

# Web Development

---

Produced  
by

Eamonn de Leastar ([edelestar@wit.ie](mailto:edelestar@wit.ie))

Department of Computing, Maths & Physics  
Waterford Institute of Technology

<http://www.wit.ie>

<http://elearning.wit.ie>

# HTML Page Structure

---

# Learning Outcomes

---

- Understand the structure of an **HTML Element**, and be able to recognize its variants.
- Be able to read and compose a **relative path**, and be able to distinguish it from an **absolute path**.
- Understand the implications of **nesting** of elements, and in particular be able to distinguish between correct and incorrect nesting.
- Be able to differentiate between **block** and **inline** elements
- Understand the general context of **wireframing**

# Agenda

---

- Elements, Attributes, & Documents
- Linking
- Nesting
- Linebreak, Block & Inline Elements
- Structuring a Page / Wireframing

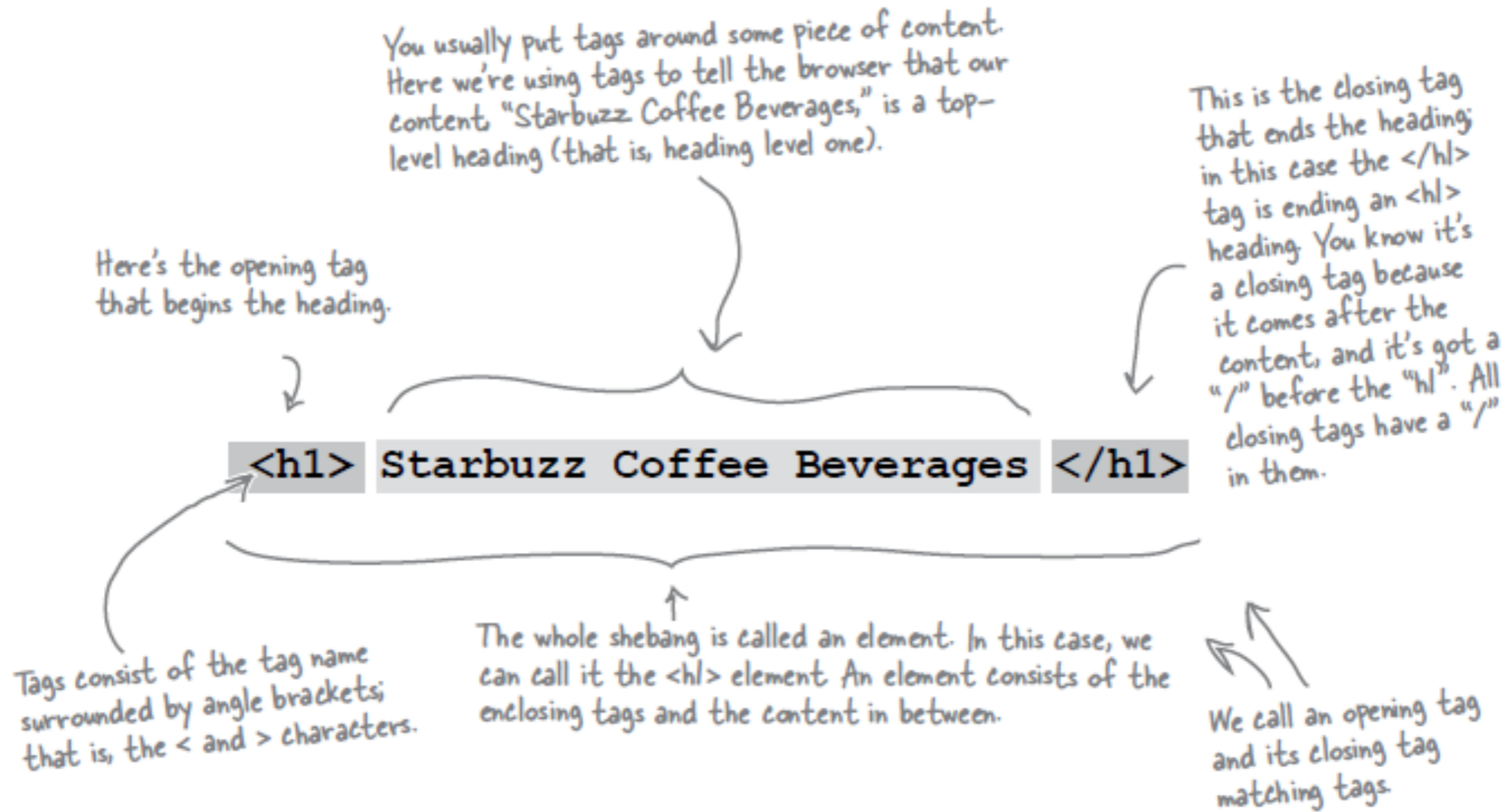
# Agenda

---

- Elements, Attributes, & Documents
- Linking
- Nesting
- Linebreak, Block & Inline Elements
- Structuring a Page / Wireframing

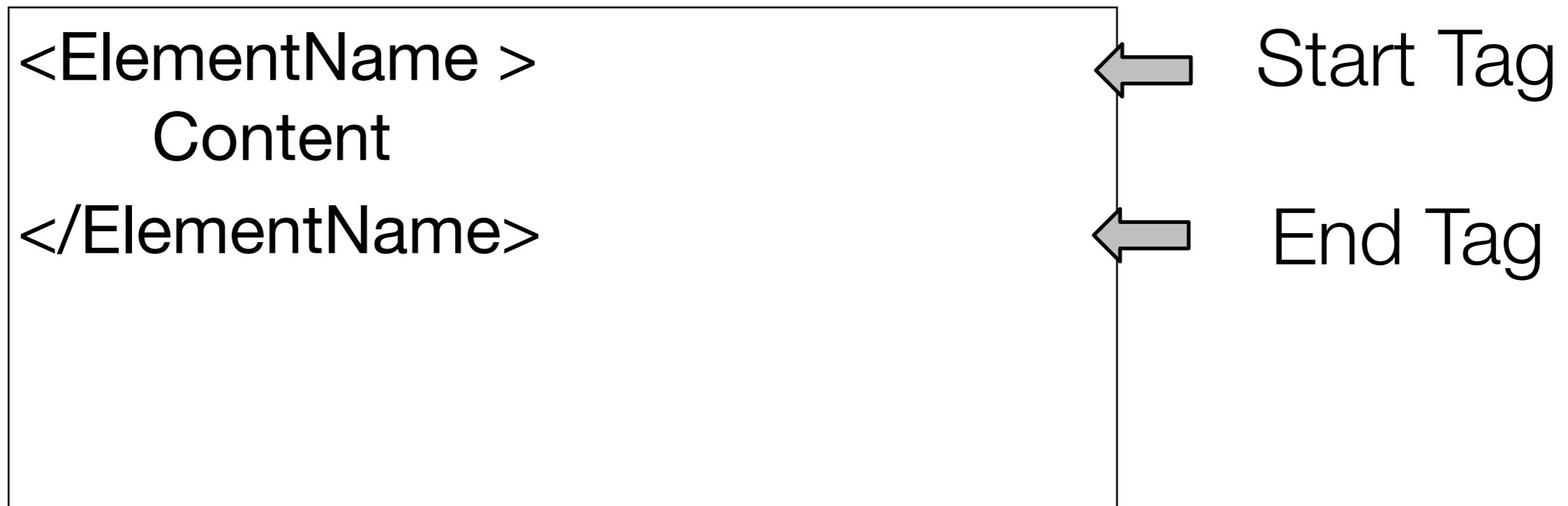
# Components of an HTML Element

---



# Components of an HTML Element

---



<title>

---

<title>My App Store</title>

**ElementName:** *<title>*  
**Content:** *My App Store*  
**ElementName:** *</title>*



<p>

---

<p> 600 million YouTube views later, Dude Perfect is back with their most epic game yet! Go BIGGER than ever hitting mind-blowing trick shots through tons of crazy levels!</p>

**ElementName:** <p>

**Content:** 600 million YouTube views later, Dude Perfect is back with their most epic game yet! Go BIGGER than ever hitting mind-blowing trick shots through tons of crazy levels!

**ElementName:** </p>

# 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>
```

Handwritten annotations with arrows pointing to the code: "attribute name" points to `href`, "equals sign" points to `=`, "double quote" points to the opening quote, "attribute value" points to `top10.html`, and "double quote" points to the closing quote.

### Not this

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

Handwritten note below the code: "No double quotes around the attribute value"

<a>

---

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

<b>ElementName:</b>	<a>
<b>AttributeName:</b>	href
<b>AttributeValue:</b>	"apps.html"
<b>Content:</b>	<i>App Store</i>
<b>ElementName:</b>	</a>

<img>

---

```

```

<b>ElementName:</b>	<i>&lt;img&gt;</i>
<b>AttributeName:</b>	<i>src</i>
<b>AttributeValue:</b>	<i>"../../images/delete.jpg"</i>
<b>Content:</b>	<i>empty</i>
<b>ElementName:</b>	<i>none (for the moment)</i>

# Attributes

---

there are no  
**Dumb Questions**

**Q:** Can I just make up new attributes for an HTML element?

**A:** Web browsers only know about a predefined set of attributes for each element. If you just made up attributes, then browsers wouldn't know what to do with them, and as you'll see later in the book, doing this will very likely get you into trouble. When a browser recognizes an element or an attribute, we like to say that it "supports" that element or attribute. You should only use attributes that you know are supported.

# 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>Strike I</li>
      <li>Crash Landing</li>
    </ul>
    <h2>Recommended Apps</h2>
    <ul>
      <li>Chop</li>
      <li>XBox mania</li>
    </ul>
  </body>
</html>
```

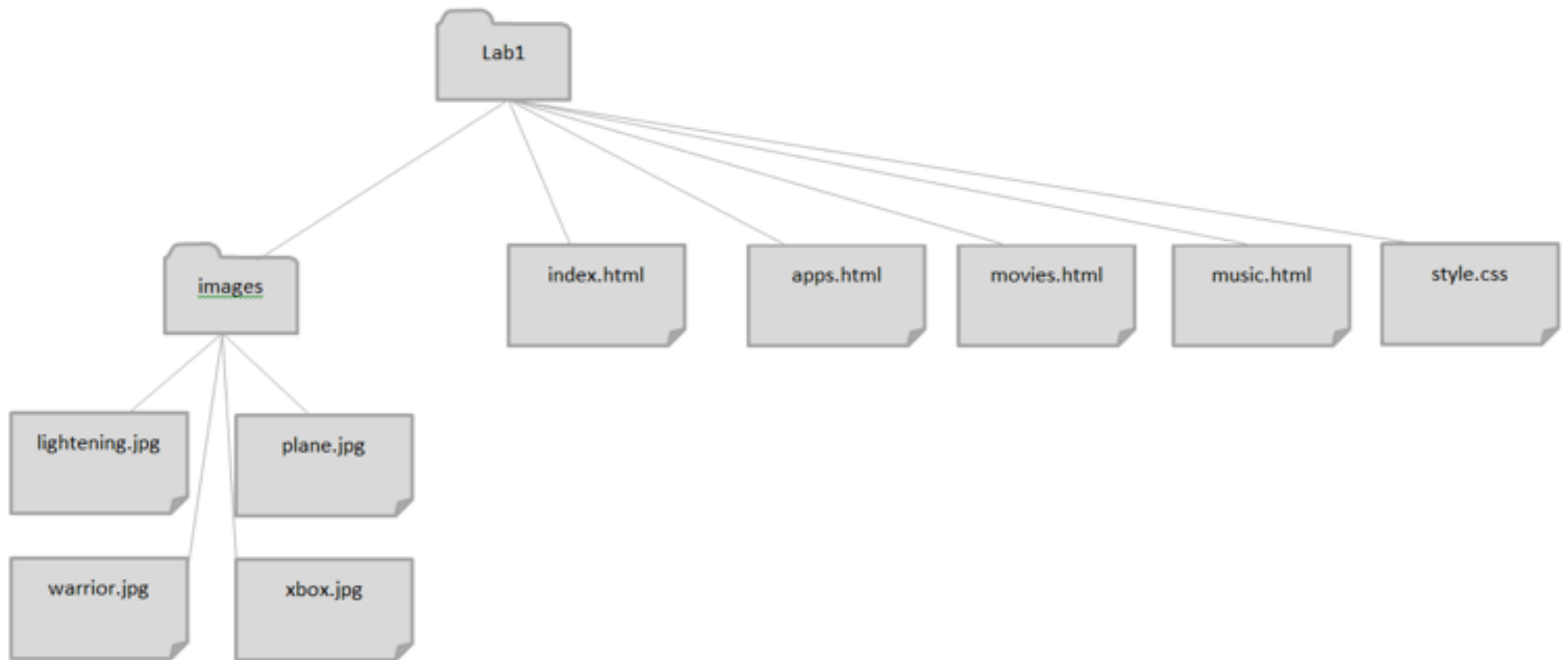
# Agenda

---

- Elements, Attributes, & Documents
- Linking
- Nesting
- Linebreak, Block & Inline Elements
- Structuring a Page / Wireframing

# Linking

---





# 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

# Links: Absolute vs Relative

---

## Absolute

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

C:/My Documents/webdevelopment/lab01/images/xbox.jpg

C:/My Documents/webdevelopment/lab01/movies.html

- Relative:

./images/xbox.jpg

../apps.html

index.html

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 “lab1” then “images/plane.png” is a relative link from the current folder (Lab1) to the images folder, and to the file “plane.png” in that folder
- If we are in the images folder, and we wish to link to one of the web pages, the link “../movies” means “go up one level, and then find “movies.html”
- Avoid absolute links!



```
<a href="movies.html">Movies</a>
```

```

```

# Agenda

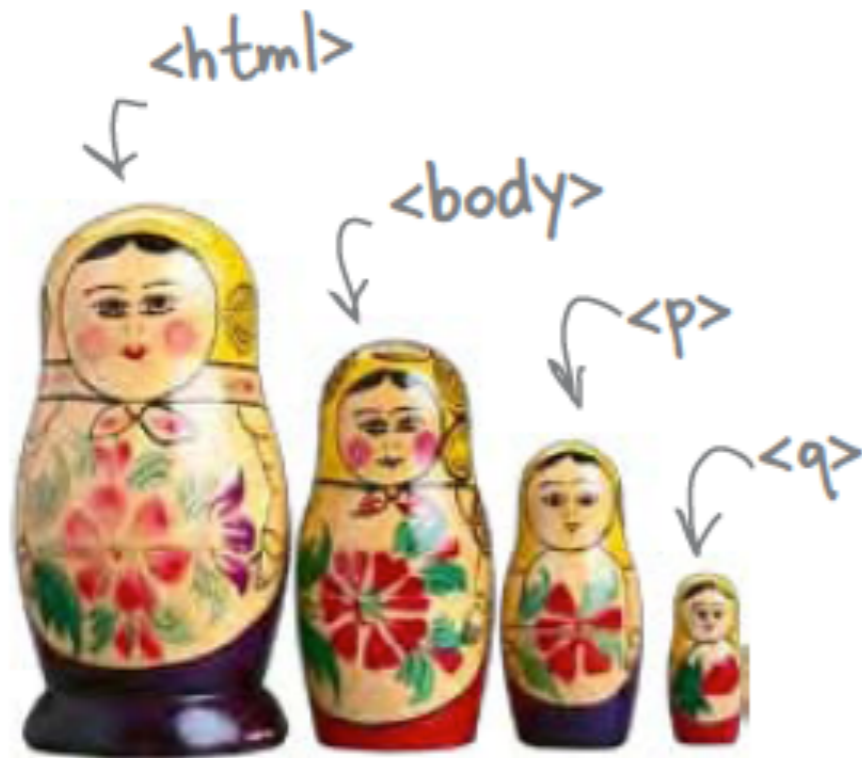
---

- Elements, Attributes, & Documents
- Linking
- Nesting
- Linebreak, Block & Inline Elements
- Structuring a Page / Wireframing

# Nesting

---

- When we put one element inside another element, we call that nesting.
- We say, the `<p>` element is nested inside the `<body>` element.
- We put a `<body>` element inside an `<html>` element, a `<p>` element inside a `<body>` element etc.



# Nesting - Tree Structure

```
<html>
  <head>
    <title>Musings</title>
  </head>
  <body>
    <p>
      To quote Buckaroo,
      <q>The only reason
        for time is so
        that everything
        doesn't happen
        at once.</q>
    </p>
  </body>
</html>
```

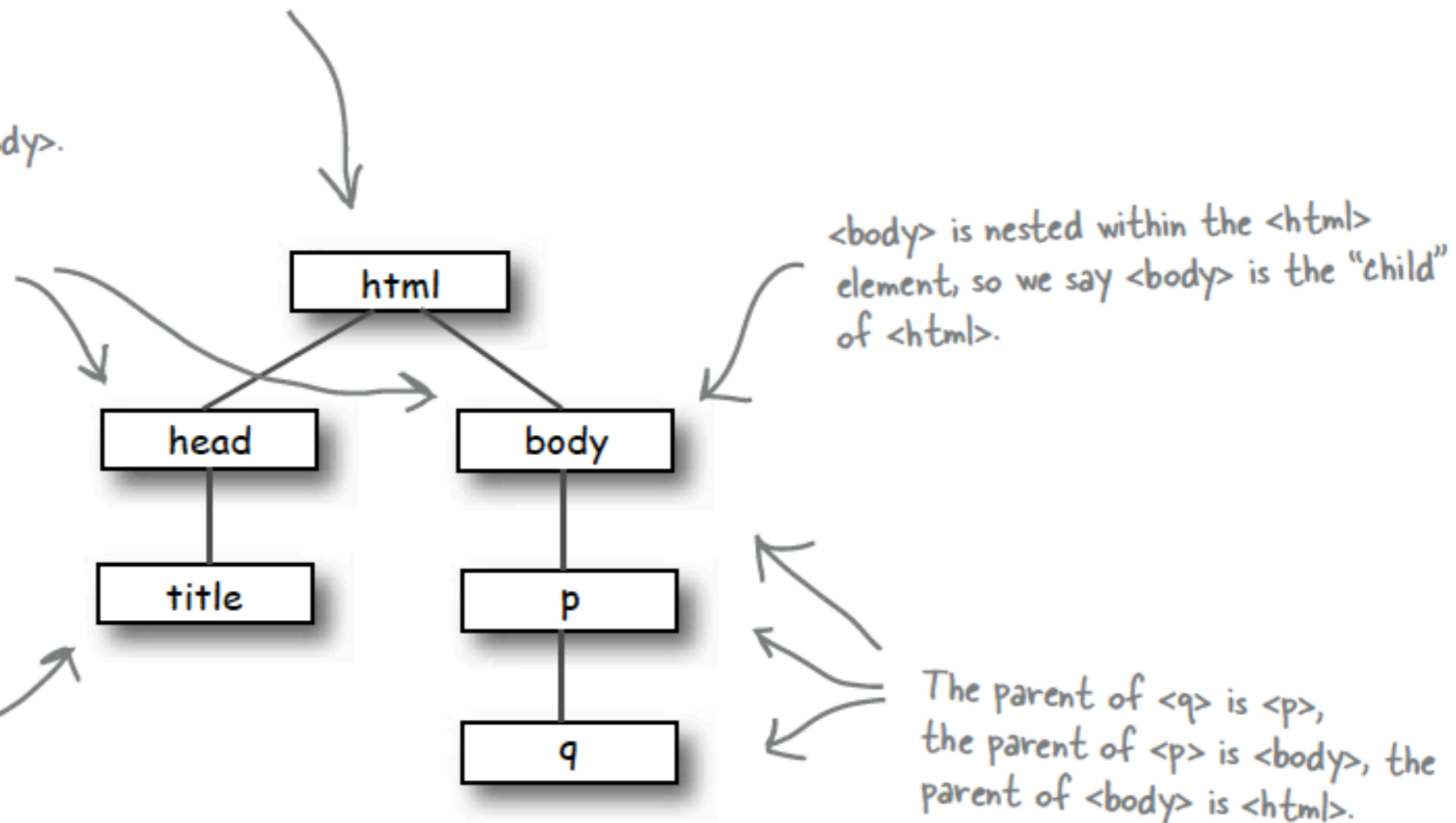
`<html>` is always the element at the root of the tree.

`<html>` has two nested elements: `<head>` and `<body>`. You can call them both "children" of `<html>`.

`<body>` is nested within the `<html>` element, so we say `<body>` is the "child" of `<html>`.

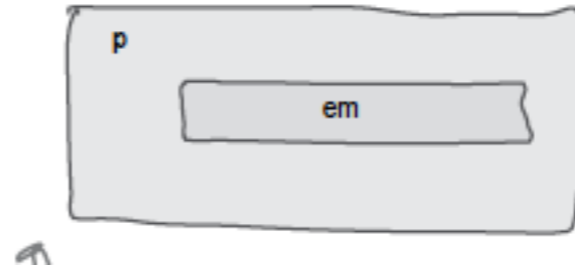
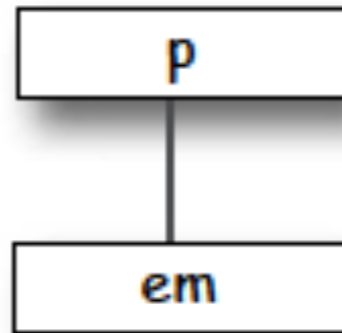
`<title>` is nested within the `<head>` element.

The parent of `<q>` is `<p>`, the parent of `<p>` is `<body>`, the parent of `<body>` is `<html>`.



# Nesting can be Incorrect!

```
<p>I'm so going to blog <em>this</em></p>
```



Good

```
<p>I'm so going to blog <em>this</p></em>
```

?



Bad

# Agenda

---

- Elements, Attributes, & Documents
- Linking
- Nesting
- Linebreak, Block & Inline Elements
- Structuring a Page / Wireframing



# Line breaks

---

there are no  
**Dumb Questions**

**Q:** I think I know what a linebreak is; it's like hitting the carriage return on a typewriter or the Return key on a computer keyboard. Right?

**A:** Pretty much. A linebreak is literally a "break in the line," like this, and happens when you hit the Return key, or on some computers, the Enter key. You already know that linebreaks in HTML files don't show up visually when the browser displays a page, right? But now you've also seen that anytime you use a block element, the browser uses linebreaks to separate each "block."



# Line Breaks

---

Here's the July 14th snippet from Tony's page. →

```
<h2>July 14, 2012</h2>
<p>
  I saw some Burma Shave style signs on the
  side of the road today:
</p>
<blockquote>
  Passing cars, <br>
  When you can't see, <br>
  May get you, <br>
  A glimpse, <br>
  Of eternity. <br>
</blockquote>
<p>
  I definitely won't be passing any cars.
</p>
```

← Add a <br> element to any line when you want to break the flow and insert a "linebreak."

# Line Breaks

---

...lla, W...agic City, ...Bouthern... East  
Chance, CO, Why, AZ and Truth or Consequences, NM.

**July 14, 2012**

I saw some Burma Shave style signs on the side of the road today:

Passing cars,  
When you can't see,  
May get you,  
A glimpse,  
Of eternity.

I definitely won't be passing any cars.

**June 2, 2012**

Each line now has a  
linebreak after it.



# Line Breaks

---

- Break: `<br> </br>`
- An Empty element: - sometimes shortened to:
  - `<br></br>`
- or just
  - `<br>`
- or
  - `<br />`

# Block vs Inline Elements

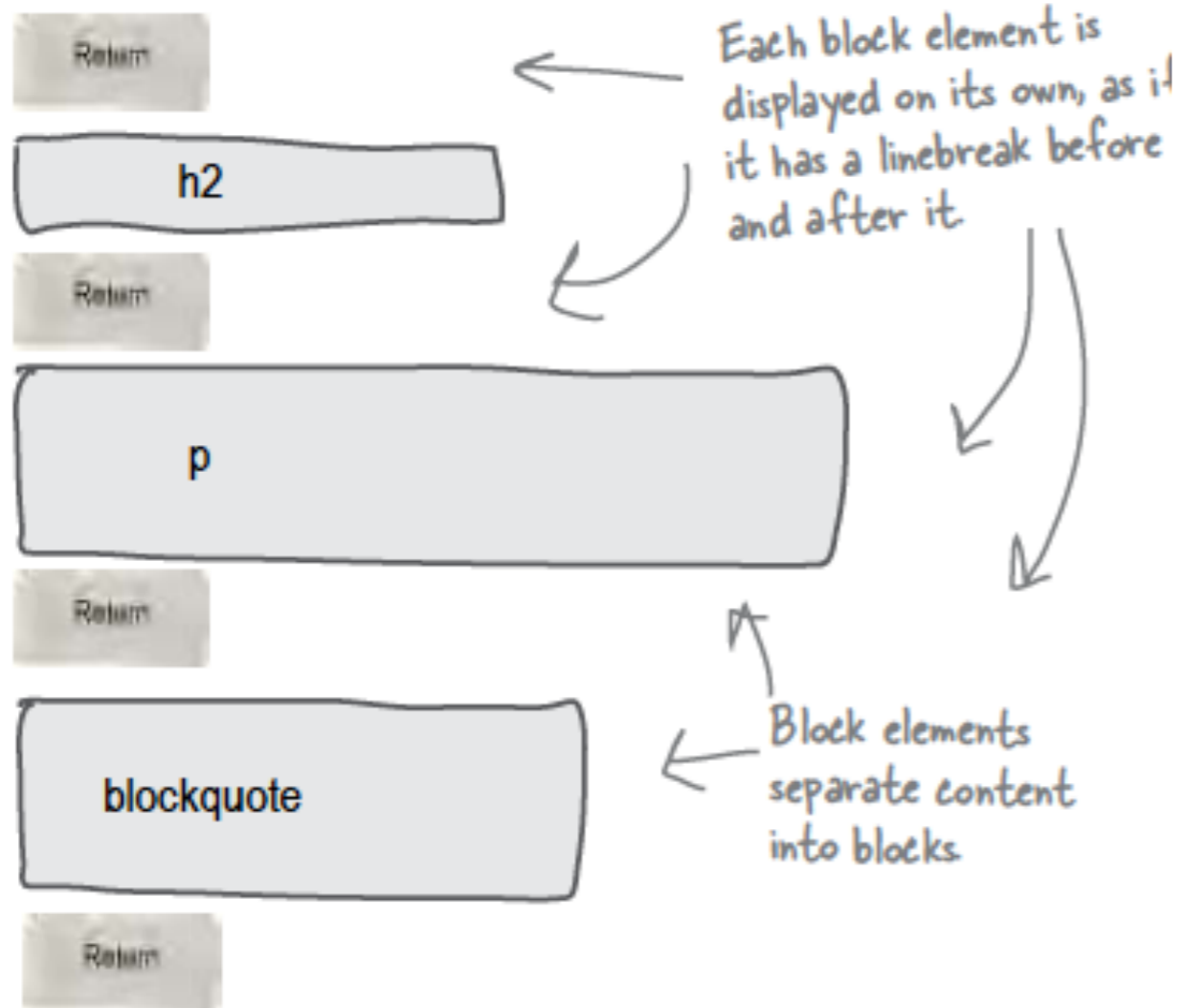
---

- Block elements are always displayed as if they have a line break before and after them
- inline elements appear “in line” within the flow of the text in your page.

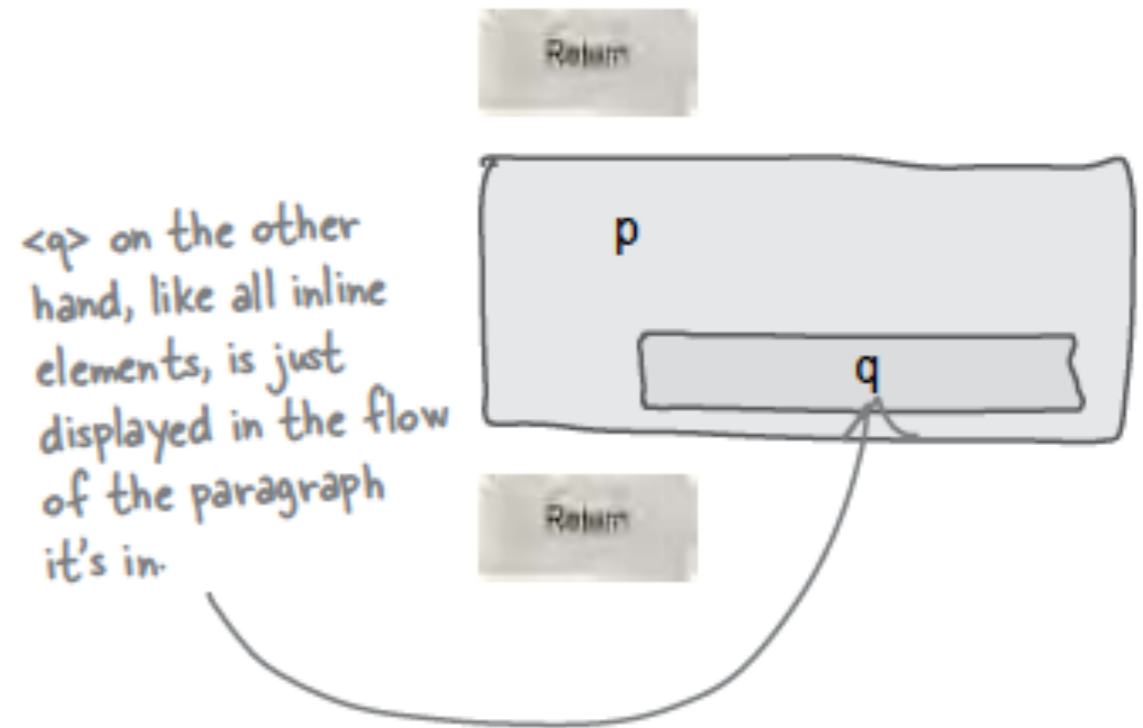
“Block elements stand on their own; inline elements go with the flow.”

# Examples

- Block - h1, h2, p, blockquote



- Inline - a, em, q



# Agenda

---

- Elements, Attributes, & Documents
- Linking
- Nesting
- Linebreak, Block & Inline Elements
- Structuring a Page / Wireframing

# Planning a Document (1)

- Visualizing End Result

## Segway'n USA

Documenting my trip around the US on my very own Segway!

August 20, 2012



Well I made it 1200 miles already, and I passed through some interesting places on the way: Walla Walla, WA, Magic City, ID, Bountiful, UT, Last Chance, CO, Why, AZ and Truth or Consequences, NM.

July 14, 2012

I saw some Burma Shave style signs on the side of the road today: "Passing cars, When you can't see, May get you, A glimpse, Of eternity." I definitely won't be passing any cars

June 2, 2012



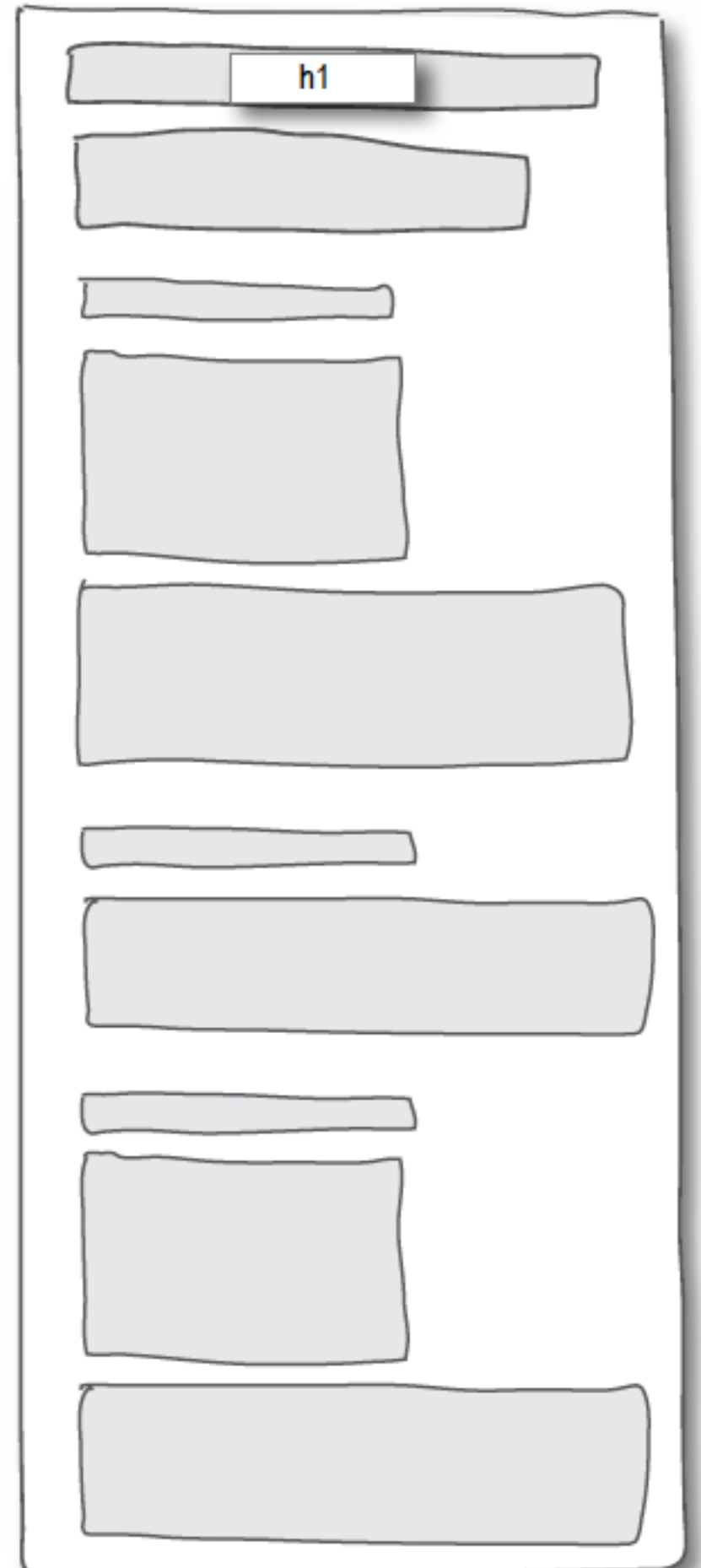
My first day of the trip! I can't believe finally got everything packed and ready to go. Because I'm on a Segway, I wasn't able to bring a whole lot with me: cell phone, iPod, digital camera, and a protein bar. Just the essentials. As Lao Tzu would have said, "A journey of a thousand miles begins with one Segway."



# Planning a Document (2)

---

- Identify Block Structure



# Planning a Document (3)

---

- Building blocks
  - Don't forget `<html>` `<head>` and `<body>`
- Translate to html
  - Build content

Don't forget, you always need the <html>, <head>, <title>, and <body> elements.

We're using the title of the journal as the title of the web page.

```
<html>
<head>
  <title>My Trip Around the USA on a Segway</title>
</head>
<body>

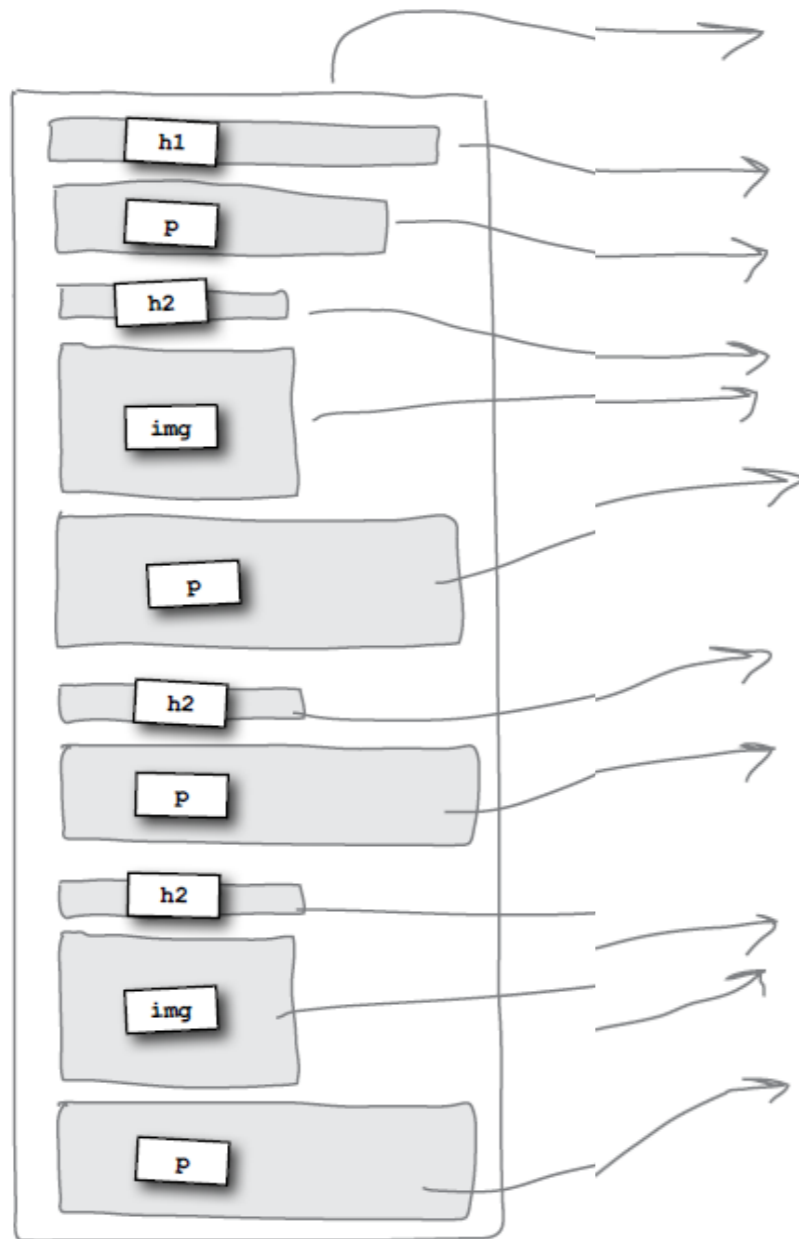
  <h1>Segway'n USA</h1>
  <p>
    Documenting my trip around the US on my very own Segway!
  </p>

  <h2>August 20, 2012</h2>
  
  <p>
    Well I made it 1200 miles already, and I passed
    through some interesting places on the way: Walla Walla,
    WA, Magic City, ID, Bountiful, UT, Last Chance, CO,
    Why, AZ and Truth or Consequences, NM.
  </p>

  <h2>July 14, 2012</h2>
  <p>
    I saw some Burma Shave style signs on the side of the
    road today: "Passing cars, When you can't see, May get
    you, A glimpse, Of eternity." I definitely won't be passing
    any cars.
  </p>

  <h2>June 2, 2012</h2>
  
  <p>
    My first day of the trip! I can't believe I finally got
    everything packed and ready to go. Because I'm on a Segway,
    I wasn't able to bring a whole lot with me: cell phone, iPod,
    digital camera, and a protein bar. Just the essentials. As
    Lao Tzu would have said, "A journey of a thousand miles begins
    with one Segway."
  </p>

</body>
</html>
```



Here's the heading and description of Tony's journal.

heading  
image  
description

Here's Tony's most recent entry.

Here's his second entry, which doesn't have an image.

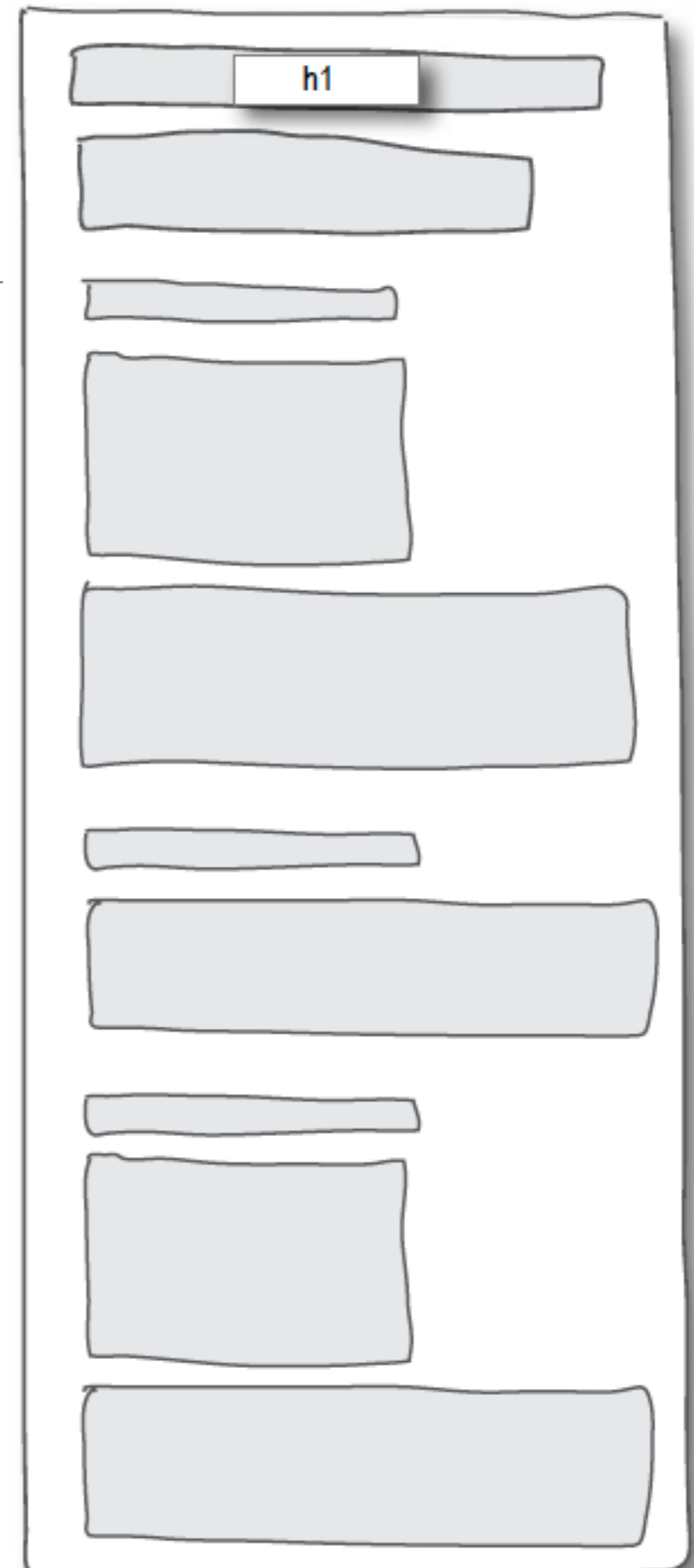
And at the bottom, Tony's first entry, with the image "segway1.jpg".

Last, but not least, don't forget to close your <body> and <html> elements.

# Wire framing

---

- A website wireframe is a basic [visual guide](#) used in [interface design](#) to suggest the structure of a [website](#) and relationships between its pages.
- A webpage wireframe is a similar illustration of the [layout](#) of fundamental elements in the [interface](#). Typically, wireframes are completed before any artwork is developed.



# Agenda

---

- Elements, Attributes, & Documents
- Linking
- Nesting
- Linebreak, Block & Inline Elements
- Structuring a Page / Wireframing

# Learning Outcomes

---

- Understand the structure of an **HTML Element**, and be able to recognize its variants.
- Be able to read and compose a **relative path**, and be able to distinguish it from an **absolute path**.
- Understand the implications of **nesting** of elements, and in particular be able to distinguish between correct and incorrect nesting.
- Be able to differentiate between **block** and **inline** elements
- Understand the general context of **wireframing**