CS637 Class 2

We are looking at slides from [HTML and CSS: Design and Build Websites](https://www.amazon.com/HTML-CSS-Design-Build-Websites/dp/1449319897) by Jon Duckett

Slides are posted on the class website, protected by a password written on the board.

Note: Exams are open-print-books, no laptops, no cell phone use (at your seat), so get a print copy of this book if you want to use it for exams.

If you must take a phone call during an exam, bring the phone to the front of the class.
CHAPTER 2

TEXT
<h1>This is a Main Heading</h1>
<h2>This is a level 2 heading</h2>
<h3>This is a level 3 heading</h3>
<h4>This is a level 4 heading</h4>
<h5>This is a level 5 heading</h5>
<h6>This is a level 6 heading</h6>
This is a Main Heading

This is a Level 2 Heading

This is a Level 3 Heading

This is a Level 4 Heading

This is a Level 5 Heading

This is a Level 6 Heading
A paragraph consists of one or more sentences that form a self-contained unit of discourse. The start of a paragraph is indicated by a new line.

Text is easier to understand when it is split up into units of text. For example, a book may have chapters. Chapters can have subheadings. Under each heading will be one or more paragraphs.
A paragraph consists of one or more sentences that form a self-contained unit of discourse. The start of a paragraph is indicated by a new line.

Text is easier to understand when it is split up into units of text. For example, a book may have chapters. Chapters can have subheadings. Under each heading there will be one or more paragraphs.
The moon is drifting away from the earth.
The moon is drifting away from Earth.
The Earth gets one hundred tons heavier every day due to falling space dust.

Note: `<br>` is the usual way to write the line break.
The Earth gets one hundred tons heavier every day due to falling space dust.
The Earth
gets one hundred tons heavier every day
due to falling space dust.
<p><strong>Beware</strong> pickpockets operate in this area.</p>

<p>I <em>think</em> Ivy was the first.</p>

<p>I think <em>Ivy</em> was the first.</p>

<p>I think Ivy was the <em>first</em>.</p>
Beware: Pickpockets operate in this area.

I think Ivy was the first.

I think Ivy was the first.

I think Ivy was the first.
CHAPTER 4
LINKS
WRITING LINKS

<a href="http://www.imdb.com">IMDB</a>
WRITING LINKS

THE PAGE THE **LINK** TAKES YOU TO

```html
<a href="http://www.imdb.com">IMDB</a>
```
WRITING LINKS

THE PAGE THE LINK TAKES YOU TO

<a href="http://www.imdb.com">IMDB</a>

THE TEXT THE USER CLICKS ON
LINKING TO OTHER SITES

<a href="http://www.empireonline.com">Empire</a>
Empire
LINKING TO OTHER PAGES ON THE SAME SITE

<a href="index.html">Home</a>

<a href="about.html">About</a>

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

<a href="contact.html">Contact</a>
• Home
• About
• Movies
• Contact
relative urls

examplearts
  index.html
movies
  cinema
    index.html
    listings.html
    reviews.html
dvd
    index.html
    listings.html
    reviews.html
music
  index.html
  listings.html
  reviews.html
theater
  index.html
  listings.html
  reviews.html
RELATIVE
URLS

SAME
reviews.html
RELATIVE
URLS

SAME
reviews.html

CHILD
music/index.html
RELATIVE
URLS

SAME
reviews.html

CHILD
music/index.html

PARENT
../index.html
RELATIVE URLs

SAME
reviews.html

CHILD
music/index.html

PARENT
../index.html

GRANDCHILD
movies/dvd/index.html
RELATIVE URLS

SAME
reviews.html

CHILD
music/index.html

PARENT
../index.html

GRANDCHILD
movies/dvd/index.html

GRANDPARENT
../../index.html
EMAIL LINKS

<a href="mailto:jon@example.org">Email Jon</a>
<a href="mailto:jon@example.org">Email Jon</a>
To: jon@example.org

Subject:
Film-Making Terms

Arc Shot
Interlude
Prologue

Arc Shot
A shot in which the subject is photographed by an encircling or moving camera

Interlude
A brief, intervening film scene or sequence, not specifically tied to the plot, that appears within a film
LINKING TO A SPECIFIC PART OF THE SAME PAGE

<h1 id="top">Film-Making Terms</h1>

<a href="#arc-shot">Arc shot</a>
<a href="#interlude">Interlude</a>
<a href="#prologue">Prologue</a>

<h2 id="prologue">Prologue</h2>

<a href="#top">Top</a>

---

<a href="#arc-shot">Arc Shot</a>
An arc shot is a shot in which the subject is photographed by an encircling or moving camera.

<a href="#interlude">Interlude</a>
A brief, intervening film scene or sequence, not specifically tied to the plot, that appears within a film
LINKING TO A SPECIFIC PART OF THE SAME PAGE

<h1 id="top">Film-Making Terms</h1>

<a href="#arc-shot">Arc shot</a>

<a href="#interlude">Interlude</a>

<a href="#prologue">Prologue</a>

<h2 id="prologue">Prologue</h2>

<a href="#top">Top</a>

Arc Shot
A shot in which the subject is photographed by an encircling or moving camera

Interlude
A brief, intervening film scene or sequence, not specifically tied to the plot, that appears within a film

Prologue
A speech, preface, introduction, or brief scene preceding the main action or plot of a film; contrast to epilogue

Top
LINKING TO A SPECIFIC PART OF THE SAME PAGE

**Arc Shot**

A shot in which the subject is photographed by an encircling or moving camera

**Interlude**

A brief, intervening film scene or sequence, not specifically tied to the plot, that appears within a film

**Prologue**

A speech, preface, introduction, or brief scene preceding the main action or plot of a film; contrast to epilogue
Film-Making Terms

Arc Shot
A shot in which the subject is photographed by an encircling or moving camera

Interlude
A brief, intervening film scene or sequence, not specifically tied to the plot, that appears within a film
CHAPTER 5
IMAGES
Images can set the tone for a site in less time than it takes to read a description.
Images are subject to copyright but there are stock photography sites where you can buy them.
 Getty Images search results for macaroons.
If a page shows several photos of products or members of a team, keep them consistent.
<img src="images/quokka.jpg" alt="A family of quokka" title="The quokka is an Australian marsupial that is similar in size to the domestic cat" />
After right-click (not on image), View Source
we can see the HTML for the page

This is using Chrome, but this can be done in any desktop browser.
Chrome on smartphone: need to use URL view-source:http://whatever.
Can carefully edit URL with pencil that shows up with tap on URL.
<img src="images/quokka.jpg" alt="A family of quokka" title="The quokka is an Australian marsupial that is similar in size to the domestic cat" width="600" height="450" />
Resizing

- Although browsers will resize images as specified by width and height, it’s not a great idea to use it.
- Can end up with distorted or fuzzy images.
- Better to make another right-size image using a graphics editor and use that.
How a page with an image gets displayed

• In this case, the user selects the page as usual
• The browser does a GET request to the server
• The server sends the HTML page with the `<img>`…
• The browser receives the HTML, parses the `<img>` URL, and issues another GET request for the image data (no user involvement here)
• The server sends the image data back
• The browser shows the completed page to the user
• This means two complete “request cycles” to the server
• We can make a chart showing this communication…
Communications Diagram: page with `<img>`
(time flows down in diagram)

1. User: requests `.html` page via browser

2. Server: sees `GET /...page.html`,
   Returns HTML on same connection

3. Browser: parses page, sees `<img>`,
   requests image data

4. Server: sees `GET /.../quokka.jpg`,
   return its data in same connection.

5. User: sees finished page
Chrome can show details on the two requests

Use right-click, inspect, choose Network tab, reload page
One page can involve many HTTP requests (from last class)

From https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview
There are around 10,000 living species of birds that inhabit different ecosystems from the Arctic to the Antarctic. Many species undertake long distance annual migrations, and many more perform shorter irregular journeys.
There are around 10,000 living species of birds that inhabit different ecosystems from the Arctic to the Antarctic. Many species undertake long distance annual migrations, and many more perform shorter irregular journeys.
There are around 10,000 living species of birds that inhabit different ecosystems from the Arctic to the Antarctic. Many species undertake long-distance annual migrations, and many more perform shorter irregular journeys.
There are around 10,000 living species of birds that inhabit different ecosystems from the Arctic to the Antarctic. Many species undertake long distance annual migrations, and many more perform shorter irregular journeys.
<img> is “inline”

• Now we look at cases where the <img> element lies inside the <p> element.
• <p> is a block element, starts a new line
• <img> Image is an inline element, keeps going on the current line
• Other block elements: <h1>, <ul>, <li>, …
• Other inline elements: <em>, <a>, <strong>
There are around 10,000 living species of birds that inhabit different ecosystems from the Arctic to the Antarctic. Man species undertake long distance annual migrations, and many more perform shorter irregular journeys.
There are around 10,000 living species of birds that inhabit different ecosystems from the Arctic to the Antarctic. Many species undertake long distance annual migrations, and many more perform shorter irregular journeys.

http://htmlandcssbook.com/code-samples/chapter-05/where-to-place-images.html
“OLD CODE”

• In the next few slides and in Duckett, “OLD CODE” means pre-HTML5

• In HTML5, we use CSS to arrange non-default positioning

• Skip “OLD CODE” for this class, but note it could be useful for working with older HTML.

• We should revisit these examples once we’ve tackled CSS.
There are around 10,000 living species of birds that inhabit different ecosystems from the Arctic to the Antarctic. Man species undertake long distance annual migrations, and many more perform shorter irregular journeys.
Example image

• See [http://htmlandcssbook.com/code-samples/chapter-05/adding-images.html](http://htmlandcssbook.com/code-samples/chapter-05/adding-images.html)

• HTML:
  
  ```html
  <body>
    <img src="images/quokka.jpg" alt="Quokka (Setonix brachyurus)" /> 
  </body>
  ```

• Note how it stays the same size while you resize the page. It shows in 600x450 screen pixels.

• On my laptop, 600 px width = 50% of screen width

• Right-click on image in Chrome, Inspect, to see its dimensions.
Pixels

- Each digital image has a grid of pixels, written as width x height, for example 600x450

- Each screen has “resolution” width x height pixels
  - My old laptop: 1280x800, new one 1920x1080
  - My desktop (20” “two-page” monitor): 1920x1080
  - iPhone 5: 640x1136,…iPhone X: 1125x2436 … iPhone XS: 1125x2436
  - Samsung S7: 1440x2560, … S9 1440x2960
  - Many old phones: 854x480

- We can expect about 1000 pixels across to work with

- In particular, the 600x450 image fits on any of these

- See pp. 377-378 for more examples.
Create each image the same width and height as you would like it to appear on your website.
Sea otters hold hands when they sleep so that they don’t drift away from each other.
<figure>
  <img src="images/otters.gif"
       alt="Photograph of two sea otters floating in the water" />
  <br />
  <figcaption>Sea otters hold hands when they sleep so that they don’t drift away from each other.</figcaption>
</figure>
<figure>
    <img src="images/otters.gif"
        alt="Photograph of two sea otters floating in the water" />
    <br />
    <figcaption>Sea otters hold hands when they sleep so that they don’t drift away from each other.</figcaption>
</figure>
Sea otters hold hands when they sleep so they don't drift away from each other.
CHAPTER 6
TABLES
WHAT'S A TABLE?
```
<table>
  <tr>
    <td>15</td>
    <td>15</td>
    <td>30</td>
  </tr>
  <tr>
    <td>45</td>
    <td>60</td>
    <td>90</td>
  </tr>
</table>
```
15 15 30
45 60 45
<table>
<thead>
<tr>
<th></th>
<th>Saturday</th>
<th>Sunday</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tickets sold</td>
<td>120</td>
<td>135</td>
</tr>
</tbody>
</table>
Saturday  Sunday

Tickets sold: 120  135
### SPANNING COLUMNS

...  

<table>
<thead>
<tr>
<th></th>
<th>Monday</th>
<th>Geography</th>
<th>Math</th>
<th>Art</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Monday</td>
<td>Geography</td>
<td>Math</td>
<td>Art</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Monday</td>
<td>Geography</td>
<td>Math</td>
<td>Art</td>
<td></td>
</tr>
</tbody>
</table>

...
<table>
<thead>
<tr>
<th>Monday</th>
<th>9am</th>
<th>10am</th>
<th>11am</th>
<th>12am</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Geography</td>
<td></td>
<td>Math</td>
<td>Art</td>
</tr>
</tbody>
</table>
### SPANNING ROWS

<table>
<thead>
<tr>
<th>Time</th>
<th>Movie</th>
<th>Comedy</th>
<th>News</th>
</tr>
</thead>
<tbody>
<tr>
<td>6pm - 7pm</td>
<td>Movie</td>
<td>Comedy</td>
<td>News</td>
</tr>
<tr>
<td>7pm - 8pm</td>
<td>Sport</td>
<td>Current Affairs</td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td>Movie</td>
<td>Comedy</td>
<td>News</td>
</tr>
<tr>
<td>-----------</td>
<td>--------</td>
<td>--------</td>
<td>------------</td>
</tr>
<tr>
<td>6pm - 7pm</td>
<td>Movie</td>
<td>Comedy</td>
<td>News</td>
</tr>
<tr>
<td>7pm - 8pm</td>
<td>Movie</td>
<td>Comedy</td>
<td>News</td>
</tr>
</tbody>
</table>
CHAPTER 7
FORMS
WHY FORMS?

Amazon.com

Facebook

Facebook helps you connect and share with the people in your life.

Sign Up
It's free and always will be.

First Name:
Last Name:
Your Email:
Re-enter Email:
New Password:

I am: Select Sex: 
Birthday: Month: Day: Year: 

Why do I need to provide my birthday?

By clicking Sign Up, you agree to our Terms and that you have read and understand our Data Use Policy.

Create a Page for a celebrity, band or business.

Facebook © 2012 - English (US)
FORM CONTROLS

**Adding Text:**
- Text input (single-line)
  - Ivy
- Password input
  - *
- Text area (multi-line)
  - Enter your comments...

**Making Choices:**
- Radio buttons
  - Rock  Pop  Jazz
- Checkboxes
  - iTunes  Last.fm  Spotify

**Submitting Forms:**
- Submit buttons
- Image buttons

**Uploading Files:**
- File upload
1: User fills in form and presses button to submit info to server
2: Name of each form control sent with value user entered
3: Server processes information using programming language
HOW FORMS WORK

4: Server creates new page to send back to the browser based on info received

Thank you, Ivy!
You voted for Herbie Hancock.
username=Ivy
NAME & VALUE PAIRS

NAME

username = Ivy
NAME & VALUE PAIRS

NAME

username = Ivy

VALUE
Name-value pairs sent to server

- **HTTP GET**: in URL query string

  ```
  GET /webapp/program?username=Ivy HTTP/1.0
  ...
  ```

- **HTTP POST**: in the body of the request:

  ```
  POST /webapp/program
  ...
  ```

  encoded name-value pairs

Note: HTTP POST is the usual way to send in form data, as we will see.
Communications Diagram: simple form handling (time flows down in diagram)

1. User: requests form page via browser

3. User: fills in form
   Browser: puts user input into params in POST request

5. User: sees response

2. Server: sees GET /…form.html, returns form.html on same connection

4. Server: sees POST /…/doit.php
   Get user input from params, do requested action, compose response, return it in same connection.
<form action="http://example.com/join.php" method="get">

This is where the form controls will appear.

</form>
<form action="http://example.com/join.php">
  <input type="text" name="username" size="15" maxlength="30" />
</form>
Username:
Username:
<input type="text" name="username" size="15" maxlength="30" />

Password:
<input type="password" name="password" size="15" maxlength="30" />
Username: Ivy
Password: *******
What did you think of this gig?

Enter your comments...
What did you think of this gig?

Enter your comments...
Your favorite genre: Rock

Pop

Jazz
Please select your favorite genre:

- [ ] Rock
- [ ] Pop
- [ ] Jazz
Your favorite music service:

- <input type="checkbox" name="service" value="iTunes" checked="checked" />
  iTunes

- <input type="checkbox" name="service" value="Last.fm" />
  Last.fm

- <input type="checkbox" name="service" value="Spotify" />
  Spotify
Your favorite music service:
☑️ iTunes ☐ Last.fm ☐ Spotify
<select name="devices">
  <option value="iPod" selected="selected">iPod</option>
  <option value="radio">Radio</option>
  <option value="PC">Computer</option>
</select>
<select name="devices" size="4" multiple="multiple">
  <option value="guitar" selected="selected">Guitar</option>
  <option value="drums">Drums</option>
  <option value="keys" selected="selected">Keyboard</option>
  <option value="bass">Bass</option>
</select>
FILE INPUT BOX: we won’t need this.

```html
<form action="http://eg.com/upload.php" method="post">

<p>Upload your song in MP3 format:</p>

<input type="file" name="user-song" />

<input type="submit" value="upload" />

</form>
```
Upload your song in MP3 format:

[Input field]

[Browse...]

[Upload]
<form action="http://eg.com/email.php">

<p>Subscribe to our email list:</p>

<input type="text" name="email" />

<input type="submit" value="Subscribe" />

</form>
Subscribe to our email list:
<form action="http://eg.com/email.php">
<p>Subscribe to our email list:</p>
<input type="text" name="email" />
<input type="image" src="images/subscribe.jpg" width="100" height="20" /></form>
Subscribe to our email list:
BUTTONS: Don’t use `<button>` inside a form!

```html
<form action="http://eg.com/add.php">
  <button>
    <img src="images/add.gif" alt="add" width="10" height="20" />
  </button>
</form>
```

This submits the form, like `<input type="submit">` or `<input type="image">`. Tip from [www.w3schools.com](http://www.w3schools.com): If you use the `<button>` element in an HTML form, different browsers may submit different values. Use `<input>` to create buttons in an HTML form.

- Let’s follow this rule.
HIDDEN FORM CONTROLS
And fix pg. 162: replace <button> with <input>

```html
<form action="http://eg.com/add.php">
  <input type="image" src="images/add.gif" alt="add" width="10" height="20" />
  <input type="hidden" name="bookmark" value="lyrics" />
</form>
```
Communications Diagram: form handling
Case of form having hidden control “bookmark”

1. User: requests form page via browser

2. Server: sees GET /…form.html, returns form.html on same connection

3. User: clicks button
   Browser: puts bookmark=lyrics in GET request

4. Server: sees
   GET /add.php?bookmark=lyrics
   Gets bookmark value, does requested action, composes response, returns it in same connection.

5. User: sees response
LABELLING FORM CONTROLS: two ways...

<form action="http://eg.com/email.php">
  <label>Age:
    <input type="text" name="Age" />
  </label>
  Gender:
  <input id="female" type="radio" name="gender" value="f" />
  <label for="female">Female</label>
  <input id="male" type="radio" name="gender" value="m" />
  <label for="female">Male</label>
</form>
Age:  
Gender:  ○ Female  ○ Male
<fieldset>
  <legend>Contact details</legend>
  <label>Email:<br />
  <input type="text" name="email"></label>
  <br />
  <label>Mobile:<br />
  <input type="text" name="mobile"></label>
  <br />
  <label>Telephone:<br />
  <input type="text" name="tel"></label>
</fieldset>
Contact details

Email:

Mobile:

Telephone:
HTML5: FORM VALIDATION
Add required attribute

<label for="username">Username:</label>
<input type="text" name="username" required="required" />

<label for="password">Password:</label>
<input type="password" name="password" required="required" />

<input type="submit" value="Submit" />
Username: Ivy
Password: [blank]
Submit

Please fill out this field.

online
<label for="date">Departure date:</label>

<input type="date" name="depart" id="date" />

<input type="submit" value="Submit" />
<input type="email" name="email" />

<input type="url" name="website" />
Please enter your email address:

Please enter an email address.

Please enter your website address:

Please enter a URL.
HTML5: SEARCH INPUT

```html
<input type="search" name="search" placeholder="Enter keyword" />

<input type="submit" value="Search" />
```
online
Whenever you want to collect information from visitors you will need a form, which lives inside a `<form>` element.

Note: this is not strictly true, since it is possible to use `<input>` outside a form along with scripts that specify resulting behavior on user actions. It is true for simple PHP websites (they have no scripts).
Information from a form is sent in name/value pairs.
Each form control is given a name, and the text the user types in or the values of the options they select are sent to the server.
HTML5 introduces new form elements which make it easier for visitors to fill in forms.
# THE EVOLUTION OF HTML

<table>
<thead>
<tr>
<th>HTML 4</th>
<th>XHTML</th>
<th>HTML5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1997</td>
<td>2000</td>
<td>Work In Progress</td>
</tr>
<tr>
<td>Had some presentational elements</td>
<td>Written in XML, presentational elements removed</td>
<td>In fact, completed in Oct., 2014</td>
</tr>
</tbody>
</table>
HTML5 DOCTYPE

<!DOCTYPE html>

Older versions—common cases:

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd" />

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/xhtml/DTD/xhtml1-transitional.dtd" />
<!-- start of introduction -->
<h1>Current Exhibition</h1>
<h2>Olafur Eliasson</h2>
<!-- end of introduction -->

<!--
<a href="mailto:info@example.org">Contact</a>
-->
Water and air. So very commonplace are these substances, they hardly attraction attention and yet they vouchsafe our very existence.

Every time I view the sea I feel a sense of calming security, as if visiting my ancestral home; I embark of a voyage of seeing.

Mystery of mysteries, water and air are right there before us in the sea.
Water and air. So very commonplace are these substances, they hardly attract attention - and yet they vouchsafe our very existence.

EVERY TIME I VIEW THE SEA I FEEL A CALMING SENSE OF SECURITY, AS IF VISITING MY ANCESTRAL HOME; I EMBARK ON A VOYAGE OF SEEING.

Mystery of mysteries, water and air are right there before us in the sea.
For a one-year period from November 2010, the Marugame Genichiro-Inokuma Museum of Contemporary Art (MIMOCA) will host...

Each will showcase works by the artist thematically contextualised under the headings...

Hours: 10:00 - 18:00 (No admittance after 17:30)
FOR A ONE-YEAR PERIOD FROM NOVEMBER 2010, THE MARUGAME GENICHIRO-INOKUMA MUSEUM OF CONTEMPORARY ART (MIMOCA) WILL HOST A CYCLE OF FOUR HIROSHI SUGIMOTO EXHIBITIONS.

Each will showcase works by the artist thematically contextualized under the headings "Science," "Architecture," "History" and "Religion" so as to present a comprehensive panorama of the artist's oeuvre.

HOURS: 10:00 - 18:00 (NO ADMITTANCE AFTER 17:30)
Hiroshi Sugimoto

The dates of the ORIGIN OF ART exhibition are as follows:

- **Science**: 21 Nov – 20 Feb 2010
- **Architecture**: 6 Mar – 15 May 2010
- **History**: 29 May – 21 Aug 2010
- **Religion**: 28 Aug – 6 Nov 2010
Hiroshi Sugimoto

The dates for the ORIGIN OF ART exhibition are as follows:

- Science: 21 Nov - 20 Feb 2010/11
- History: 29 May - 21 Aug 2011
- Religion: 28 Aug - 6 Nov 2011
Timed to a single revolution of the planet around the sun at a 23.4 degrees tilt that plays out the rhythm of the seasons, this <em>Origins of Art</em> cycle is organized around four themes: <b>science</b>, <b>architecture</b>, <b>history</b> and <b>Religion</b>.
Timed to a single revolution of the planet around the sun at a 23.4 degrees tilt that plays out the rhythm of the seasons, this *Origins of Art* cycle is organized around four themes: science, architecture, history and religion.
Block vs. inline elements

- Note: Although it’s normal to put inline elements inside block elements, it’s not a good idea to try to put block elements inside inline elements.

- This distinction is not cast in stone. Inline elements can be converted to block elements and vice versa through use of the display attribute—see pg. 317.
GROUPING TEXT & ELEMENTS IN A BLOCK

```html
<div id="header">
    <img src="logo.gif" alt="Anish Kapoor" />
    <ul>
        <li><a href="index.html">Home</a></li>
        <li><a href="bio.html">Bio</a></li>
        <li><a href="works.html">Works</a></li>
        <li><a href="contact.html">Contact</a></li>
    </ul>
</div> <!-- #header -->
GROUPING TEXT & ELEMENTS IN A BLOCK

```html
<div id="header">
  <img src="logo.gif" alt="Anish Kapoor" />
  <ul>
    <li><a href="index.html">Home</a></li>
    <li><a href="bio.html">Bio</a></li>
    <li><a href="works.html">Works</a></li>
    <li><a href="contact.html">Contact</a></li>
  </ul>
</div> <!-- header -->
```
ANISH KAPOOR

- Home
- Biography
- Works
- Contact
Anish Kapoor won the Turner Prize in 1991 and exhibited at the Tate Modern gallery in London in 2003.
Anish Kapoor won the Turner Prize in 1991 and exhibited at the Tate Modern gallery in London in 2003.
<iframe width="450" height="350"
    src="http://maps.google.co.uk/maps?q=moma+new+york&amp;output=embed"
    frameborder="0"
    scrolling="0">
</iframe>

[Online page](http://maps.google.co.uk/maps?q=moma+new+york)
ESCAPE CHARACTERS

<p>&copy; 2012</p>
<p>&#169; 2012</p>

<p>The &lt;html&gt; tag</p>
<p>The &lt;html&gt; tag</p>
The `<html>` tag

The `<html>` tag
DOCTYPES tell browsers which version of HTML you are using.
You can add comments to your code between the <!-- and --> markers.
The id and class attributes allow you to identify particular elements.
The `<div>` and `<span>` elements allow you to group block-level elements and inline content together.
<iframe> cut windows into your web pages through which other pages can be displayed.
The `<meta>` tag allows you to supply all kinds of information about your web page.
Escape characters are used to include special characters in your pages such as <, >, and ©.