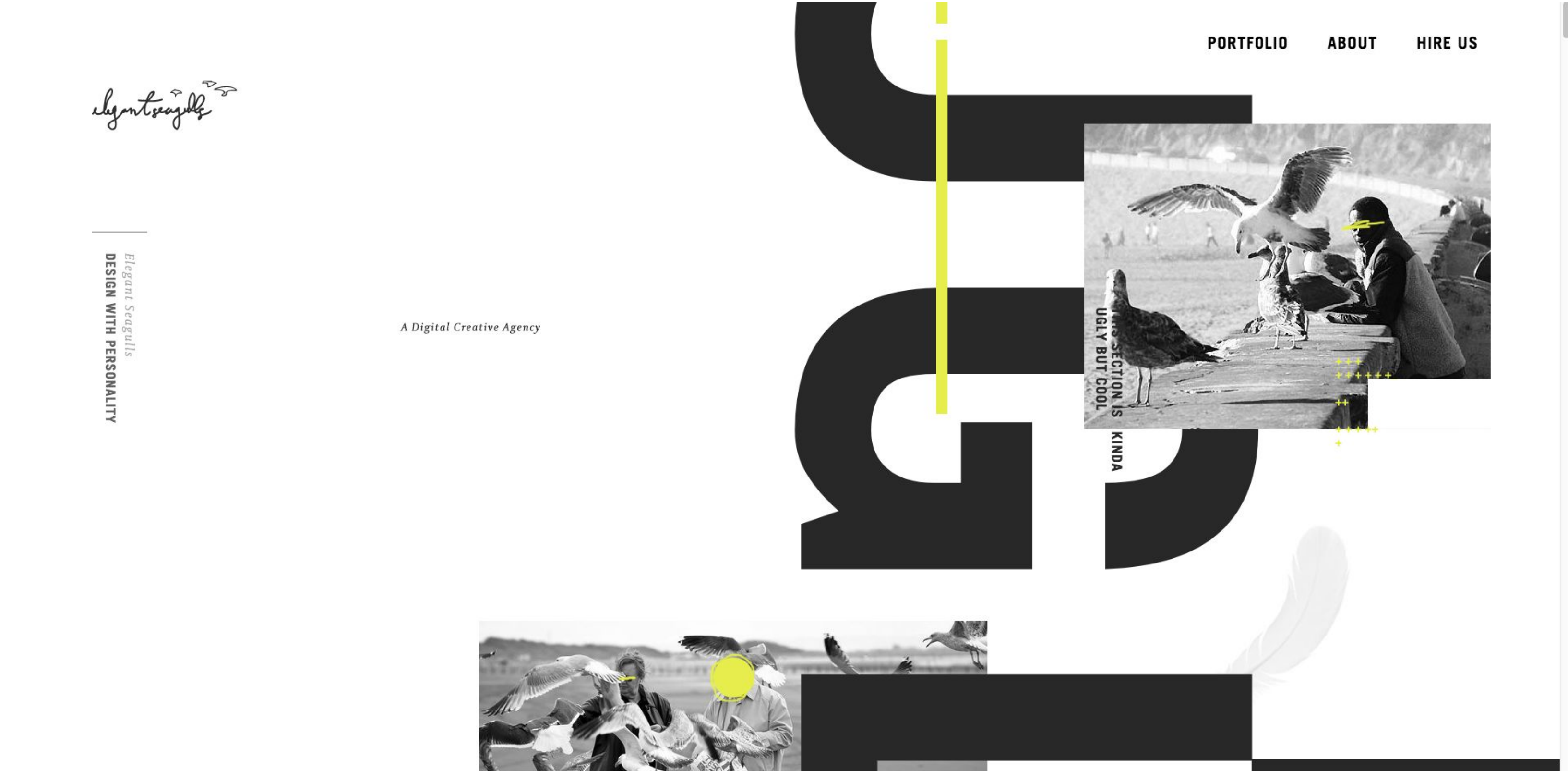
Links and References

Nic Aguirre   j363 Spring 2018

Today

1. Inspiration
2. Quiz
3. Review
4. Lecture: Links and References
5. Practice: Links and References
6. Homework and Readings

Inspiration: www.elegantseagulls.com



Quiz

Canvas > Assignments > Quizzes

Review:
Basic HTML

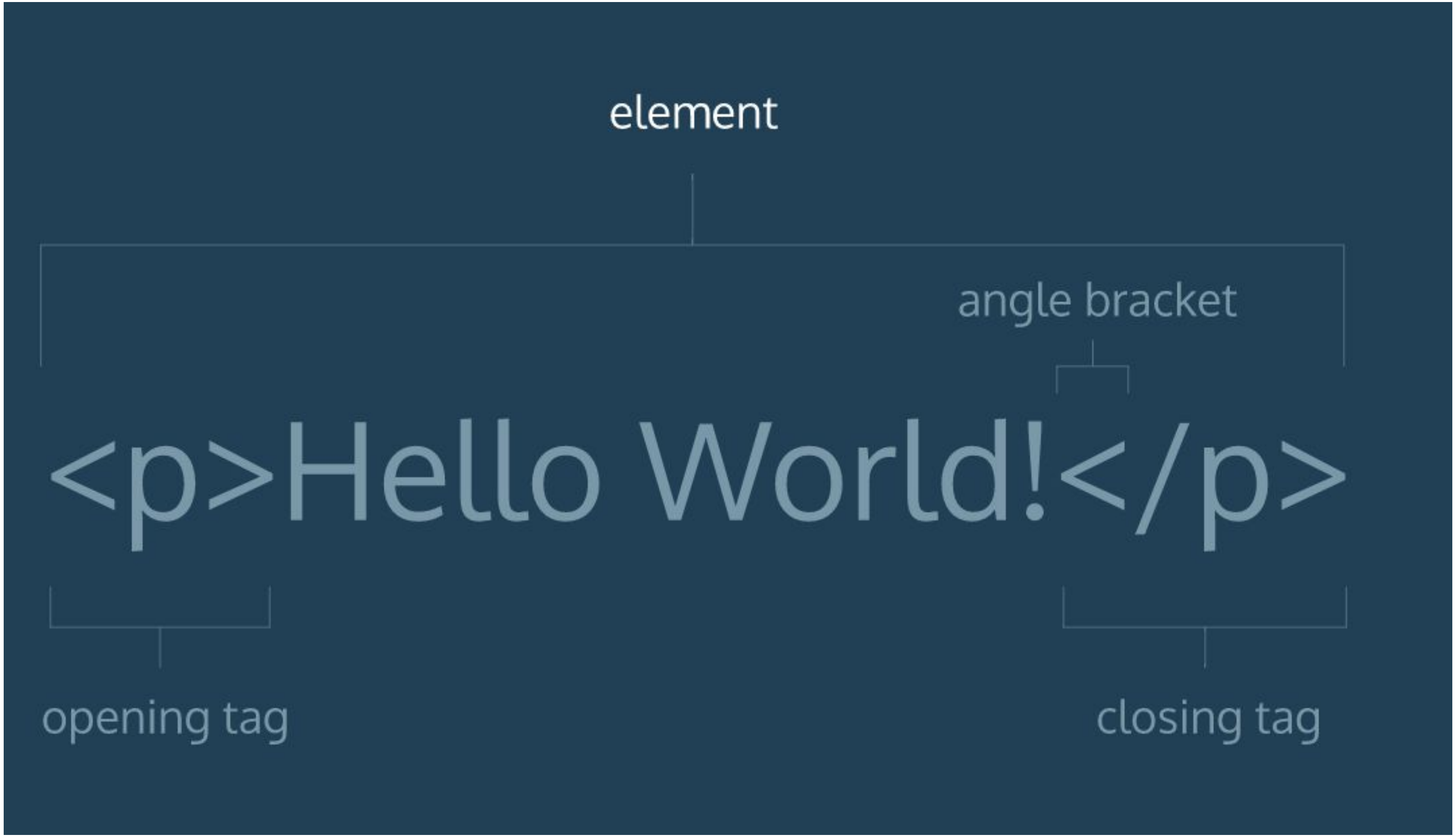
Three main components

Text content - Words, headings, paragraphs

References to other files - images, video, audio, stylesheets

Markup - The HTML elements that describe your text content and make references work

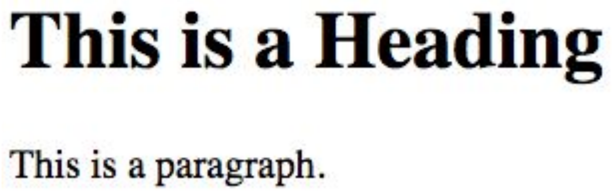
HTML stands for **Hypertext Markup Language**


element
angle bracket
<p>Hello World!</p>
opening tag
closing tag

Thinking in HTML...

```
<body>

     <h1>This is a Heading</h1>

     <p>This is a paragraph.</p>

</body>
```

$\longrightarrow$



This is a Heading

This is a paragraph.

Basic Structure

```html
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Basic Structure

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Tells the browser this is an HTML5 page.

This DOCTYPE declaration should always be the first line in your pages.

Basic Structure

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Notice that tags usually occur **in pairs**.

The *html* tag tells the browser where the page begins and ends.

Basic Structure

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

<head> is loaded before <body>

<head> doesn’t usually handle visible content, it contains information ABOUT the web page.

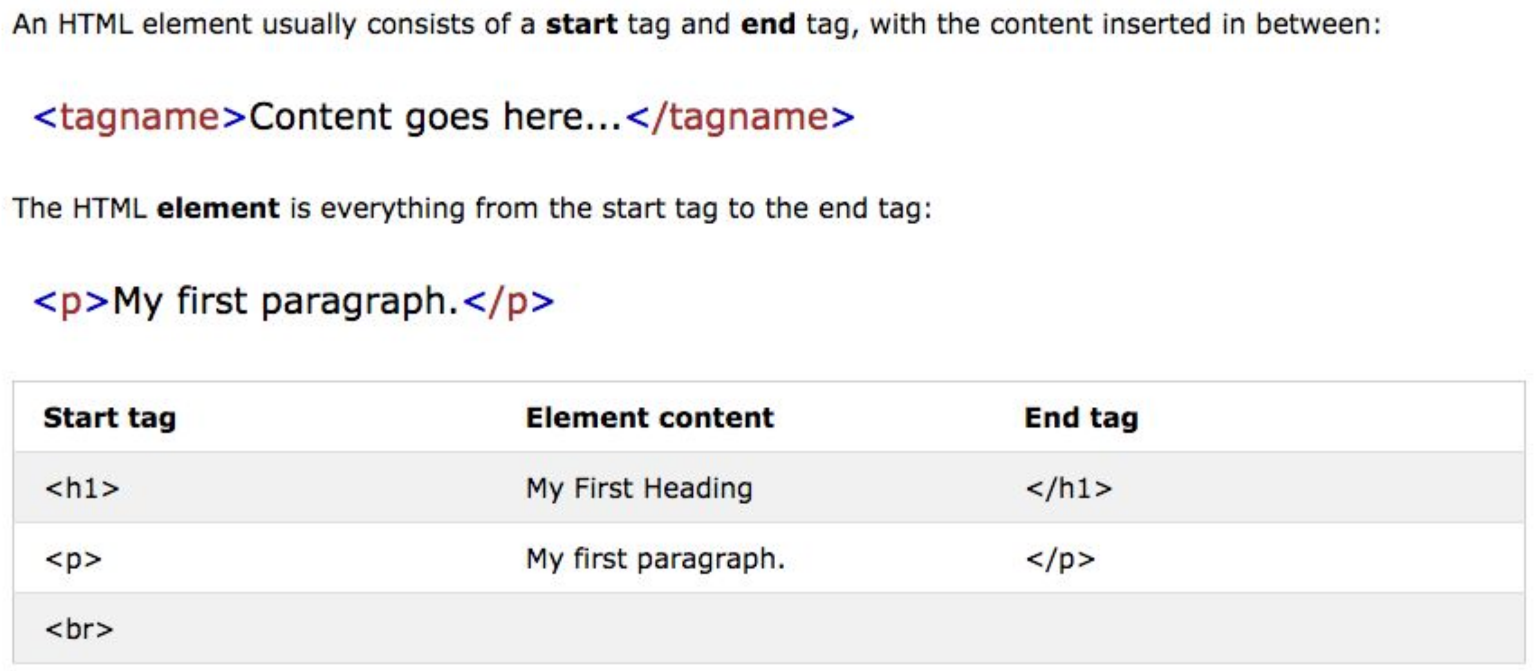
Basic Structure

```html
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Body = page content

Usually, if you can see it on the page, it’s somewhere in the body.

HTML Elements



An HTML element usually consists of a **start** tag and **end** tag, with the content inserted in between:

```
<tagname>Content goes here...</tagname>
```

The HTML **element** is everything from the start tag to the end tag:

```
<p>My first paragraph.</p>
```

Start tag	**Element content**	**End tag**
<h1>	My First Heading	</h1>
<p>	My first paragraph.	</p>
<br>		

Difference between a tag and an element?

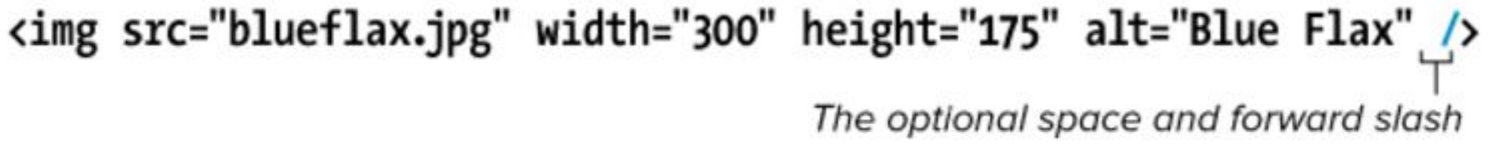
HTML tags are the opening or closing entities. For example:

`<p>` and `</p>` are called HTML tags

HTML element encompasses opening tag, closing tag, content (optional for content-less tags) Ex:

`<p>This is the content</p>` : This complete thing is called a HTML element

Empty elements



```
<img src="blueflax.jpg" width="300" height="175" alt="Blue Flax" />
```

The optional space and forward slash

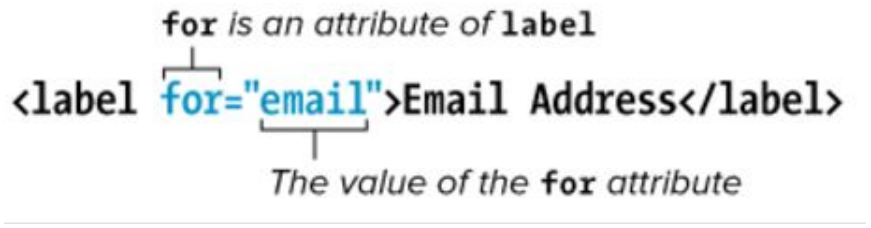
```
<br>
<br/>
```

And

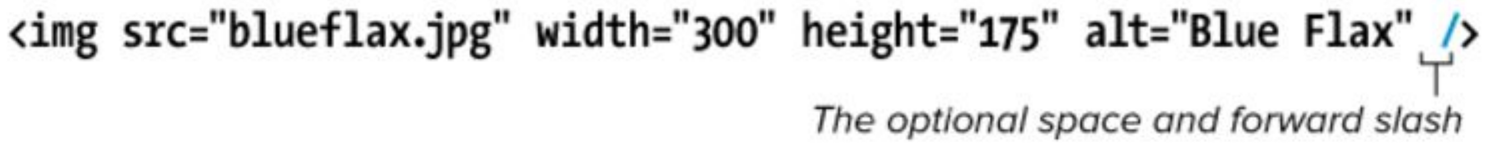
```
<br />
```

Will produce identical results in your browser

Attributes and Values

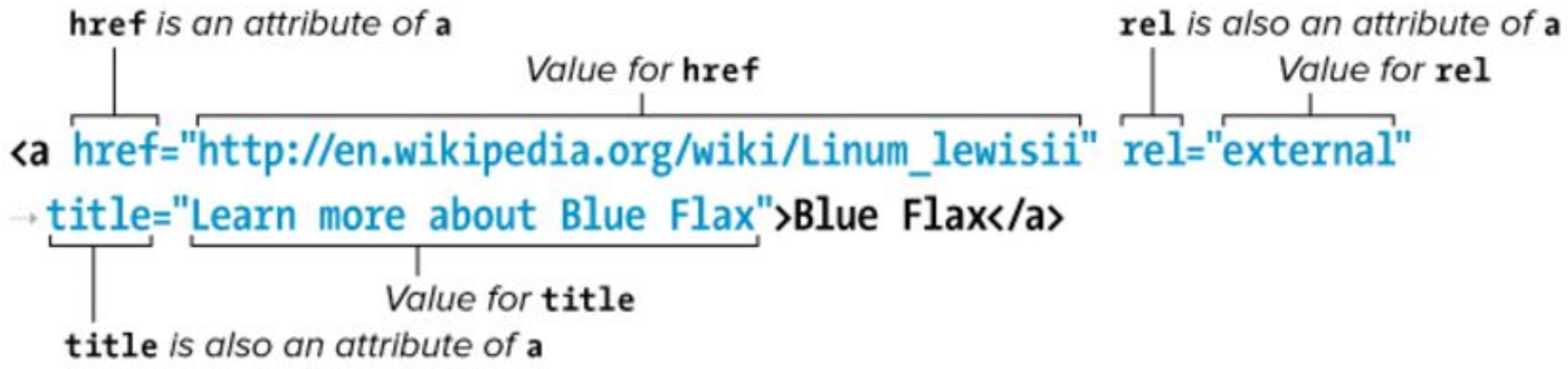

for *is an attribute of* **label**
<label for="email">Email Address</label>
The value of the **for** *attribute*

What attributes do you see?



```
<img src="blueflax.jpg" width="300" height="175" alt="Blue Flax" />
```

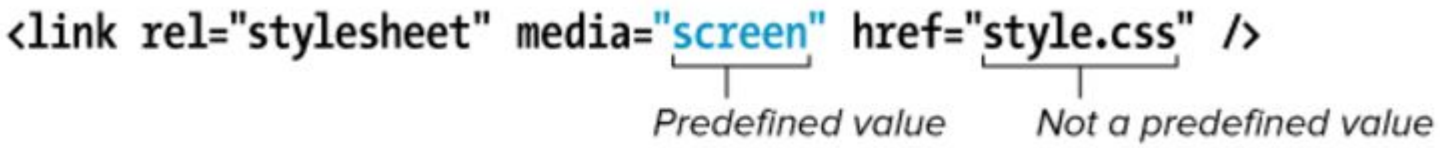
Attribute-value pairs


href is an attribute of a
Value for href
rel is also an attribute of a
Value for rel
Value for title
title is also an attribute of a

```
<a href="http://en.wikipedia.org/wiki/Linum_lewisii" rel="external"
→ title="Learn more about Blue Flax">Blue Flax</a>
```

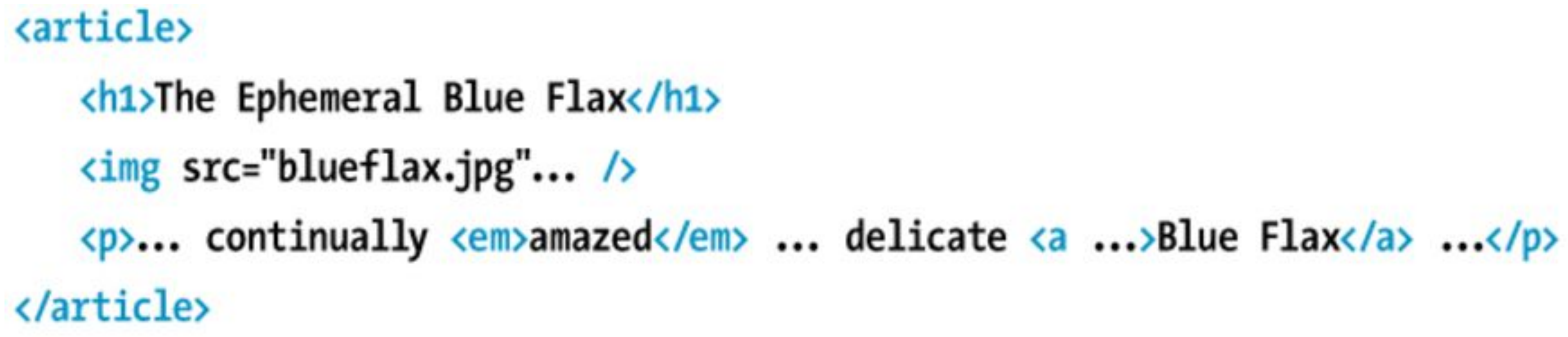
Attribute-value pairs

Some elements only accept specific values.
We call them **predefined** values.



```
<link rel="stylesheet" media="screen" href="style.css" />
```

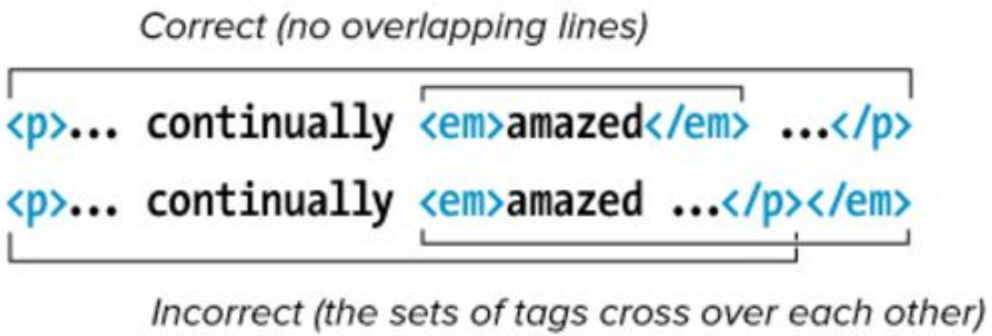
Parent-child relationship



```
<article>
    <h1>The Ephemeral Blue Flax</h1>
    <img src="blueflax.jpg"... />
    <p>... continually <em>amazed</em> ... delicate <a ...>Blue Flax</a> ...</p>
</article>
```

Parent-child relationship

Important to note - if one element contains another, they must be properly **nested**.


Correct (no overlapping lines)
<p>... continually <em>amazed</em> ...</p>
<p>... continually <em>amazed ...</p></em>
Incorrect (the sets of tags cross over each other)

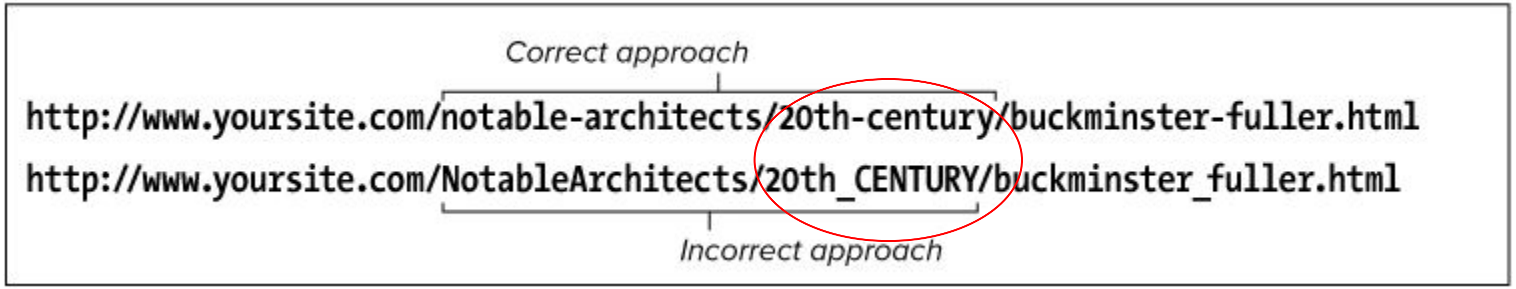
Vocab

1. Tag
2. Element
3. Attribute
4. Value
5. Parent-child

Links and References

File and Folder Names

- Lower case
- No spaces; separate words with a hyphen



Correct approach

```
http://www.yoursite.com/notable-architects/20th-century/buckminster-fuller.html
http://www.yoursite.com/NotableArchitects/20th_CENTURY/buckminster_fuller.html
```

Incorrect approach

File and Folder Names

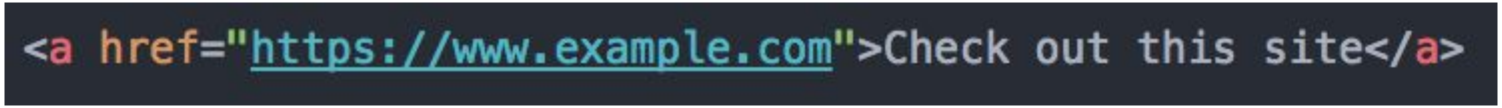
- Lower case
- No spaces; separate words with a hyphen

The <a> tag

Hyperlinks on your website are created with the <a> tag. There are two main parts:

1) Display text - this is the text between the <a> and </a> tags
2) The URL it takes you to. This is defined with the **href** attribute.

The <a> tag

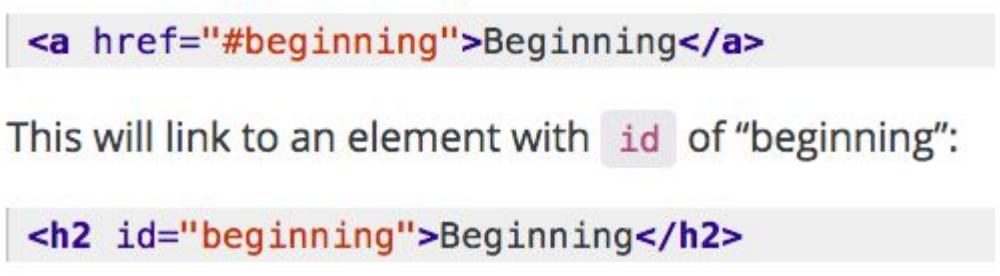


```html
<a href="https://www.example.com">Check out this site</a>
```

1) The text for this link will read ***Check out this site***
2) Clicking the link will take you to **example.com**

The <a> tag

The **href** attribute can also reference an element with a specific id, and it will take you to that element on the page.



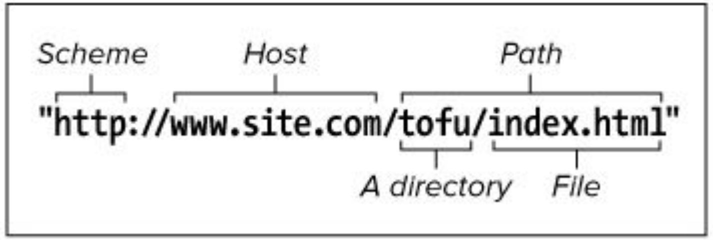
```
<a href="#beginning">Beginning</a>
```

This will link to an element with `id` of “beginning”:

```
<h2 id="beginning">Beginning</h2>
```

URLs

- Scheme is almost always **http** or **https**
- Host is generally the domain name
- Path is the most relevant to you - these are the actual folders and files themselves
- We use the **href** attribute on links, or the **src** attribute on images, to specify a path

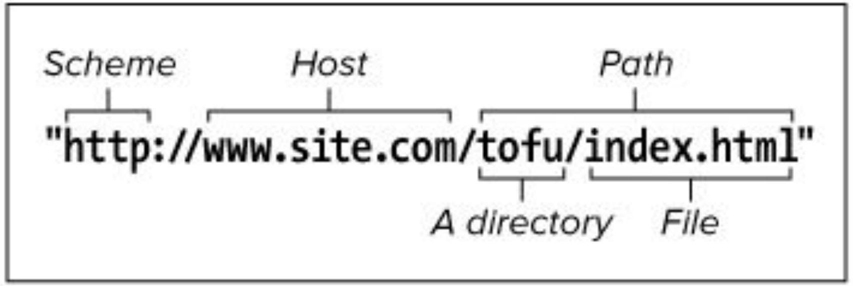


URLs

```
"http://www.site.com/tofu/index.html"
```

A Your basic URL contains a scheme, a host, and a path. The path may contain one or more directory (folder) names and a single file name at the end.

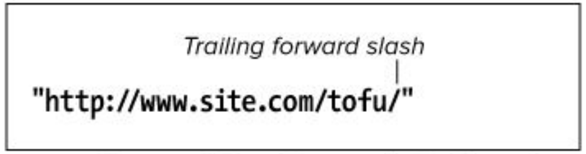
URLs



A Your basic URL contains a scheme, a host, and a path. The path may contain one or more directory (folder) names and a single file name at the end.

URLs

Sometimes, a URL path omits a file name and ends with a directory, which may or may not include a trailing forward slash B. In this case, the URL refers to the default file in the last directory in the path, typically named **`index.html`**. (Virtually all web servers are configured to recognize **`index.html`** as a default file name, so you don't have to change any server settings.)



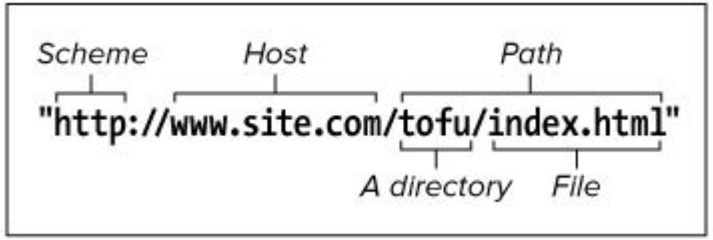
```
"http://www.site.com/tofu/"
```

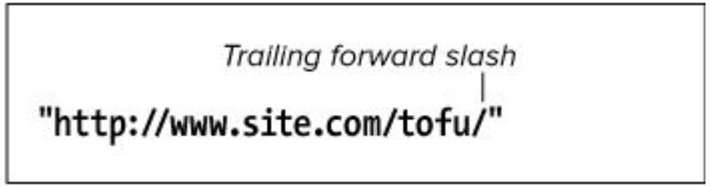
B A URL with a trailing forward slash and no file name points to the default file in the last directory named (in this case, the **`tofu`** directory). The most common default file name is **`index.html`**. So, this URL and the one in the previous example point to the same page.

URLs

When the browser sees a trailing forward slash, it automatically looks for **index.html**

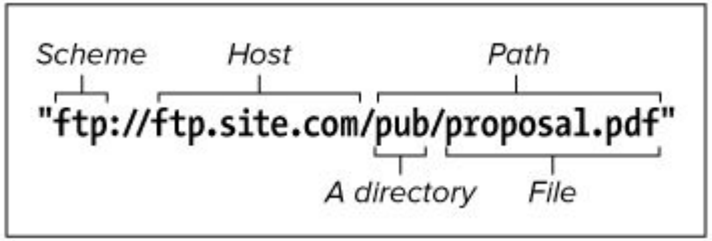
Both of these paths point to the same file!

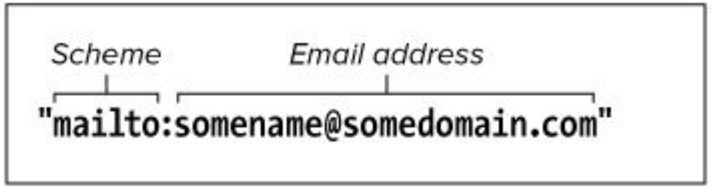




URLs

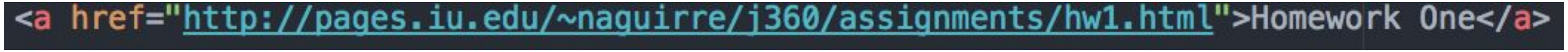
Other, less common protocols include **FTP** (File transfer protocol) and **mailto** (used for e-mail addresses)





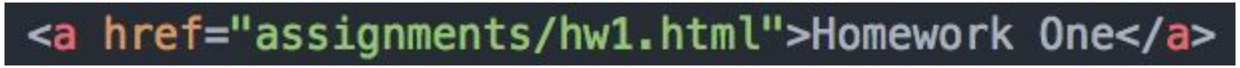
URLs

URLs can be **absolute**



```
<a href="http://pages.iu.edu/~naguirre/j360/assignments/hw1.html">Homework One</a>
```

or **relative**



```
<a href="assignments/hw1.html">Homework One</a>
```

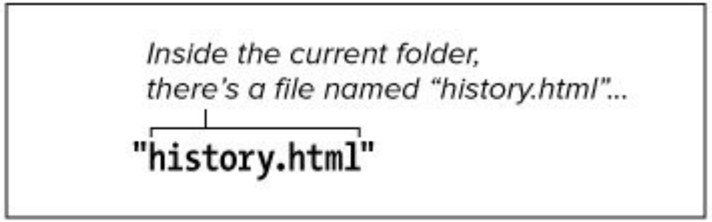
URLs

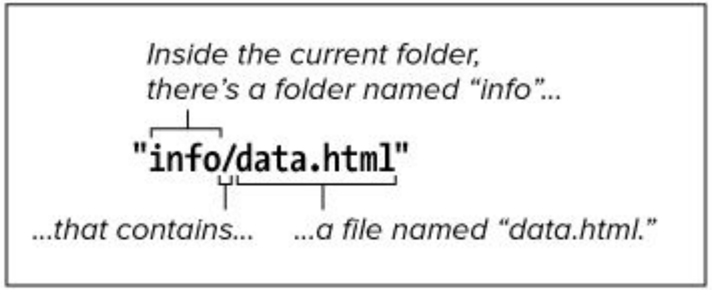
As a general rule…

- If you want to link to a page hosted on a different site (e.g., Wikipedia), you will use an **absolute path**.
- If you want to link to a file, folder, or page on the same site, you will use a **relative path**

URLs

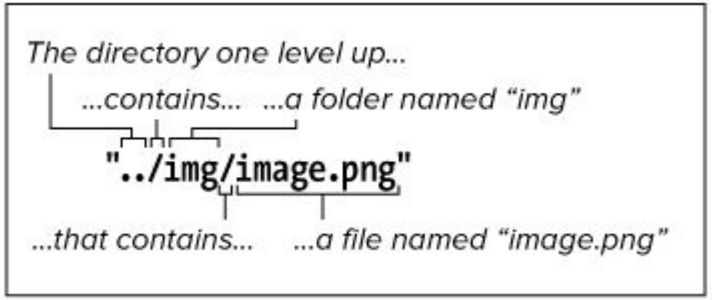
Relative Paths...


Inside the current folder,
there's a file named "history.html"...
"history.html"

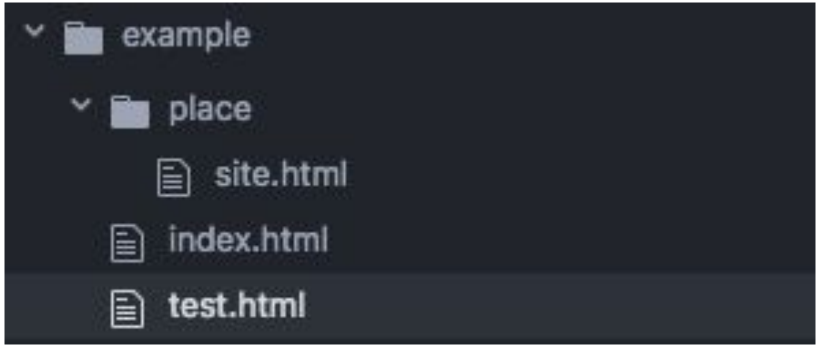

Inside the current folder,
there's a folder named "info"...
"info/data.html"
...that contains...
...a file named "data.html."

URLs

You can reference a file in the parent folder by using two periods to preface the path.



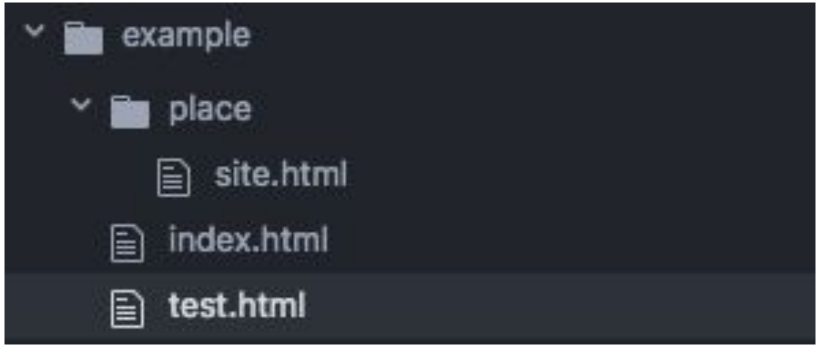
URLs



Suppose you’re in this project, working on **index.html**

If you want to link to site.html, you will use the relative path **“place/site.html”**

URLs



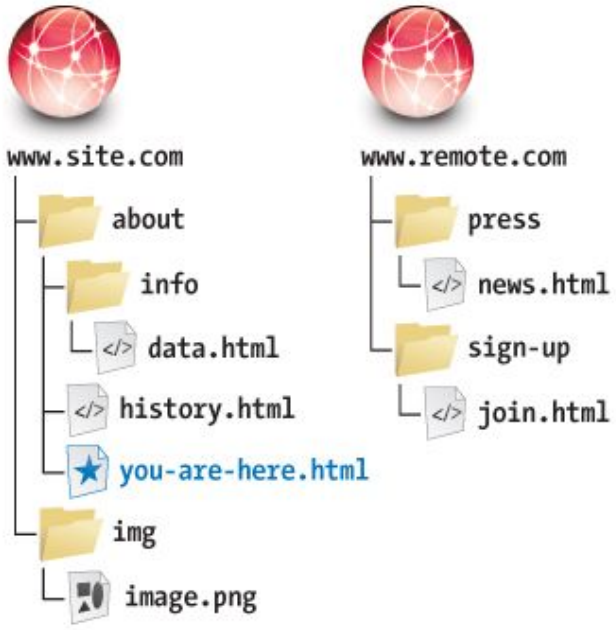
Suppose you're in this project, working on **site.html**

If you want to link to test.html, you will use the relative path **"../test.html"**

If you want to link to index.html, you will use the relative path **"../index.html"**

or

the relative path **"../"**



File name	Absolute URL (can be used anywhere)	Relative URL (only works in you-are-here.html)
history.html	http://www.site.com/about/history.html	history.html
data.html	http://www.site.com/about/info/data.html	info/data.html
image.png	http://www.site.com/img/image.png	../img/image.png
news.html	http://www.remote.com/press/news.html	(none: use absolute)
join.html	http://www.remote.com/sign-up/join.html	(none: use absolute)

Naming elements - class and id

You can name an element with the **class** or **id** attribute. A class can be reused, applying to several elements, while an id is applied to just one element.

We might have several <p> elements but wish to style them differently based on their classes...

```
<p>This is text.</p>

<p class=”green”>This is text.</p>
```

Naming elements - class and id

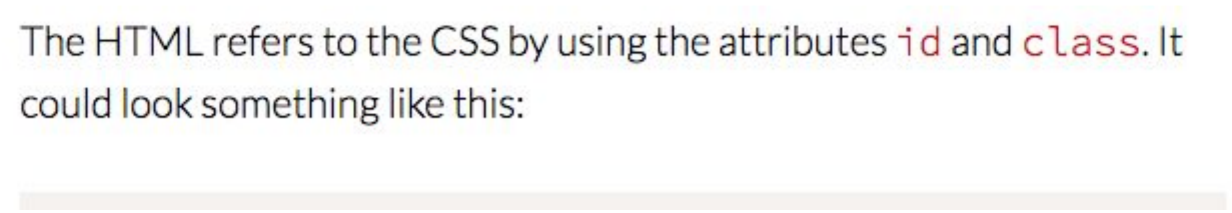
Important to note is that classes and ids don’t “do” anything on their own - we can use CSS to stylize them or use a link’s **href** attribute to refer to an element by id

Classes and ids are mainly used for:

- Giving semantic meaning to an element (*e.g.*, “*container*”)
- Styling elements with CSS
- Linking to different parts of the same page

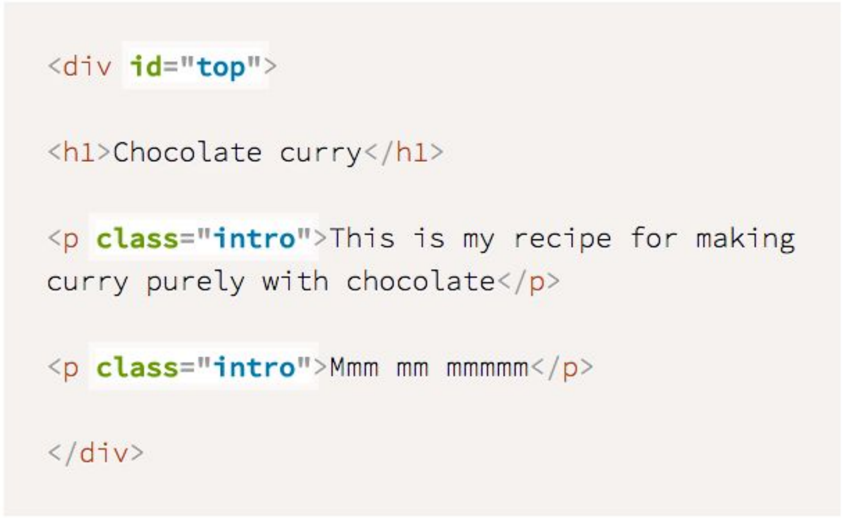
In the CSS, a class selector is a name preceded by a **full stop** (“.”) and an ID selector is a name preceded by a **hash character** (“#”).

So the CSS might look something like:



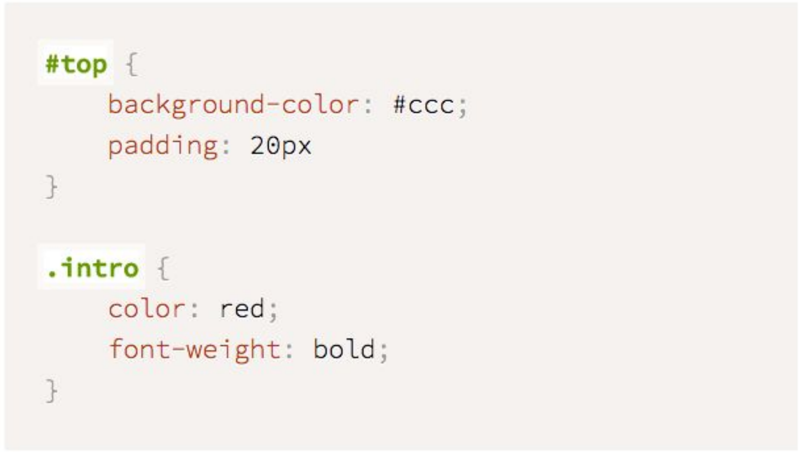
The HTML refers to the CSS by using the attributes `id` and `class`. It could look something like this:

HTML



```html
<div id="top">

<h1>Chocolate curry</h1>

<p class="intro">This is my recipe for making
curry purely with chocolate</p>

<p class="intro">Mmm mm mmmmm</p>

</div>
```

CSS



```css
#top {
    background-color: #ccc;
    padding: 20px
}

.intro {
    color: red;
    font-weight: bold;
}
```

Practice

Homework