HTML

- HTML provides the ***structure*** and ***content*** of the web.
    - HTML is the first thing the browser reads to display a page.

- HTML is an evolving language.
    - The current standard (as per W3C) is HTML5.
        - HTML5 is both a language specification and a label for a larger set of technologies (multimedia, etc.) for building modern web pages.

- HTML is a *markup language* (specifically HyperText Markup Language).
    - i.e. it is a specific syntax for “marking up” plain text to enhance it and to provide structure.

- HTML markup takes the form of `<tags></tags>`.
    - A tag can encapsulate some text:

      ```
      <h1>This is a header</h1>
      ```

    - A tag can contain another tag:

      ```
      <p>Some of this text <em>will be emphasized</em></p>
      ```

    - A tag can be empty and/or have attributes:

      ```
      <img src="https://placekitten.com/300/300">
      ```

- NOTE: the browser is our development environment for this work. All modern browsers have developer tools to help us out with that work.
    - e.g. in Chrome:

    - These tools are extremely helpful when you're developing a web page, so figure out how to access them in your browser.
        - e.g. Google "Firefox developer tools"

Basic HTML skeleton

- Every HTML page has the same basic skeleton:

```
<!DOCTYPE html>
<html>
     <head>
          <meta charset="utf-8">
          <title><!-- Your page title --></title>
     </head>
     <body>
          <!-- Your page content goes here. -->
     </body>
</html>
```

    - `<!DOCTYPE html>` tells the browser to interpret your HTML and CSS according to the W3C standard.
    - `<html></html>` encloses all the HTML for the page.
    - `<head></head>` contains information about the page that isn't displayed on the page.
        - `<meta charset="utf-8">` says we're using UTF-8 characters.
        - `<title></title>` contains the page title (goes in your browser tab).
    - `<body></body>` contains the actual displayed content of your page. We'll see what goes in here below.

Basic structure: headings and paragraphs

- `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>` designate different levels of heading, in decreasing significance.
    - e.g. this is a most significant heading:

```
<h1>This is a very important section of the page</h1>
```

- `<p>` designates a paragraph, e.g.

```
<p>Here's a paragraph of text that will go on my page.  I
hope you find it useful.</p>
```

- We combine these to create structure on the page:

```
<h1>HTML Crash Course</h1>

<h2>Chapter 1: About HTML</h2>

<p>Here's some info about HTML</p>

<h2>Chapter 2: Basic HTML skeleton</h2>

<p>Here's the common skeleton of all HTML pages…</p>
```

- This structure helps provide semantics to the content, which makes it easier for the user to quickly digest and understand your page.

More structure: lists

- There are two types of list in HTML:
  - `<ul>`: unordered list (i.e. a bulleted list)
  - `<ol>`: ordered list (i.e. a numbered/lettered list)

- These work the same way, with each item in the list being represented by an `<li>` (list item) element:

```
<ul>
     <li>A thing in a list</li>
     <li>Another thing</li>
</ul>

<ol>
     <li>The first thing</li>
     <li>The second thing</li>
</ol>
```

- You can nest lists to any level. You can even nest ordered and unordered lists in any order, e.g.:

```
<ul>
     <li>Here's a thing in a list</li>
     <li>
          <ol>
               <li>First sublist thing</li>
               <li>Second sublist thing</li>
          </ol>
     </li>
</ul>
```

Emphasizing text (and a note on assistive technologies)

- There are two main ways to emphasize text in HTML.
    - `<em>`: emphasized text
    - `<strong>`: strongly important text

- By default, the browser displays text within an `<em>` element as italic text and text within a `<strong>` as bold text, but try not to think of these tags as providing italics and bold text.

- Assistive technologies like screen readers rely on `<em>` and `<strong>` elements to figure out how much emphasis to put on the text contained inside.
    - The users of these technologies may not know the connotations of italics and bold text because they may never have actually even *seen* text.

- So, instead, think of `<em>` and `<strong>` as corresponding respectively to emphasized text and strongly important text.
    - We can use CSS to make the text appear however we want for sighted users looking at our site.

Hyperlinks

- Linking a page to other content in a different location (within the same site or elsewhere on the internet) is fundamental.

- We use an `<a>` element to link to another place. The `href` attribute tells the browser where to link to:

```
<a href="https://www.google.com">Here's a link to Google</a>
```

- We can link to a location within the same site using a full URL, like above, but this can be undesirable for a few reasons.
    - When we're developing, we may want to link to local content instead of content on the web.
    - If part of the URL changes, e.g. we switch from HTTP to HTTPS or move to a new domain, we'd have to update all the URLs on our site.

- Instead, we usually link within our own site in one of two ways:
    - Using an absolute path, relative to the top level of our site, e.g.:

```
<a href="/people/staff/deirdra">Deirdra</a>
```

    - Using a path relative to the current file:

```
<a href="staff/deirdra">Deirdra</a>
```

Images

- We use the `<img>` tag to display an image in a page. The `src` attribute provides a URL to the image to display:

```
<img src="http://placekitten.com/480/640/">
```

    - Note that `<img>` is an empty tag, i.e. it does not have a closing `</img>` tag.

- We can use the `alt` attribute to provide descriptive text for the image, used when:
    - The image can't be loaded for some reason. In this case the `alt` text is displayed on the screen instead.
    - A user with an assistive technology like a screen reader is using the site. In this case, the `alt` text will be read.

```
<img src="http://placekitten.com/480/640/" alt="a small kitten">
```

Block content vs. inline content

- There are two important types of element in HTML:
    - **Block level** elements represent visible blocks on a page.
        - They appear on a new line below whatever content came before, and content after them goes on a new line as well.
        - They typically represent structural elements on the page like headers, paragraphs, lists, footers, etc.
        - e.g. `<h1>`, `<p>`, `<ul>`, etc.
        - `<div>` is a generic block level element that we can style as needed with CSS.
    - **Inline elements** represent only small parts of a page's content.
        - They are contained within block level elements.
        - They do not cause a new line to appear in the page.
        - e.g. `<a>`, `<img>`, `<em>`, `<strong>`, etc.
        - `<span>` is a generic inline element that we can style as needed with CSS.

Advanced structure

- Many modern web pages share some common structure, at a high level.

- HTML5 contains several tags for composing the high-level structure of a web page with block elements.
    - These tags have semantic meaning, which makes it easy for assistive technologies like screen readers to make sense of a page and help with fundamental tasks.
    - `<header>` contains introductory content for the entire site.

        - Contents could include the site's title, a header image, etc.
    - `<nav>` represents a navigation element, such as a list of links to other pages on the site.
        - This is commonly included in the `<header>`.
        - Often also contains a search box.
    - `<main>` contains the main content of the page.
        - This typically contains one or more of the following tags:
            - `<section>` – a part of a page representing one piece of functionality or related content.
            - `<article>` – a block of related content that could stand on its own without the rest of the page (e.g. a blog post).
            - `<div>` – more on this later.
    - `<aside>` contains content that is not directly related to the main content but that can be useful nonetheless.
        - Used for side bars.
    - `<footer>` contains some end content for the page.
        - e.g. copyright info, contact information, navigation links, etc.
    - `<div>` represents a generic block of non-semantic content.
        - Typically used when no semantic tag above nicely represents the block's content.

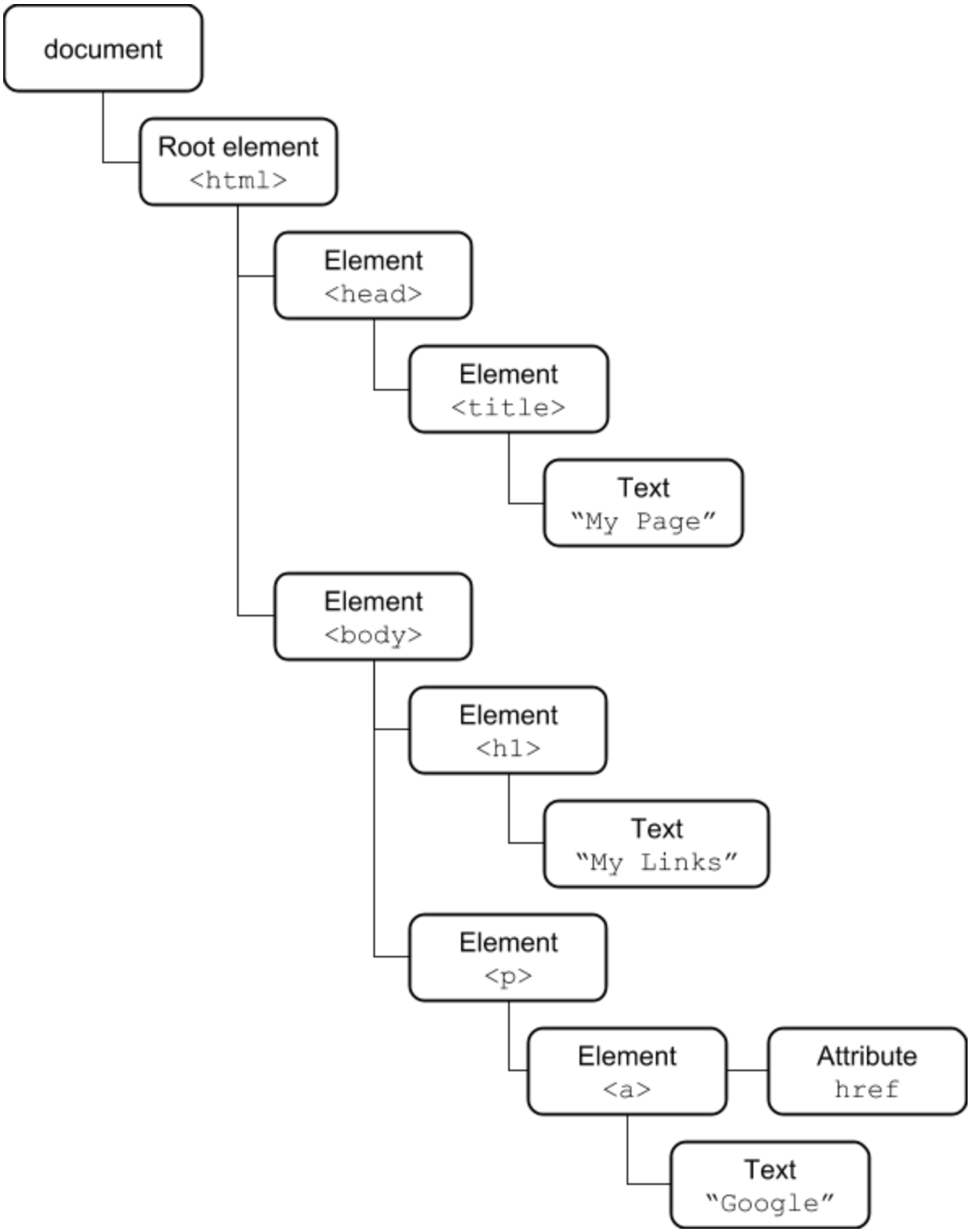
The Document Object Model (DOM)

- The Document Object Model, or DOM, is a representation of an entire HTML page as a tree structure. It is also an API for interacting with that HTML page.
    - The DOM is the representation of HTML stored in memory after parsing.

- Here's some simple HTML and the corresponding DOM representation:

```
<html>
     <head>
          <title>My Page</title>
     </head>
     <body>
          <h1>My Links</h1>
          <p>
               <a href="http://google.com">Google</a>
          </p>
```

```
        </body>
    </html>
```


document
Root element
<html>
Element
<head>
Element
<title>
Text
"My Page"
Element
<body>
Element
<h1>
Text
"My Links"
Element
<p>
Element
<a>
Attribute
href
Text
"Google"