CS637 Class 2

We are looking at slides from HTML and CSS: Design and Build Websites 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.

WRITING LINKS

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

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>

RELATIVE URLS

SAME
reviews.html

CHILD
music/index.html
Film-Making Terms

Arc Shot
A shot in which the subject is photographed by an existing 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 existing 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
CHOOSING IMAGES FOR YOUR SITE

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.

If a page shows several photos of products or members of a team, keep them consistent.
STORING IMAGES ON YOUR SITE

Adding Images

HTML

```html
<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"/>
```

Result

Book examples: this page online

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
HEIGHT & WIDTH OF IMAGES

HTML

```html
<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
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

WHERE TO PLACE IMAGES IN YOUR CODE

<p><img src="images/bird.gif" alt="bird" width="100" height="100" /></p>

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.</p>

WHERE TO PLACE IMAGES IN YOUR CODE

<p><img src="images/bird.gif" alt="bird" width="100" height="100" /></p>

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.</p>

<em>is “inline”</em>

• 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. 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
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.

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 laptop: 1280x800
  - My desktop (21" "two-page" monitor): 1920x1080
  - iPhone 5: 640x1136...iPhone X: 2436x1125 ...iPhone 8: 1344x750
  - Samsung S4: 1080x1920, ...S8 2960x1440
  - 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.

HTML5: FIGURE & FIGURE CAPTION

```html
<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>
```

HTML5: FIGURE & FIGURE CAPTION

```html
<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>
```

RESULT

Sea otters hold hands when they sleep so that they don’t drift away from each other.
WHAT'S A TABLE?

BASIC TABLE STRUCTURE

```html
<table>
  <tr>
    <td>15</td>
    <td>15</td>
    <td>30</td>
  </tr>
  <tr>
    <td>45</td>
    <td>60</td>
    <td>90</td>
  </tr>
</table>
```

RESULT

```
15 15 30
45 60 45
```

TABLE HEADINGS

```html
<table>
  <tr>
    <th scope="row">Saturday</th>
    <th scope="row">Sunday</th>
  </tr>
  <tr>
    <td>Tickets sold</td>
    <td>120</td>
    <td>135</td>
  </tr>
</table>
```

RESULT

```
Saturday Sunday
Tickets sold: 120 135
```
SPANNING COLUMNS

```html
...<tr>
<th>Monday</th>
<td colspan="2">Geography</td>
<td>Math</td>
<td>Art</td>
</tr>
...
```

SPANNING ROWS

```html
...<tr>
<th>6pm - 7pm</th>
<td rowspan="2">Movie</td>
<td>Comedy</td>
<td>News</td>
</tr>
<tr>
<th>7pm - 8pm</th>
<td>Sport</td>
<td>Current Affairs</td>
</tr>
```
WHY FORMS?

1: User fills in form and presses button to submit info to server

HOW FORMS WORK

VOTE FOR YOUR FAVORITE JAZZ MUSICIAN OF ALL TIME

Username: ivy
I vote for: Ella Fitzgerald, Herbie Hancock, John Coltrane, Miles Davis, Thelonious Monk

3: Server processes information using programming language

HOW FORMS WORK

FORM CONTROLS

ADDING TEXT:
- Text input (single-line)
- Password input
- Text area (multi-line)

MAKING CHOICES:
- Radio buttons
- Checkboxes
- Drop-down boxes

SUBMITTING FORMS:
- Submit buttons
- Image buttons

UPLOADING FILES:
- File upload

2: Name of each form control sent with value user entered

HOW FORMS WORK

VOTE FOR YOUR FAVORITE JAZZ MUSICIAN OF ALL TIME

Username: ivy
I vote for: Ella Fitzgerald, Herbie Hancock, John Coltrane, Miles Davis, Thelonious Monk

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

Thank you, ivy!
You voted for Herbie Hancock.
Name-value pairs sent to server

- HTTP GET: in URL query string
  
  GET /webapp/program?username=Ivy HTTP/1.0
  ... headers

- HTTP POST: in the body of the request:
  
  POST /webapp/program
  ... headers
  ... encoded name-value pairs

Note: HTTP POST is the usual way to send in form data, as we will see.

**FORM STRUCTURE**

```html
<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>
**PASSWORD**

```html
<p>Username: 
<input type="text" name="username" size="15" maxlength="30" />
</p>

<p>Password: 
<input type="password" name="password" size="15" maxlength="30" />
</p>
```

**TEXTAREA**

```html
<p>What did you think of this gig?</p>
<textarea name="comments" cols="20" rows="4">
Enter your comments...
</textarea>
```

**RADIO BUTTON**

```html
<p>Your favorite genre:<br />
<input type="radio" name="genre" value="rock" checked="checked" /> Rock
<input type="radio" name="genre" value="pop" /> Pop
<input type="radio" name="genre" value="jazz" /> Jazz
</p>
```
Your favorite genre:

- [ ] Rock
- [ ] Pop
- [ ] Jazz

Your favorite music service:

- [ ] iTunes
- [ ] Last.fm
- [ ] Spotify
Your favorite music service:<br />
<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
</p>

Choose your devices:

<select name="devices">
  <option value="iPod" selected="selected">iPod</option>
  <option value="radio">Radio</option>
  <option value="PC">Computer</option>
</select>
DROP DOWN LIST BOX

```html
<select name="devices">
    <option value="iPod" selected="selected">iPod</option>
    <option value="radio">Radio</option>
    <option value="PC">Computer</option>
</select>
```

DROP DOWN LIST BOX

```html
<select name="devices">
    <option value="iPod" selected="selected">iPod</option>
    <option value="radio">Radio</option>
    <option value="PC">Computer</option>
</select>
```

DROP DOWN LIST BOX

```html
<select name="devices">
    <option value="iPod" selected="selected">iPod</option>
    <option value="radio">Radio</option>
    <option value="PC">Computer</option>
</select>
```

MULTIPLE SELECT BOX, fixed (red)

```html
<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>
```
<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>

FILE INPUT BOX: FYI, we won't need this.

<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>

FILE INPUT BOX

<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>

FILE INPUT BOX

<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>
<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>

---

<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>

---

<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>

---

<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>

---

<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>
<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>

**BUTTONS: Don’t use <button>!**

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

Tip from www.w3schools.com:

**Note:** 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">
  <button> <img src="images/add.gif" alt="add" width="10" height="20" /></button>
  <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

Gets bookmark value, does requested action, composes response, returns it in same connection.

5. User: sees response

RESULT

LABELLING FORM CONTROLS: two ways...

HTML

```html
<form action="http://eg.com/email.php">
  <label>Age: </label>
  <input type="text" name="Age" />
</form>

<label>Gender:
  <input id="female" type="radio" name="gender" value="f" />
  Female
  <input id="male" type="radio" name="gender" value="m" />
  Male
</label>
```

LABELLING FORM CONTROLS

HTML

```html
<form action="http://eg.com/email.php">
  <label>Age: </label>
  <input type="text" name="Age" />
</form>

<label>Gender:
  <input id="female" type="radio" name="gender" value="f" />
  Female
  <input id="male" type="radio" name="gender" value="m" />
  Male
</label>
```

LABELLING FORM CONTROLS

HTML

```html
<form action="http://eg.com/email.php">
  <label>Age: </label>
  <input type="text" name="Age" />
</form>

<label>Gender:
  <input id="female" type="radio" name="gender" value="f" />
  Female</label>
  <input id="male" type="radio" name="gender" value="m" />
  Male
</label>
```

LABELLING FORM CONTROLS

HTML

```html
<form action="http://eg.com/email.php">
  <label>Age: </label>
  <input type="text" name="Age" />
</form>

<label>Gender:
  <input id="female" type="radio" name="gender" value="f" />
  Female
  <input id="male" type="radio" name="gender" value="m" />
  Male
</label>
```
GROUPING FORM ELEMENTS

```html
<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>
```

GROUPING FORM ELEMENTS

```
<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>
```

GROUPING FORM ELEMENTS

```
<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>
```

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" />
```

RESULT
**HTML5: DATE INPUT**

```html
<label for="date">Departure date:</label>
<input type="date" name="depart" id="date" />
<input type="submit" value="Submit" />
```

**RESULT**

![Date Input Example](image)

**HTML5: EMAIL & URL INPUT**

```html
<input type="email" name="email" />
<input type="url" name="website" />
```

**RESULT**

![Email and URL Input Example](image)

**HTML5: SEARCH INPUT**

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

**RESULT**

![Search Input Example](image)
Whenever you want to collect information from visitors you will need a form, which lives inside a `<form>` element.

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.