HTML

Frank Walsh

Diarmuid 0'Connor

Agenda

- HTML
  - Overview
  - Fundamentals
  - Elements
- CSS
  - Overview
  - CSS Rules
- CSS Frameworks
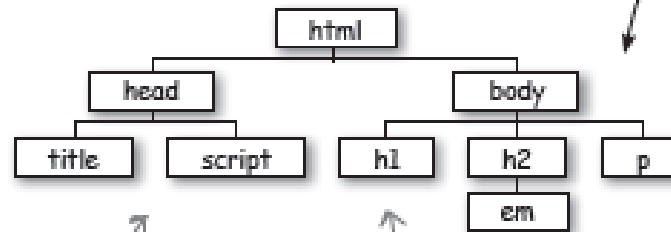 - Overview
  - Example Template

HTML

- Mark up Language of the internet.
- Used to structure and present content in the world wide web.
- HTML5 is intended to subsume HTML4 and XHTML.
- Designed to support latest multimedia
- Features include:
  - New semantic elements(<nav>, <header>, <footer>…
  - New apis (canvas, geolocation, drag and drop, geolocation…)
- See [here](#) for a full description…

How HTML Works 1

❶ The **browser** loads a document, which includes markup written in HTML and style written in CSS.

❷ As the browser loads your page, it also creates an internal **model of your document** that contains all the elements of your HTML markup.

```
                    html
            ┌────────┴────────┐
          head              body
        ┌───┴───┐       ┌──────┼──────┐
      title   script   h1     h2      p
                               │
                              em
```

This is where it gets interesting; for each element in your HTML, the browser creates an object that represents it and places it in a tree-like structure with all the other elements...

...we call this tree the Document Object Model or the DOM for short. You'll be seeing a lot more of the DOM in this book because it plays a vital role in how we add behavior to pages with JavaScript (we'll get to that shortly, in Chapter 2).

Markup

CSS

The page's style (if it has any) comes from CSS3, which has been expanded from CSS2 to include many common idioms that are in use across the Web (like drop shadows and rounded corner borders).
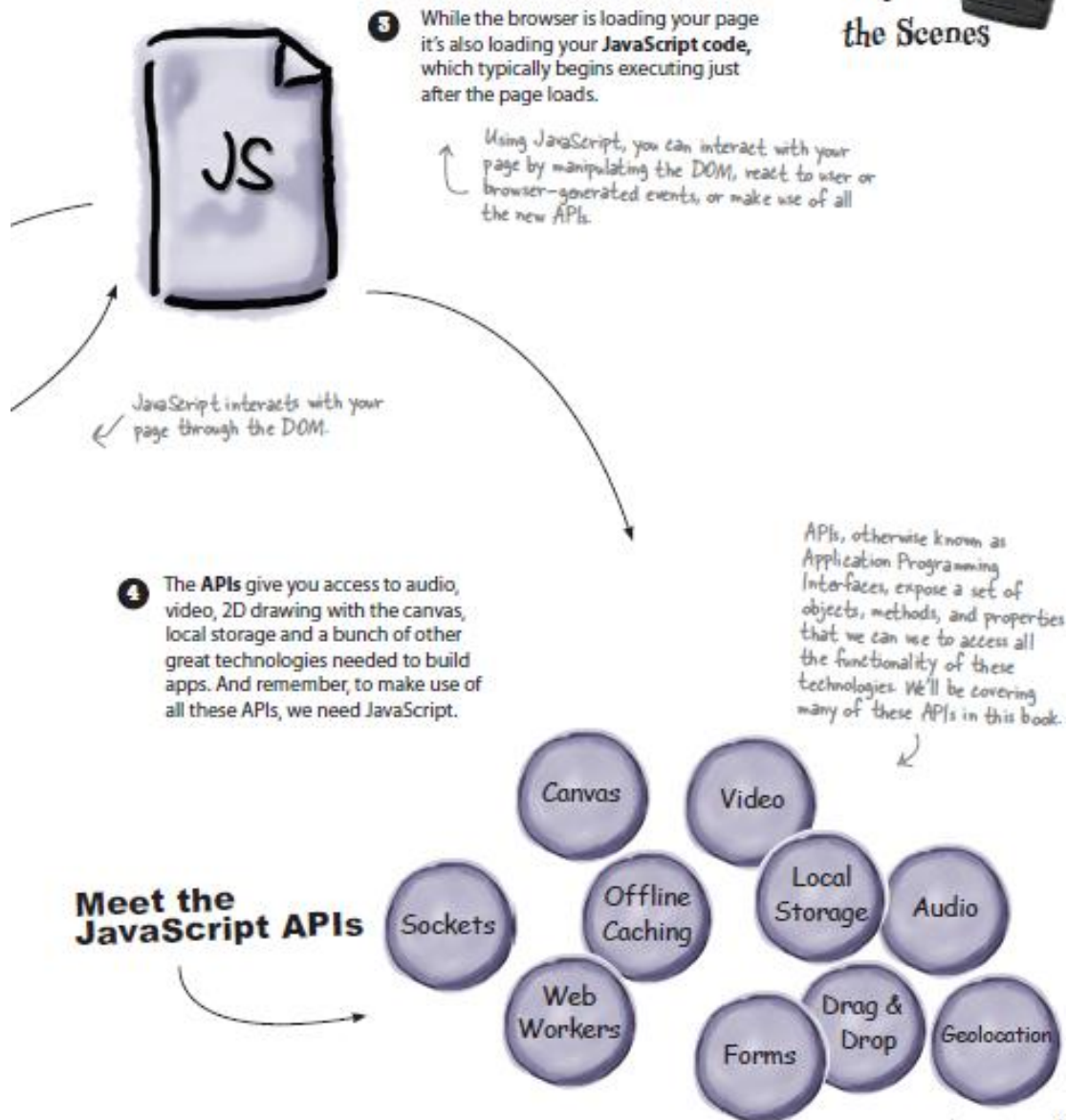
With HTML5 the markup has some improvements, as you've seen with the tags in the <head> element, and there are some additional elements you can use (we'll see a few in this book).
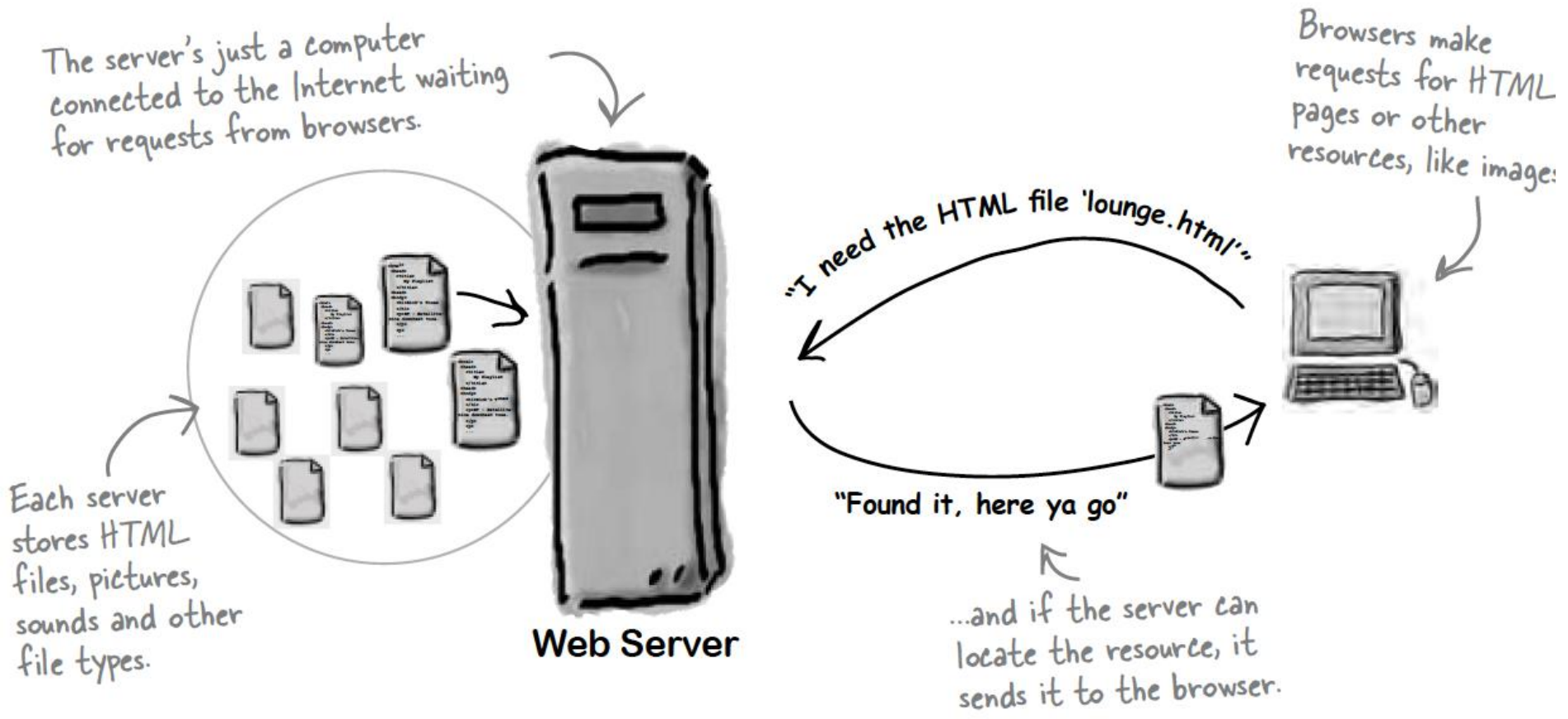
How HTML Works 2

3 While the browser is loading your page it's also loading your **JavaScript code,** which typically begins executing just after the page loads.

Behind the Scenes

Using JavaScript, you can interact with your page by manipulating the DOM, react to user or browser-generated events, or make use of all the new APIs.

JavaScript interacts with your page through the DOM.

4 The **APIs** give you access to audio, video, 2D drawing with the canvas, local storage and a bunch of other great technologies needed to build apps. And remember, to make use of all these APIs, we need JavaScript.

APIs, otherwise known as Application Programming Interfaces, expose a set of objects, methods, and properties that we can use to access all the functionality of these technologies. We'll be covering many of these APIs in this book.

Meet the JavaScript APIs

Canvas
Video
Sockets
Offline Caching
Local Storage
Audio
Web Workers
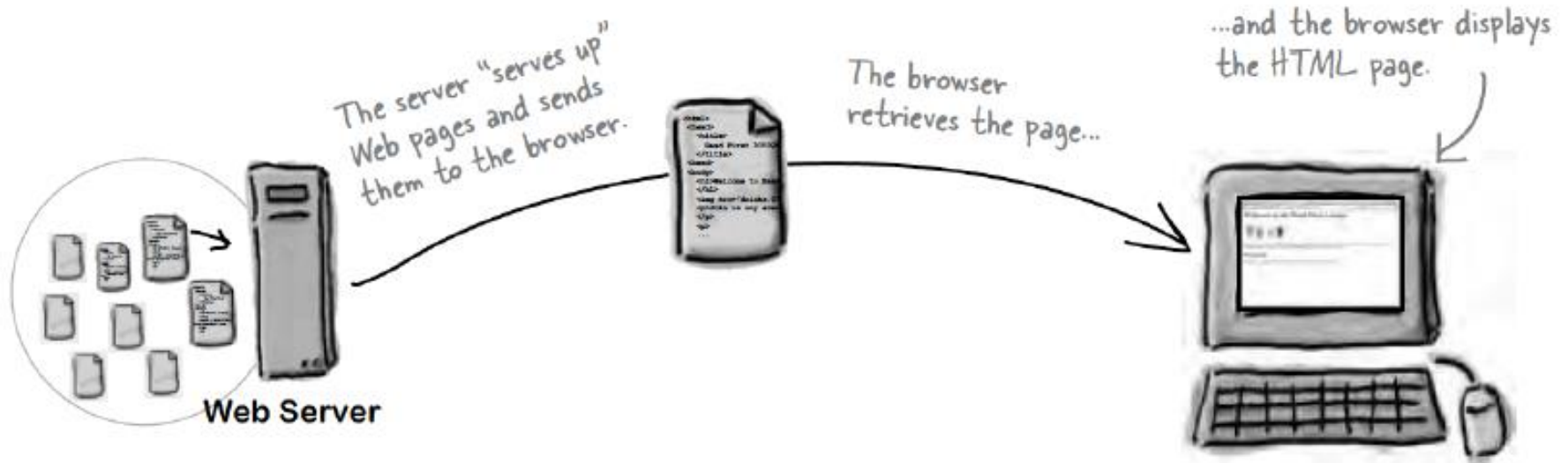Forms
Drag & Drop
Geolocation

HTML Fundamentals: The Web

- **Protocols & Standards**
  - Protocol : agreed vocabulary to enable to programs to communicate
  - Standard: an agreed definition of the structure and meaning of a document
- **Web Protocol**
  - Hyper Text Transfer Protocol - HTTP
- **Web Standard**
  - Hyper Text Markup Language - HTML
  - Web Servers and Web Browsers use HTTP to exchange HTML documents

HTML Fundamentals: Role of Server

The server's just a computer connected to the Internet waiting for requests from browsers.

Browsers make requests for HTML pages or other resources, like images.

"I need the HTML file 'lounge.html'"

"Found it, here ya go"

Each server stores HTML files, pictures, sounds and other file types.

...and if the server can locate the resource, it sends it to the browser.

Web Server

HTML Fundamentals: Role of Client

The server "serves up" Web pages and sends them to the browser.

The browser retrieves the page...

...and the browser displays the HTML page.

Web Server

HTML Fundamentals: HTML

- HTML tell the browser about the structure of your document
  - where the headings are
  - Where paragraphs are
  - Where text has emphasis
  - Where links are
  - What images are to be displayed
- Using HTML alone, browsers have default rules on how to display each element
- Information is conveyed using **"tags"** or **"elements".**
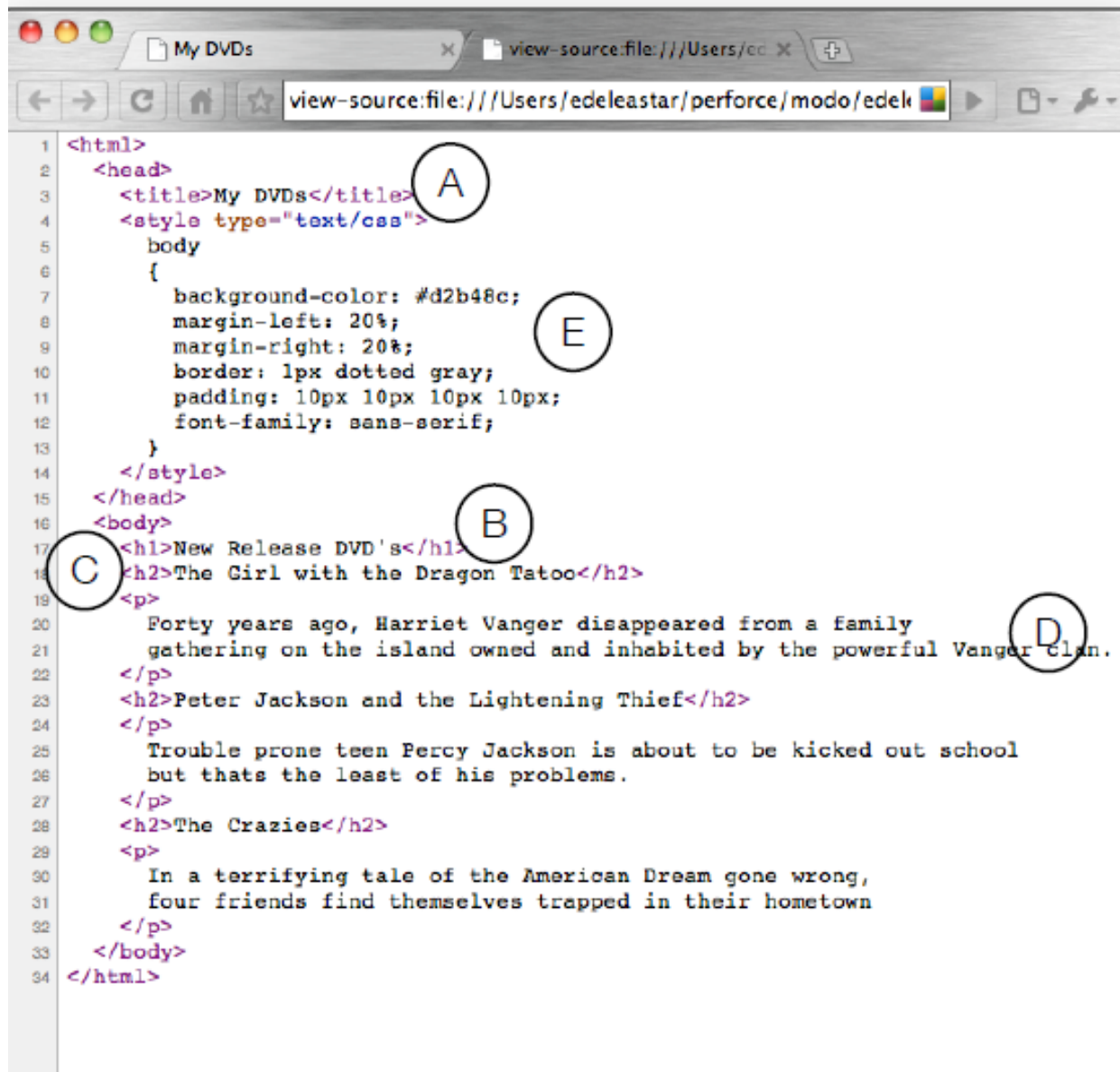
HTML Fundamentals: Rendering
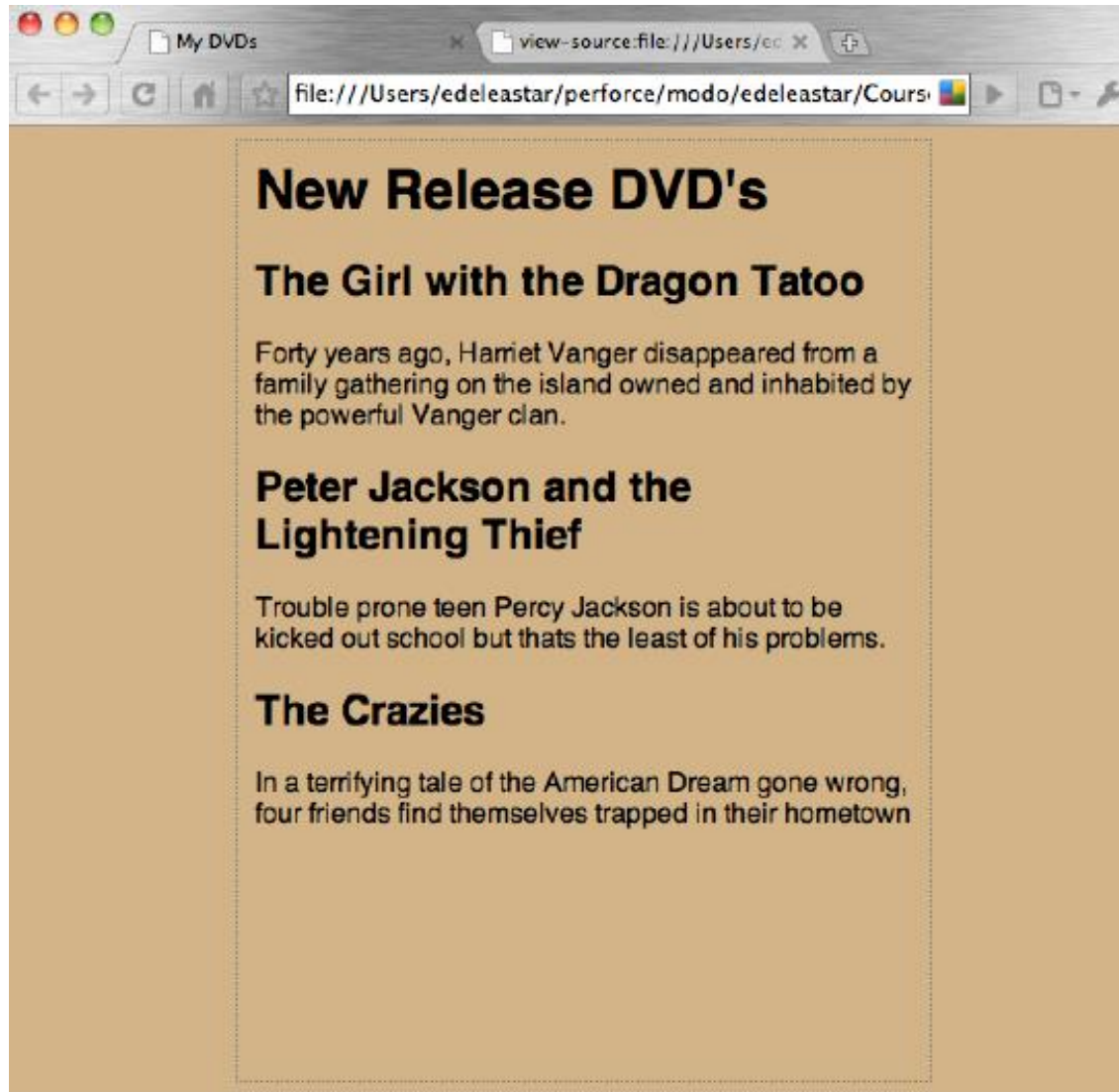
HTML Fundamentals: Elements
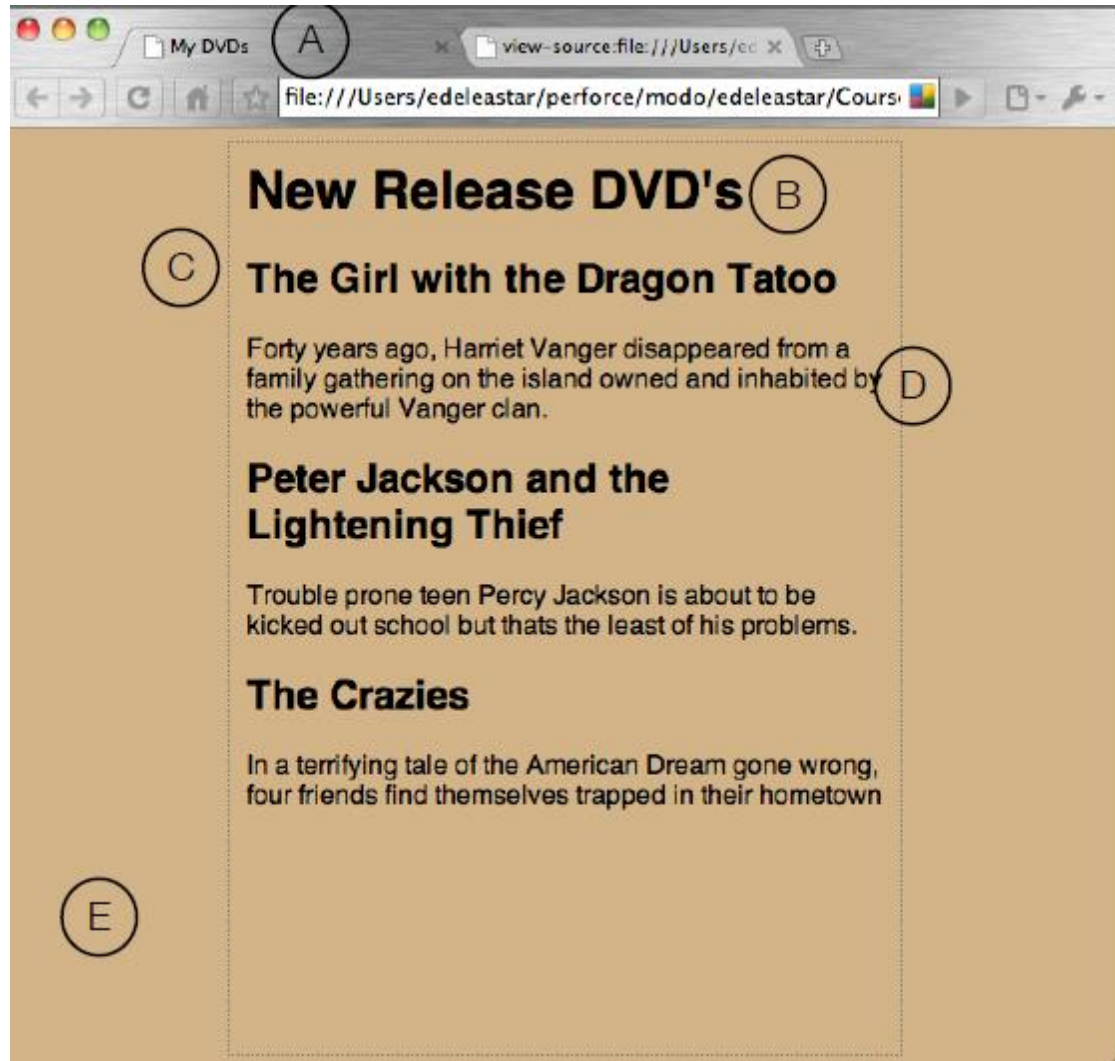
HTML Fundamentals: Source


```html
<html>
  <head>
    <title>My DVDs</title>
    <style type="text/css">
      body
      {
        background-color: #d2b48c;
        margin-left: 20%;
        margin-right: 20%;
        border: 1px dotted gray;
        padding: 10px 10px 10px 10px;
        font-family: sans-serif;
      }
    </style>
  </head>
  <body>
    <h1>New Release DVD's</h1>
    <h2>The Girl with the Dragon Tatoo</h2>
    <p>
      Forty years ago, Harriet Vanger disappeared from a family
      gathering on the island owned and inhabited by the powerful Vanger clan.
    </p>
    <h2>Peter Jackson and the Lightening Thief</h2>
    </p>
      Trouble prone teen Percy Jackson is about to be kicked out school
      but thats the least of his problems.
    </p>
    <h2>The Crazies</h2>
    <p>
      In a terrifying tale of the American Dream gone wrong,
      four friends find themselves trapped in their hometown
    </p>
  </body>
</html>
```

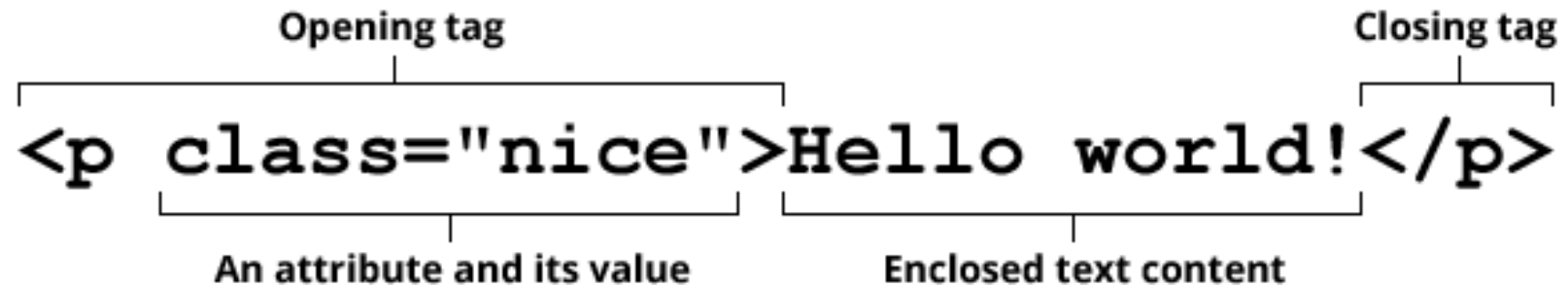# HTML Fundamentals: Elements

```
<html>
<head
<title>
<body>
<h1>
<h2>
<em>
<a>
<img>
<style>
```

- Grouped by purpose:
  - Structural
    - <html>, <head>, <title>, <body>
  - Text
    - Heading
      - <h1>, <h2>
    - Text
      - <p>, <em>

  - Hypertext
    - <a>
  - Image
    - <img>
  - Style
    - <style>

HTML Fundamentals: Anatomy of a HTML Element

Anatomy of an HTML element

Opening tag

Closing tag

`<p class="nice">Hello world!</p>`

An attribute and its value

Enclosed text content

HTML Fundamentals: Attributes

- Attributes let you specify additional information about an element.

Do this (correct form)

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

attribute name · equals sign · double quote · attribute value · double quote

Not this (incorrect form)

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

WRONG — no double quotes around the attribute value.

HTML Fundamentals: Images

- Use the <img> element

```
<img src="../images/pjackson.jpg">
```

```
ElementName: <img>
    Attribute:
        name: src
        value:"../images/pjackson.jpg"
Content: empty
ElementName - none (for the moment)
```
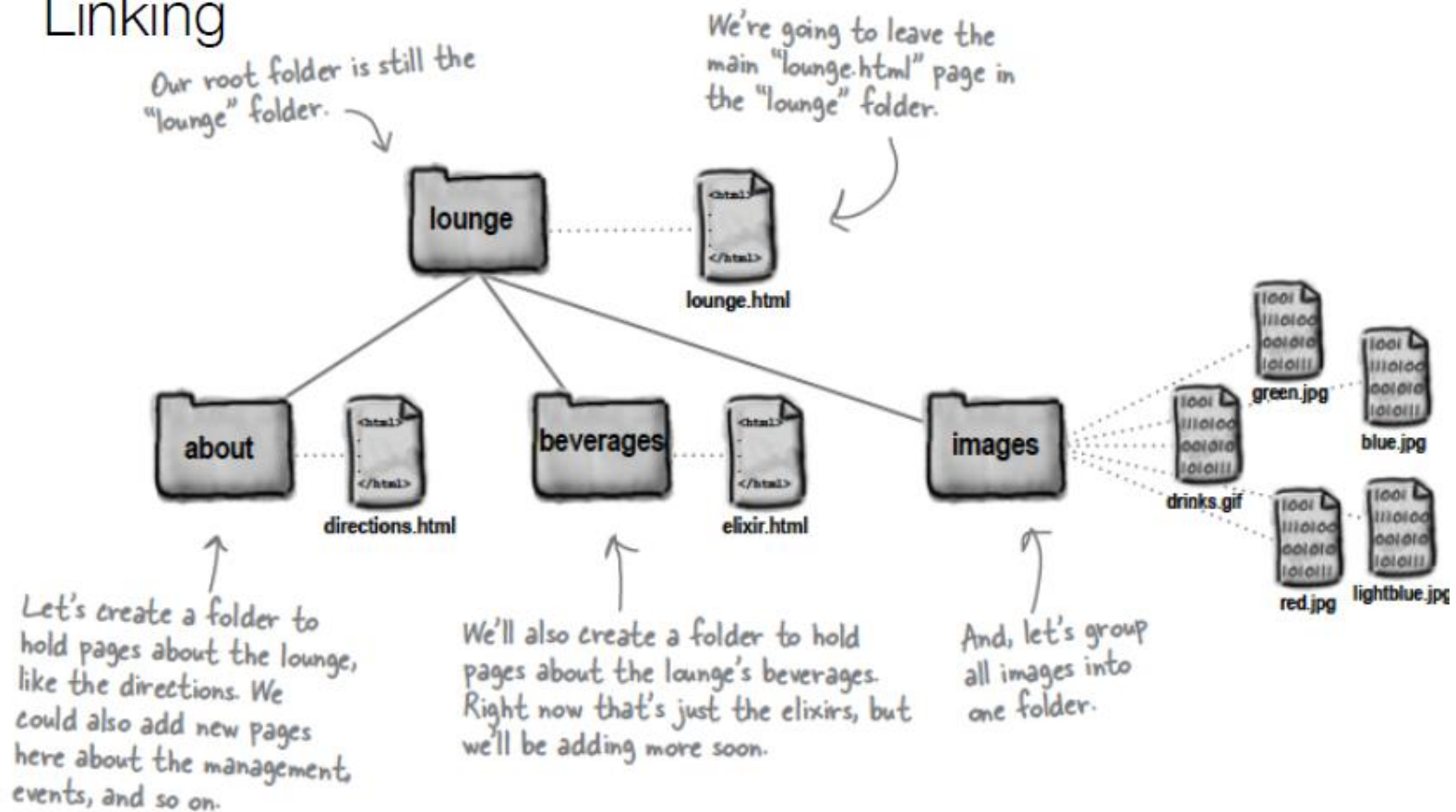
LINKS

HTML Fundamentals: Links

HTML Fundamentals: Link Referencing

- Two ways:
- Absolute
  - Complete path to file
    - C:\labs\lab02\tatoo.jpg
- Relative
  - Trace route from current position to destination
  - ".." means go up a level

HTML Fundamentals: Example

- If in lab02 then *"about/directions.html"* is a relative link from current folder to directions.html.

- If in the catalogue folder then *"../images/Pjackson.jpg"* means *"go up one level and then into images and then link to Pjackson.jpg"*.

- **Avoid absolute links**
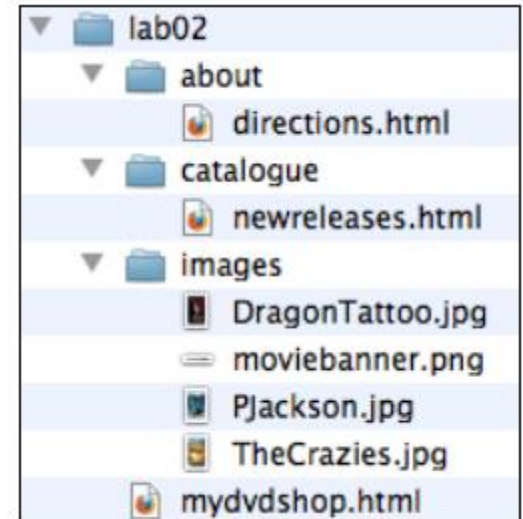

```
<a href="catalogue/newreleases.html">new releases</a>
```

```
<img src="../images/DragonTattoo.jpg">
```

```
<a src="faq.html">FAQ</a>
```

HTML Fundamentals: Nesting

- One element inside another is called nesting.

<html>

<body>

<p>

<q>

<q> nested inside <p>,
nested inside <body>,
nested inside <html>.

HTML Fundamentals: Nesting

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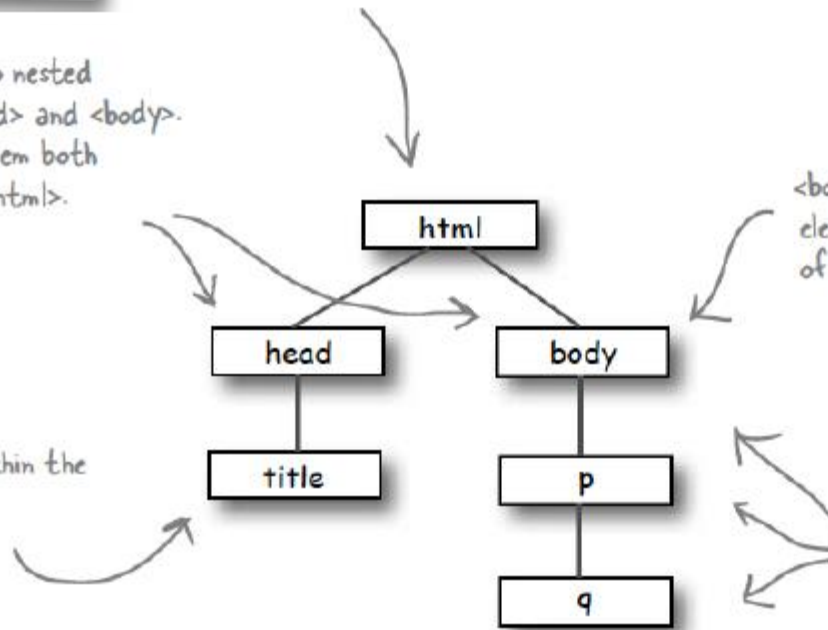
<html> is always the element at the root of the tree.

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

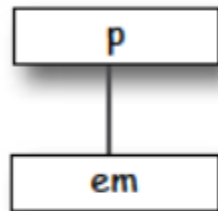<body> is nested within the <html> element, so we say <body> is the "child" of <html>.

```
                    html
                   /    \
              head        body
                |           |
              title         p
                            |
                            q
```

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>.

HTML Fundamentals: Nesting
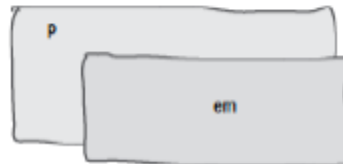


```
<p>I'm so going to blog <em>this</em></p>
```

Good

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

?

Bad

HTML Fundamentals: Line Break

- Break element can be:
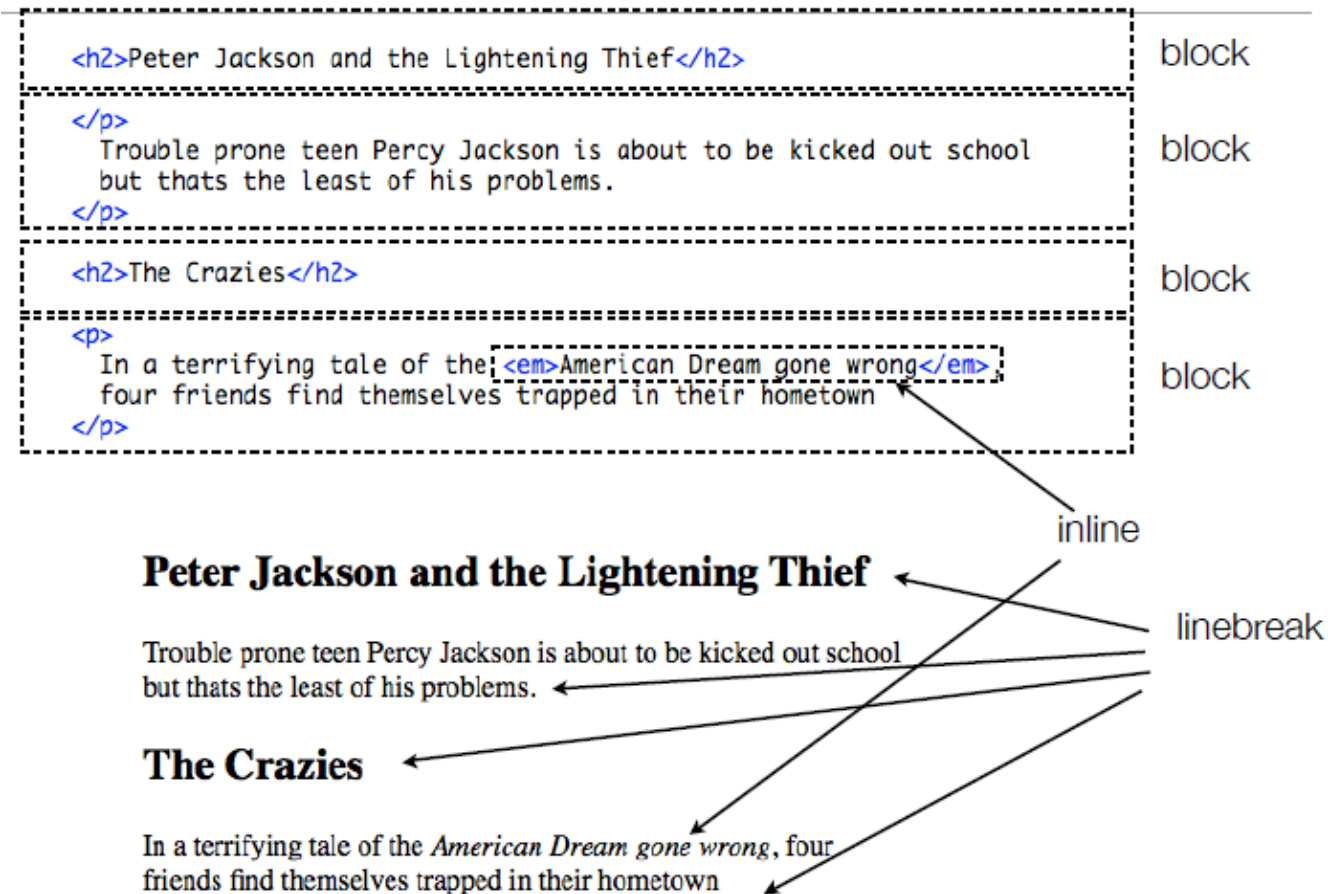 - \
\</br>

  Or normally just

  \

  Or

  \

HTML Fundamentals: Block & Inline

- Block elements always displayed as if they have a line break before them

- Inline elements appear "in line" with the flow. of text

HTML Fundamentals: Block & Inline

```
<h2>Peter Jackson and the Lightening Thief</h2>          block

</p>                                                      block
  Trouble prone teen Percy Jackson is about to be kicked out school
  but thats the least of his problems.
</p>

<h2>The Crazies</h2>                                      block

<p>                                                       block
  In a terrifying tale of the <em>American Dream gone wrong</em>
  four friends find themselves trapped in their hometown
</p>
```

inline

linebreak

Peter Jackson and the Lightening Thief

Trouble prone teen Percy Jackson is about to be kicked out school
but thats the least of his problems.

The Crazies

In a terrifying tale of the *American Dream gone wrong*, four
friends find themselves trapped in their hometown

HTML ELEMENTS

HTML Elements: Categories

- Structural Elements
- Head Elements
- Text Formatting Elements
- List Elements
- Image and Media Elements
- Table Elements
- Form Elements
- Frame and Window Elements
- Deprecated Elements
- Proprietary & Nonstandard Elements

HTML Elements: Structural

- ## **<html>**
    - The html element is the outer container for everything that appears in an HTML document.
    - It can only contain two elements as direct descendants, the head element and body or element.

```
<html>
    <head>
                <title>Page title goes here</title>
    </head>
    <body>
                <!-- content goes here -->
    </body>
</html>
```

HTML Elements: Structural

- **<head>**
    - The head element is the wrapper for all the head elements that:
        - instruct the browser where to find style sheets,
        - define relationships that the document has to others in the web site;
        - provide essential meta information;
        - point to or include scripts that the document will need to apply later on.

HTML Elements: Structural

- **<body>**
  - The body element wraps around all of the content that will be displayed on screen.
  - You should not place character data (text) or inline elements such as em, directly into the body.
  - Instead, these should be contained inside block-level elements such as <p> elements

HTML Elements: Structural

- ## <h1-h6>
  - The h1 element is used to indicate the most important (or highest-level) heading on the page.
  - There are six heading levels to—h1 to h6—to add structure to the web page.
  - A document's first heading should be an h1, followed by one or more h2 headings; each of these h2 headings can then have a further series of h3 headings below them, and so on
  - May contain any text content, but it can't include any block-level elements: only inline elements can be included

```
<h1>News and Views</h1>
<p>
Bits of news, words of advice
</p>
!
<h2>September 3, 2010</h2>
<p>
    <img src="images/robinhood.jpg"
            alt="Robin Hood">
</p>
```

HTML Elements: Head

- **<title>**
  - Displaying a title in the browser toolbar or in the task bar (on Windows).
  - Providing for the document a name that's used by the browser when you add the page as a favorite or bookmark.
  - Displaying a title of the page when it appears in search engine results.

HTML Elements: Head

- **<meta>**
  - The <meta> element is used to specify page description, keywords, author, and other metadata.
  - Providing for the document a name that's used by the browser when you add the page as a favorite or bookmark.
  - Displaying a title of the page when it appears in search engine results.
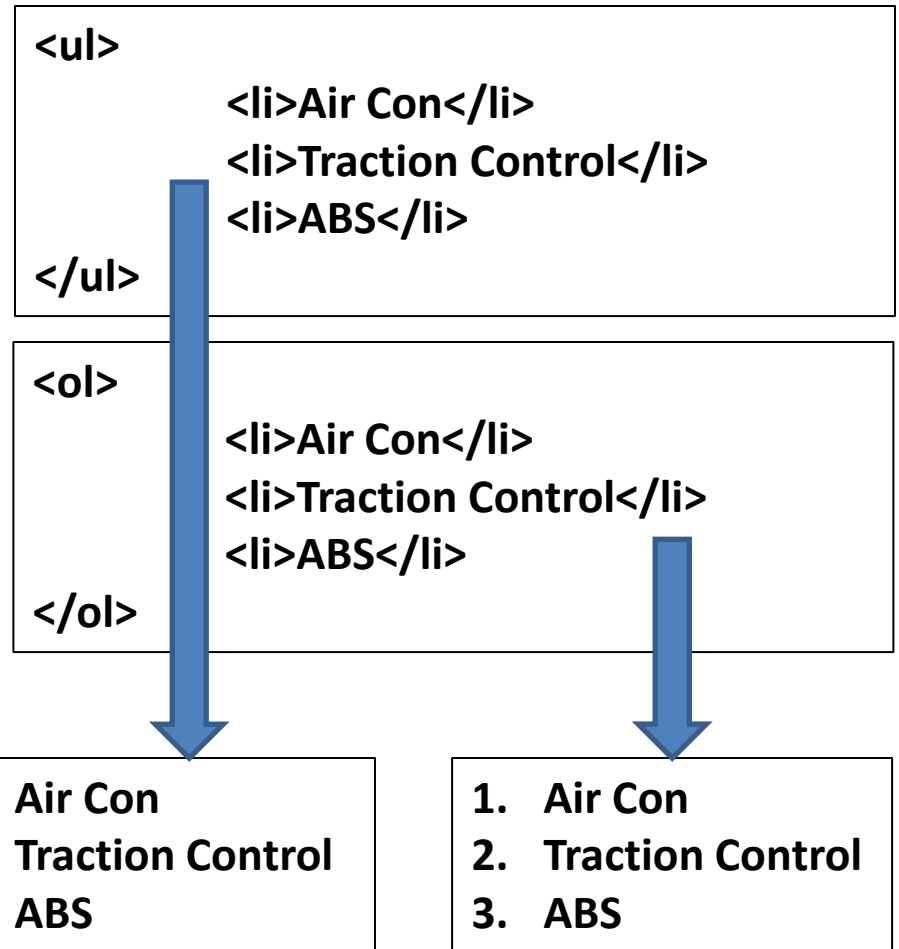 - See this in Bootstrap section.
  - Examples:

```
<meta charset="UTF-8">   <!--specifies character set for page -->

<meta name="description" content="Free Web tutorials on HTML and CSS">
```

HTML Elements: Text Formatting

- **\<a\>**
  - Usually referred to as a link or hyperlink
  - Has a 7 possible attributes in total, of which we have seen just one: href,
  - which defines the document to which the link leads. This may be
    - a web page in the same directory,
    - a page somewhere else on the same server,
    - a location within the current page,
    - a web page—or any another kind of document—stored on another server

`<a href="stock/cars.html">new cars</a>`

HTML Elements: List

- **<ul>**
  - Used to group a collection of items together in a list, but in a way that doesn't suggest an order of precedence or importance.

- **<ol>**
  - Same as <ul> except bullets replaced by numbers

```
<ul>
        <li>Air Con</li>
        <li>Traction Control</li>
        <li>ABS</li>
</ul>
```

```
<ol>

        <li>Air Con</li>
        <li>Traction Control</li>
        <li>ABS</li>
</ol>
```

- **Air Con**
- **Traction Control**
- **ABS**

1. **Air Con**
2. **Traction Control**
3. **ABS**

HTML Elements: div

- **<div>** Tag
  - defines a division or a section in an HTML document
  - often used to group block-elements to format them with CSS
  - Used extensively in front-end Javascript and CSS frameworks (e.g. React and Bootstrap)

```
<div id="navbar" class="collapse navbar-collapse">
        <ul class="nav navbar-nav">
                <li class="active"><a href="#">Home</a></li>
                <li><a href="#about">About</a></li>
                <li><a href="#contact">Contact</a></li>
        </ul>
</div>
```

HTML Elements: script

- <script> tag:
  - used to define a client-side script (JavaScript)
  - either contains scripting statements, or it points to an external script file through the src attribute.
- Use the script tag to pull in client side js frameworks
  - Jquery, React…

```
<!DOCTYPE html>
<html>
<body>
   <p id="demo"></p>
<script>
  document.getElementById("demo").innerHTML = "Hello JavaScript!";
</script>

</body>
</html>
```

CSS

Markup, Style, Function

- Markup (HTML)
  - Structure
  - Content
- Style (CSS)
  - Style
  - Presentation
  - Appearance
- Function (Javascript)
  - Actions
  - Manipulations
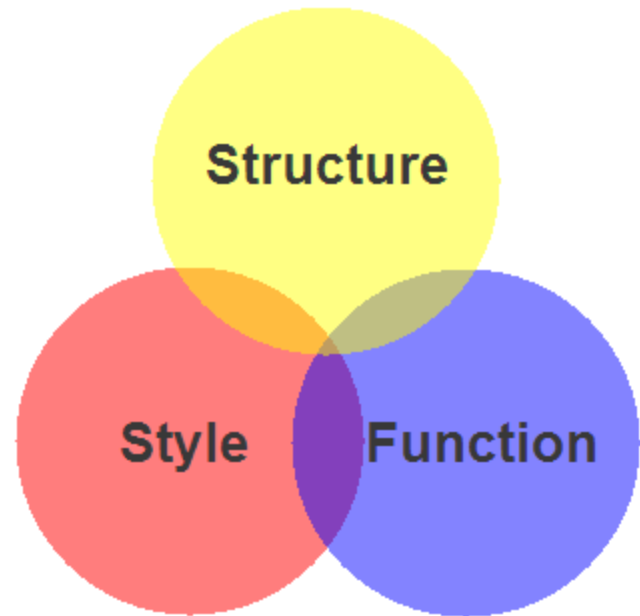
CSS Advantages

- CSS provides superior control over presentation than do presentational element types in HTML.

- The separation of content from presentation makes it easier for site owners to reuse the content for other purposes, such as RSS feeds or text-to-speech conversion.

CSS Rules

The first thing you do is select the element you want to style, in this case the <p> element. Notice in CSS, you don't put <> around the name.

Then you specify the property you want to style, in this case the <p> element's background color.

And you're going to set the background-color to red.

```
p {
      background-color: red;
}
```

Place all the styles for the <p> element in between { } braces.

There's a colon in between the property and its value.

At the end, put a semicolon.

We call the whole thing a RULE.

You could also write the rule like this:

CSS Rules

- You can add as many properties and values as you like in each CSS rule.

- To put a border around your paragraphs:

```
p {
    background-color: red;
        border: 1px solid gray;
}
```

All you have to do is add another property and value.

The <p> element will have a border...

...that is 1 pixel thick, solid, and gray.

How to Use CSS

- Inline
  - `<span style="font-size: 20px;margin-left: 10px">`
- Style CSS tag
  - `<span class="span-norm">…</span>`
  - `<style>`
    `.span-norm {font-size: 20px;margin-left: 10px}`
    `</style>`
- External CSS file (put in \<head\>)
  - `<link rel="stylesheet" href="style.css"/>`

CSS file

- Best practice to separate your style out into a css file and link to it in the head element.

Use the link element to "link in" external information.

The type of this information is "text/css". In other words, a CSS style sheet.

And the style sheet is located at this href (in this case we're using a relative link, but it could be a full-blown URL).

`<link type="text/css" rel="stylesheet" href="lounge.css" />`

The rel attribute specifies the relationship between the XHTML file and the thing you're linking to. We're linking to a style sheet, so we use the value "stylesheet".

<link> is an empty element.

CSS Rules

- Rules can be combined if they are identical

```
h1 {
    font-family:  sans-serif;
    color:        gray;
}


h2 {
  font-family:  sans-serif;
  color:        gray;
}


p {
   color:  maroon;
}
```

Here's the rule to select <h1> elements and change the font-family to sans-serif and the font color to gray. We'll talk a lot more about fonts later.

And here's another rule to the exact same thing to the <h2> element.

```
h1, h2 {
  font-family:  sans-serif;
  color:        gray;
}


p {
   color:  maroon;
}
```

To write a rule for more than one element, just put commas between the selectors, like "h1, h2".
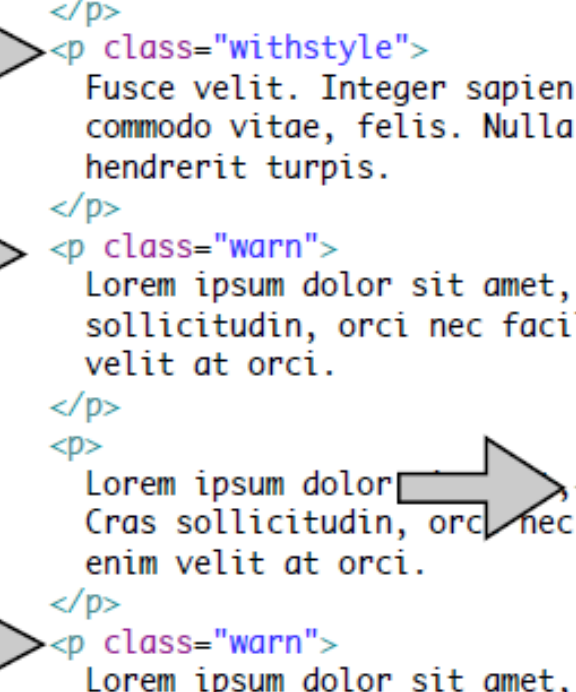
CSS Selectors

- Class attribute
  - While the name of an element specifies its *type*, the class attribute lets you assign to it one or more *subtypes*.
  - These subtypes may then be used in CSS code for styling purposes.
  - In particular, several elements may share a class and only those elements can be identified for restyling

CSS: Using class to identify elements

```html
<body>
  <p>
    Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Cras sollicitudin, orci
    nec facilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.
  </p>
  <p class="withstyle">
    Fusce velit. Integer sapien enim, rhoncus vitae, cursus non,
    commodo vitae, felis. Nulla convallis ante sit amet urna. Maecenas condimentum
    hendrerit turpis.
  </p>
  <p class="warn">
    Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Cras
    sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim
    velit at orci.
  </p>
  <p>
    Lorem ipsum dolor             , <span class="warn">consectetuer adipiscing elit</span>.
    Cras sollicitudin, orc nec acilisis vehicula, neque urna porta risus, ut sagittis
    enim velit at orci.
  </p>
  <p class="warn">
    Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Cras
    sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim
    velit at orci.
  </p>
</body>
```

CSS: Using class to identify elements

- Class names are referenced in css as element.classname

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Cras sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.

Fusce velit. Integer sapien enim, rhoncus vitae, cursus non, commodo vitae, felis. Nulla convallis ante sit amet urna. Maecenas condimentum hendrerit turpis.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Cras sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Cras sollicitudin, orci nec acilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Cras sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.

```
p
{
  background-color: white;
  color: black;
  font-family: times;
  margin: 0.5em;
  padding: 0.5em;
}

p.withstyle
{
  background-color: olive;
  color: navy;
  font-family: sans-serif;
  margin: 0.5em;
  padding: 0.5em;
}

p.warn
{
  background-color: yellow;
  color: red;
  font-weight: bold;
}
```

CSS: Using class to identify elements

- Also can have just .class name
  - Selector now matches any element of the given class

```css
p
{
    background-color: white;
    color: black;
    font-family: times;
    margin: 0.5em;
    padding: 0.5em;
}

.withstyle
{
    background-color: olive;
    color: navy;
    font-family: sans-serif;
    margin: 0.5em;
    padding: 0.5em;
}

.warn
{
    background-color: yellow;
    color: red;
    font-weight: bold;
}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Cras sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.

Fusce velit. Integer sapien enim, rhoncus vitae, cursus non, commodo vitae, felis. Nulla convallis ante sit amet urna. Maecenas condimentum hendrerit turpis.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Cras sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Cras sollicitudin, orci nec acilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Cras sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.

CSS: Using class to identify elements

- You can also use multiple CSS classes on a single element
  - For example, this paragraph has three classes:

    ```
    <p class="pullquote featured left">...</p>
    ```

- Bootstrap example

  ```
  <ul class="nav navbar-nav">

  ....

  </ul>
  ```

IDs

- Id attribute provides a unique identifier for an element.
- Must be unique **– no other element in the same page should have the same id**

```
<p id="mainpoint">
  Fusce velit. Integer sapien enim, rhoncus vitae, cursus non,
  commodo vitae, felis. Nulla convallis ante sit amet urna. Maecenas condimentum
  hendrerit turpis.
</p>
```

```
#mainpoint
{
background-color: yellow;
color: red;
font-weight: bold;
}
```

Notice that, in a CSS, an id selector starts with a '#'.

CSS FRAMEWORKS

What's a CSS Framework

- Easier, more standards-compliant web design for creating web applications.

- Usually contains HTML and CSS-based design templates for styling text, forms, buttons, navigation.

- Also has optional JavaScript extensions.

- Most of these frameworks contain at least a **grid.**

Bootstrap

- Can include Bootstrap from a Content Delivery Network (CDN).
  - Already present in many other web apps so cached.
  - Served from closest cdn server.
- CDN example: https://www.maxcdn.com/
- Include the following in the head element:

```
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css">

<!-- jQuery library -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>

<!-- Latest compiled JavaScript -->
<script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.2.0/js/bootstrap.min.js"></script>
```

Bootstrap is "Mobile First"

- Designed to work on heterogenous mobile, touchscreen devices.

- Use a <meta> tag in the <head> element to ensure proper rendering.

```html
<meta name="viewport" content="width=device-width, initial-scale=1">
```

The **width=device-width** sets the width of the page to follow the screen-width of the device (which will vary depending on the device).

The **initial-scale=1** part sets the initial zoom level when the page is first loaded by the browser.

Bootstrap uses Containers

- Bootstrap requires a containing element with a **container** or **container-fluid** classifier to wrap site contents.

```html
<div class="container">
  <h1>My First Bootstrap Page</h1>
  <p>This is some text.</p>
</div>
```

Bootstrap Grid

- Responsive, fluid grid system that appropriately scales up to 12 columns
- predefined classes for easy layout options
- Use rows and columns
  - Rows must be placed within a .container class for proper alignment and padding. Create horizontal groups of columns
  - Content should be placed within columns
  - classes like **.row** and **.col-xs-4** are available for quickly making grid layouts

Grids
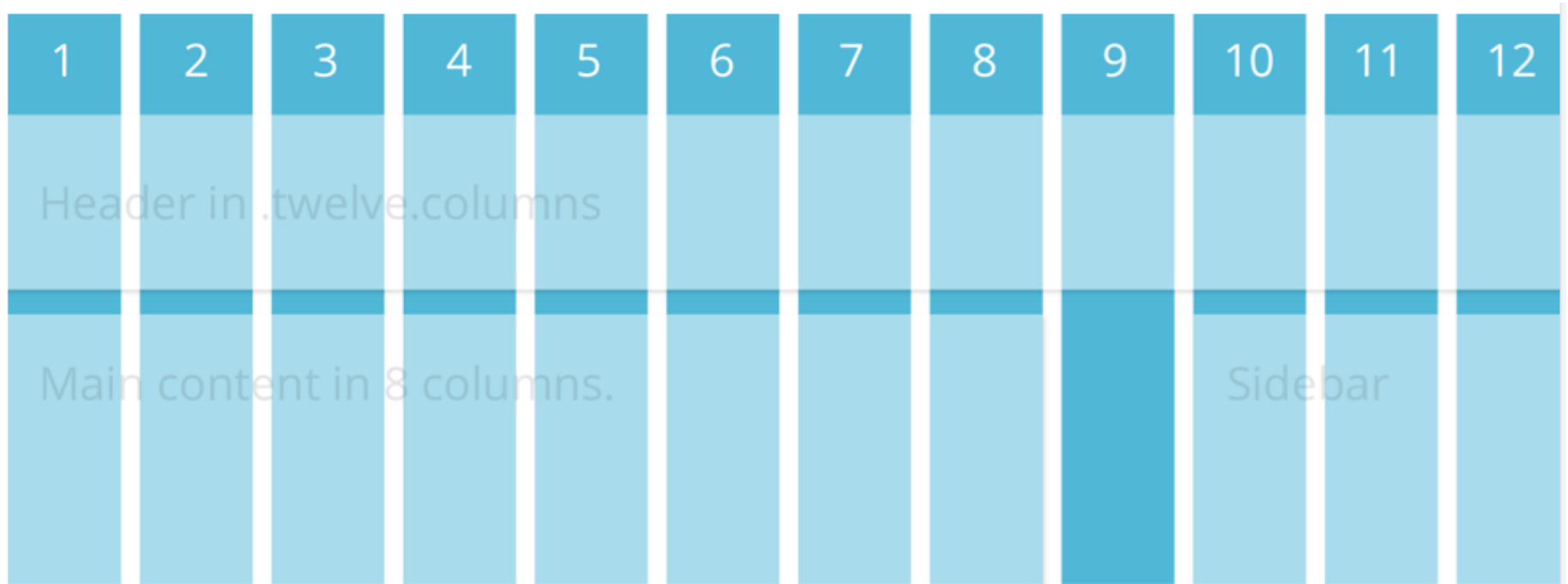
Grid Configurations

	Extra small devices Phones (<768px)	Small devices Tablets (≥768px)	Medium devices Desktops (≥992px)	Large devices Desktops (≥1200px)
Grid behavior	Horizontal at all times	Collapsed to start, horizontal above breakpoints	Collapsed to start, horizontal above breakpoints	Collapsed to start, horizontal above breakpoints
Max container width	None (auto)	750px	970px	1170px
Class prefix	**.col-xs-**	**.col-sm-**	**.col-md-**	**.col-lg-**
# of columns	12	12	12	12
Max column width	Auto	60px	78px	95px
Gutter width	30px (15px on each side of a column)	30px (15px on each side of a column)	30px (15px on each side of a column)	30px (15px on each side of a column)
Nestable	Yes	Yes	Yes	Yes
Offsets	Yes	Yes	Yes	Yes
Column ordering	Yes	Yes	Yes	Yes

Basic Grid Structure

```html
<div class="container">
    <div class="row">
        <div class="col-*-*"></div>
        <div class="col-*-*"></div>
    </div>
    <div class="row">...</div>
</div>
<div class="container">....
```

Bootstrap Template

- Example Bootstrap Template

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap 101 Template</title>
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">

<!-- Optional theme -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-theme.min.css">

  </head>  <body>
    <h1>Hello, world!</h1>

    <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
    <!-- Include all compiled plugins (below), or include individual files as needed -->
    <script src="js/bootstrap.min.js"></script>
  </body>
</html>
```