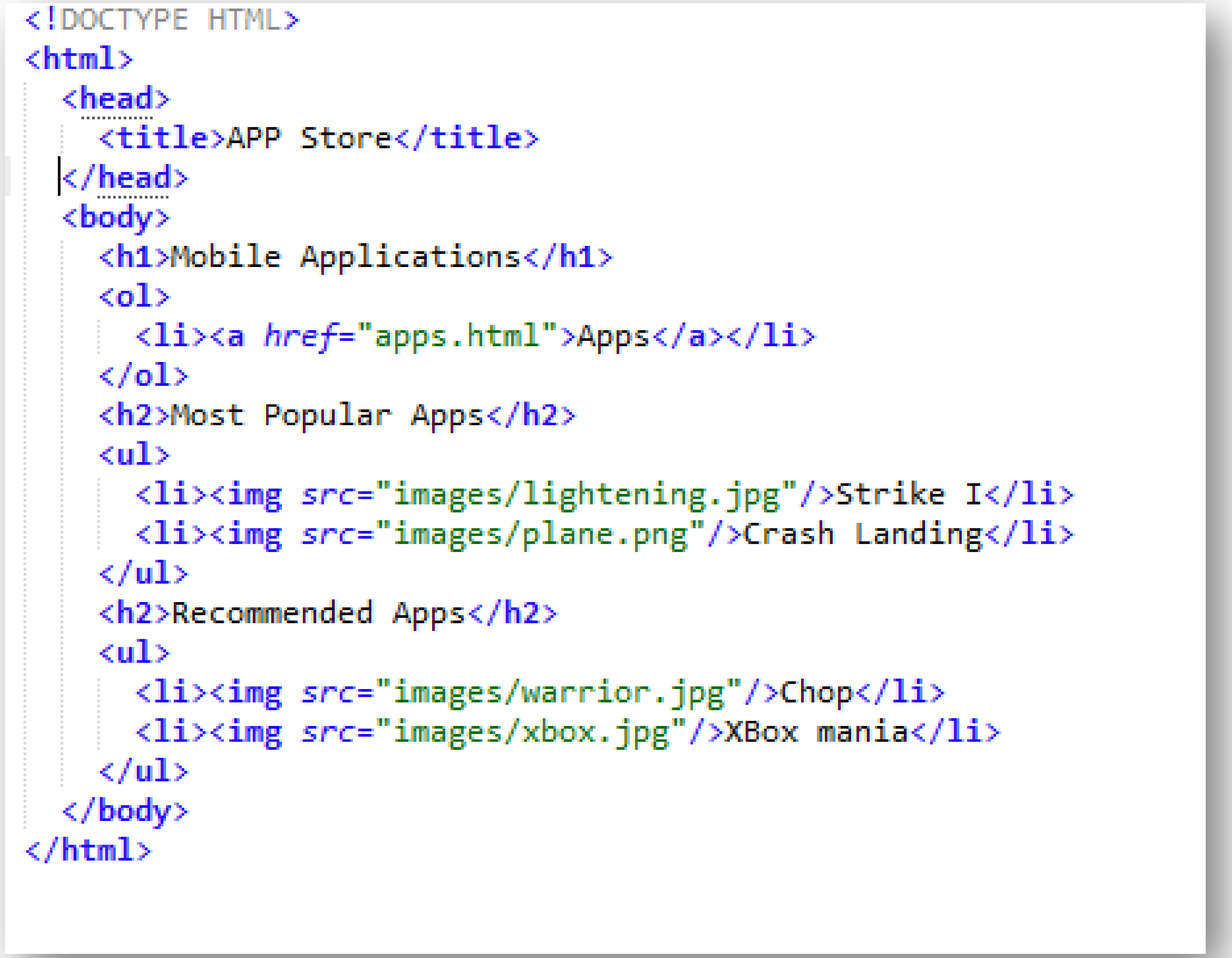
```
<img src=“../images/delete.jpg”/>
```

ElementName:	*<img>*
AttributeName:	`src`
`AttributeValue:`	`"../images/delete.jpg"`
Content:	*empty*
ElementName:	*none*

```
<img src=“../images/delete.jpg”>
```

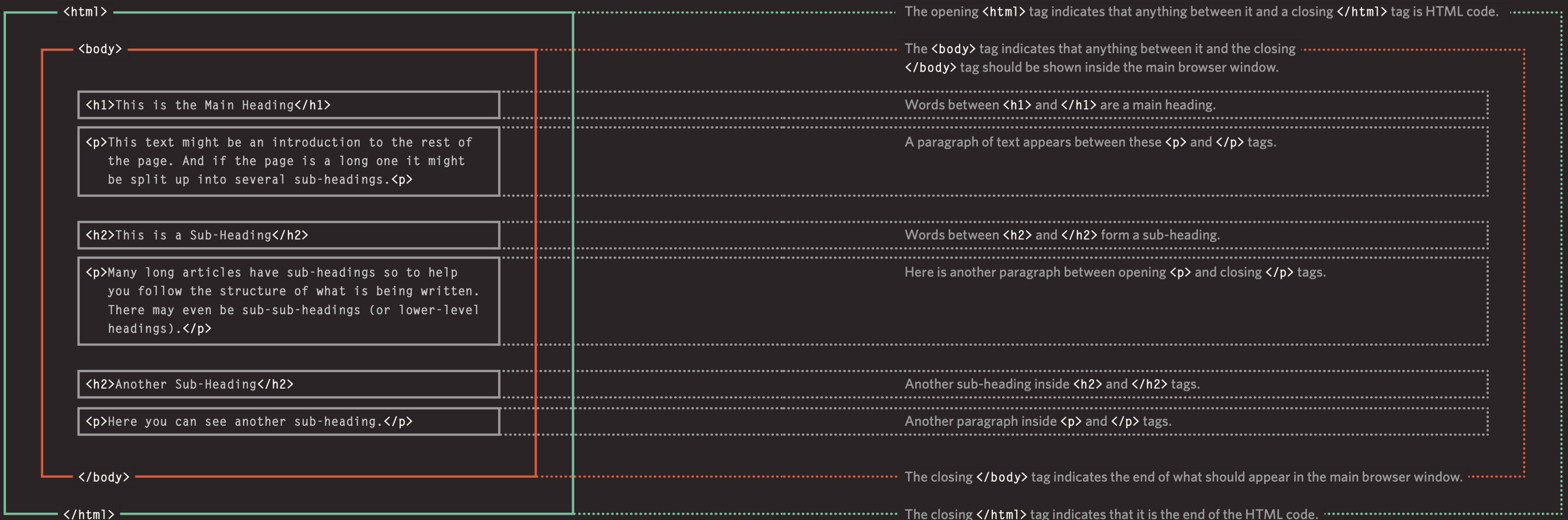
HTML Document Structure

- html
  - head
    - title
  - body
    - h1
    - ol
    - etc...



```
<!DOCTYPE HTML>
<html>
  <head>
    <title>APP Store</title>
  </head>
  <body>
    <h1>Mobile Applications</h1>
    <ol>
      <li><a href="apps.html">Apps</a></li>
    </ol>
    <h2>Most Popular Apps</h2>
    <ul>
      <li><img src="images/lightening.jpg"/>Strike I</li>
      <li><img src="images/plane.png"/>Crash Landing</li>
    </ul>
    <h2>Recommended Apps</h2>
    <ul>
      <li><img src="images/warrior.jpg"/>Chop</li>
      <li><img src="images/xbox.jpg"/>XBox mania</li>
    </ul>
  </body>
</html>
```

Tags act like containers. They tell you something about the information that lies between their opening and closing tags.



```
<html>
  <body>
    <h1>This is the Main Heading</h1>
    <p>This text might be an introduction to the rest of
       the page. And if the page is a long one it might
       be split up into several sub-headings.<p>
    <h2>This is a Sub-Heading</h2>
    <p>Many long articles have sub-headings so to help
       you follow the structure of what is being written.
       There may even be sub-sub-headings (or lower-level
       headings).</p>
    <h2>Another Sub-Heading</h2>
    <p>Here you can see another sub-heading.</p>
  </body>
</html>
```

DESCRIPTION

The opening `<html>` tag indicates that anything between it and a closing `</html>` tag is HTML code.

The `<body>` tag indicates that anything between it and the closing `</body>` tag should be shown inside the main browser window.

Words between `<h1>` and `</h1>` are a main heading.

A paragraph of text appears between these `<p>` and `</p>` tags.

Words between `<h2>` and `</h2>` form a sub-heading.

Here is another paragraph between opening `<p>` and closing `</p>` tags.

Another sub-heading inside `<h2>` and `</h2>` tags.

Another paragraph inside `<p>` and `</p>` tags.

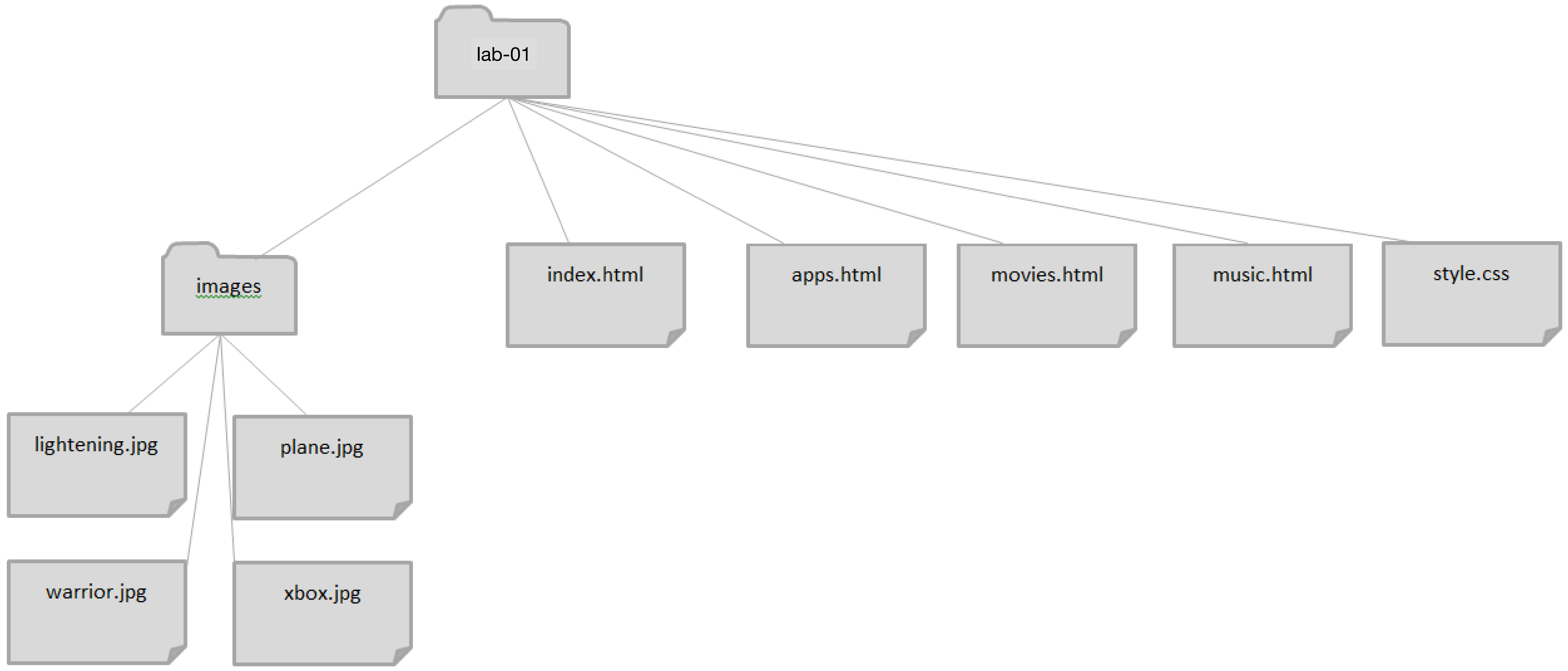
The closing `</body>` tag indicates the end of what should appear in the main browser window.

The closing `</html>` tag indicates that it is the end of the HTML code.

Agenda

- Elements, Attributes, & Documents
- Linking
- Nesting
- Line break, Block & Inline Elements

Linking


lab-01
images
index.html
apps.html
movies.html
music.html
style.css
lightening.jpg
plane.jpg
warrior.jpg
xbox.jpg

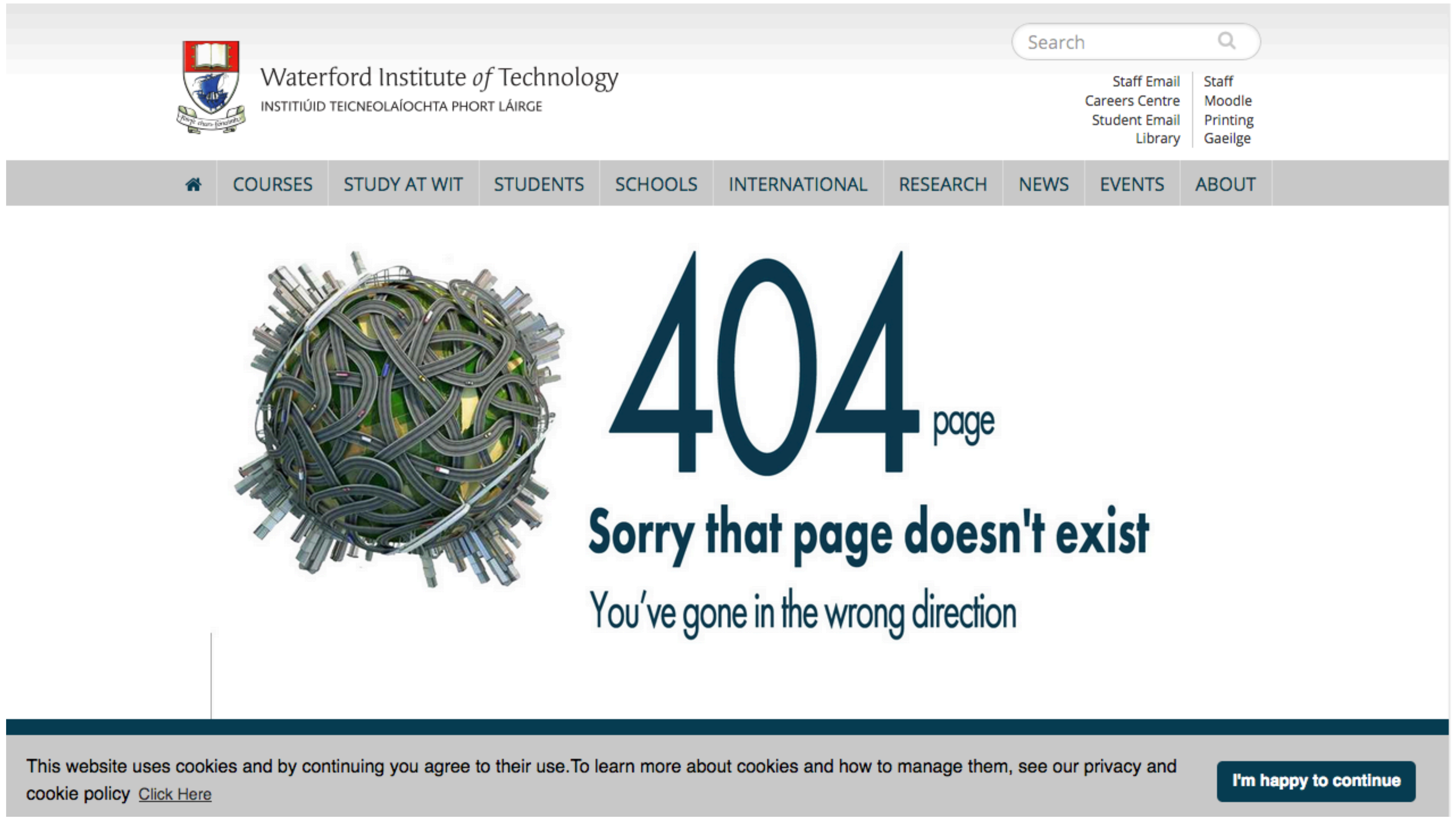
Linking to other pages or images

Creating links to other web pages and to image files can get confusing! If your link to a file or page is incorrect you get:

- Whoops we can't seem to find that page! (404 error)

Or

- No image shows and you have a broken link to an image



404 page

Sorry that page doesn't exist

You've gone in the wrong direction

This website uses cookies and by continuing you agree to their use.To learn more about cookies and how to manage them, see our privacy and cookie policy Click Here

Links: Absolute vs Relative

Absolute

- Complete path to a file on the hard disk: e.g:

  c:/web-development/lab-01/images/xbox.jpg

  c:/web-development/lab-01/index.html

Relative:

./images/xbox.jpg

../apps.html

index.html

You can trace route from “current position” to the destination

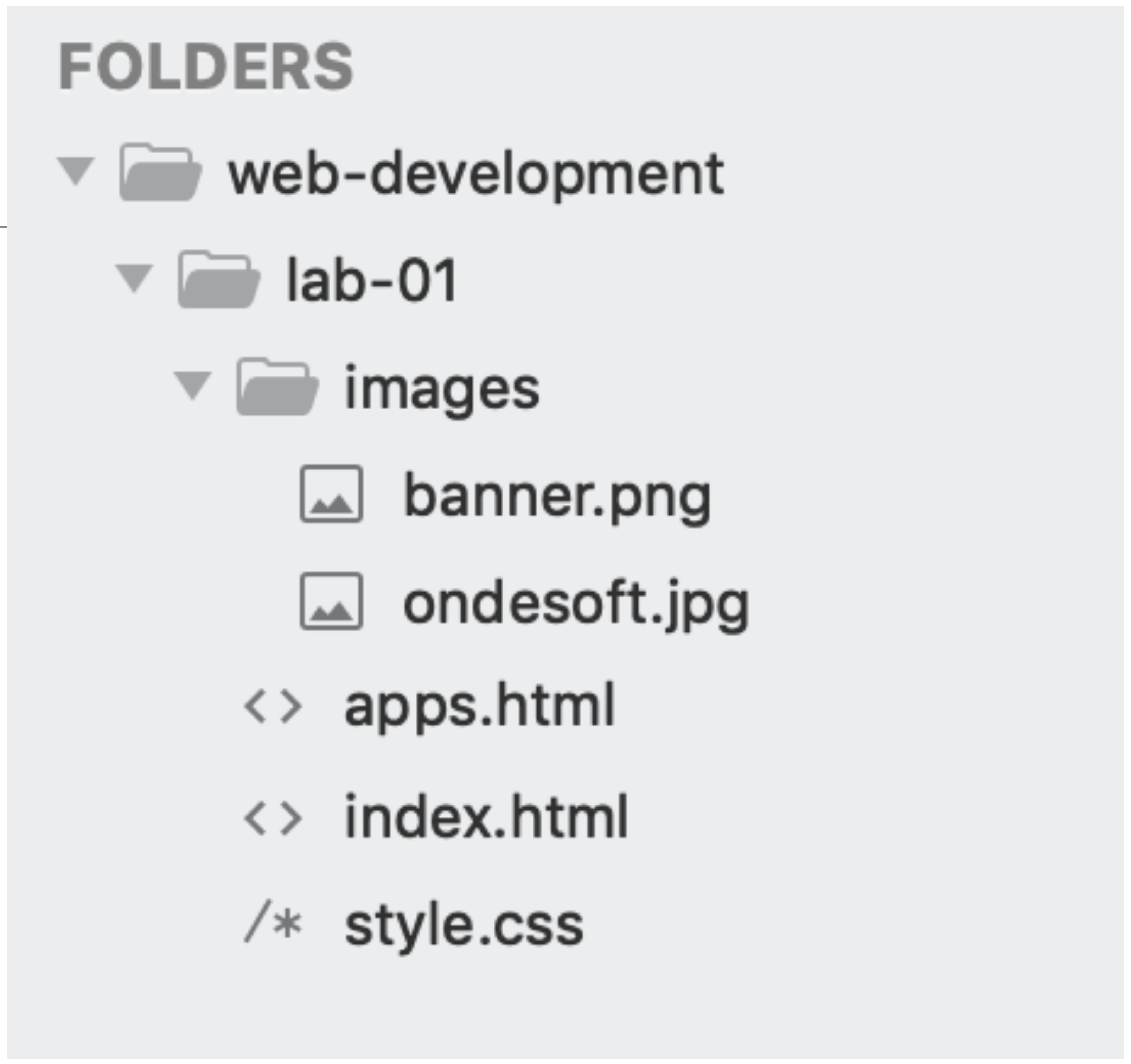
“..” means go up one level

Directory name may prefix filename

Relative Link Examples

- If we are in “lab-01” then “images/banner.png” is a relative link from the current folder (lab-01) to the images folder, and to the file “banner.png” in that folder

- Avoid absolute links!



```
FOLDERS
▼ web-development
  ▼ lab-01
    ▼ images
        banner.png
        ondesoft.jpg
      <> apps.html
      <> index.html
      /* style.css
```

```
<a href=“apps.html”>Movies</a>
```
✓

```
<img src=“./images/banner.jpg”>
```
✓

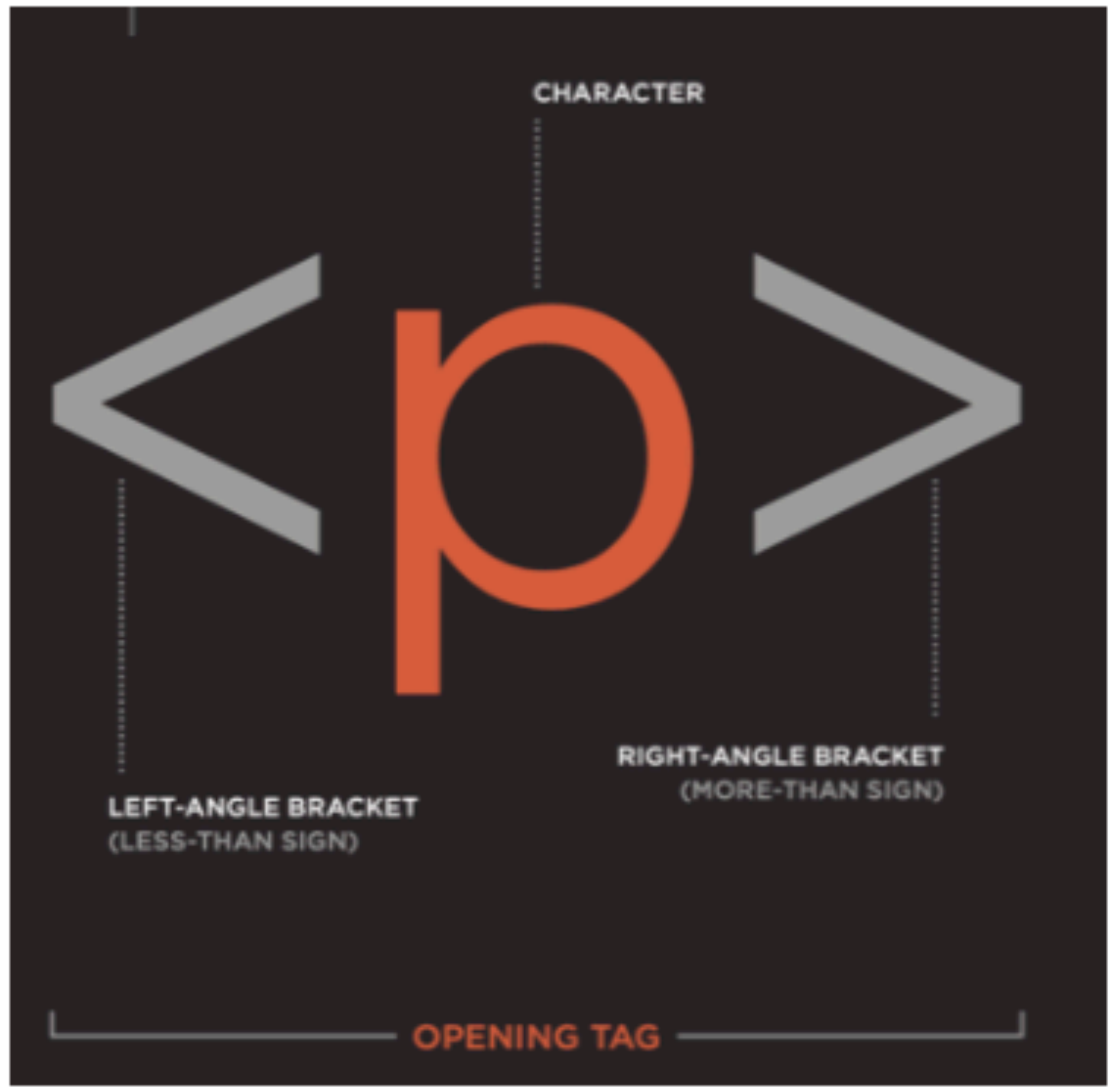
```
<img src=“c:/web-development/lab-01/images/banner.jpg”>
```
✗

- Relative
  - No drive name
  - **../** one level up,
  - **../../** two levels up, etc
  - **somefolder** one level down into folder named someFolder
  - **somefolder/otherfolder** two levels down
  - **../../baffin/data** two levels up and two levels down

Absolute & Relative paths again

- Absolute
  - Drive + {folder(s)} + {file}
  - e.g. C:\projects\baffin\data

HTML : Elements & Linking



Element tags and attributes. The structure of links