HTML (Advanced)

Outline

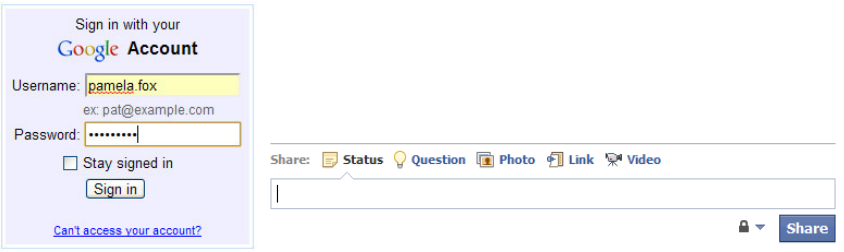
- Forms
- Tables
- iFrames
- Audio/Video

Forms

Forms

The web started as a way for people to post static information, but now is largely about "user-contributed content." All that content means a lot of forms!

Users see form on social sites, login screens, banking sites, and more:



Form element

A form consists of a `form` element with multiple input controls inside of it that help retrieve user data in some way, and attributes that specify how a server should handle the form.

The example below asks for a search query, and then sends the query to the Google server once submitted.

```
<form action="http://www.google.com/search" method="get">
 <label>Google:<input type="search" name="q">
 <input type="submit" value="Search"> </label>
</form>
```

Google: Search...

Input element

The `input` element is used for many different types of user input, depending on value of the `type` attribute. It defaults to a simple 1-line text input.

The "name" attribute gives the server a way to identify that piece of attribute. The "value" attribute can be set to specify a default value. The "placeholder" provides a hint to the user.

```
<input type="text" name="animal" placeholder="e.g. fox">
```

Label element

The `label` element is used to caption a form control, so users know what they should put in it.

```
<label> Animal:
<input type="text" name="animal" placeholder="e.g. fox">
</label>
```

Animal e.g. fox

Checkboxes

The `input` element can be used for letting users pick multiple in a list or simply indicate yes/no, using `type="checkbox"`.

```
<label> Extra cheese?
<input type="checkbox" name="extracheese"/> </label>
```

Extra cheese? ☐

```
Pizza Toppings:
<label> <input type="checkbox"/> Bacon </label>
<label> <input type="checkbox"/> Extra Cheese </label>
<label> <input type="checkbox"/> Onion </label>
```

Pizza Toppings: ☐ Bacon ☐ Extra Cheese ☐ Onion ☐ Mushroom

Radio buttons

Similarly, the `input` element can be used for creating a group of radio buttons, for letting users pick one in a list, with `type="radio"`.

```
Pizza Size:
<label>
  <input type="radio" name="size" value="small"/>Small
</label>
<label>
  <input type="radio" name="size" value="medium"/>Medium
</label>
<label>
  <input type="radio" name="size" value="large"/>Large
</label>
```

Pizza Size: ○ Small ○ Medium ○ Large

Range sliders

The `input` element can be used for users to pick a value in a range, using `type="range"`.

```
<label> Cheesiness:
<input type="range" min="0" max="10" name="cheesiness"/>
</label>
```

Cheesiness:

*New, only supported in modern browsers.

The select element

The `select` element used with the `option` element produces a dropdown of options for the user to pick from.

```
<label>Crust type:
<select name="crust">
  <option value="thin">Thin</option>
  <option value="thick">Thick</option>
  <option value="cheese">Cheese</option>
</select>
</label>
```

Crust type: Thin

Textarea element

To let the user specify multiple lines of text, use the `textarea` element, and specify `cols` or `rows` to specify a specific size.

```
<label>Delivery instructions: <br>
<textarea></textarea></label>
```

Delivery instructions:

Button element

The `button` element is most commonly used to submit a form to the server, but can also be used for resetting a form or other scripted actions.

```
<button type="submit">Submit order</button>
```

Submit order

Form processing

A form is processed by a server, which typically takes the inputs and stores them in a database.

If you don't have a server, you can also have all the form contents sent to the email address of your choice, using extra attributes in the `form` element:

```
<form action="mailto:admin@example.com"
  enctype="text/plain" method="post">
  <input name="info" type="text"/>
  <button type="submit">Send Mail</button>
</form>
```

Tables

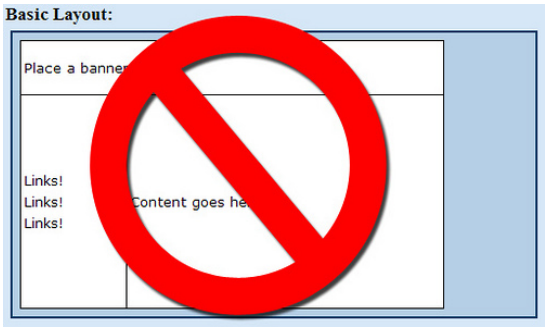
Tables

Tables are useful for displaying tabular data on the web, like research study results.

Table name										
	col 01	col 02	col 03	col 04	col 05	col 06	col 07	col 08	col 09	col 10
ROW 1	1.0	11.0	21.0	31.0	41.0	51.0	61.0	71.0	81.0	91.0
ROW 2	2.0	12.0	22.0	32.0	42.0	52.0	62.0	72.0	82.0	92.0
ROW 3	3.0	13.0	23.0	33.0	43.0	53.0	63.0	73.0	83.0	93.0
ROW 4	4.0	14.0	24.0	34.0	44.0	54.0	64.0	74.0	84.0	94.0
ROW 5	5.0	15.0	25.0	35.0	45.0	55.0	65.0	75.0	85.0	95.0
ROW 6	6.0	16.0	26.0	36.0	46.0	56.0	66.0	76.0	86.0	96.0
ROW 7	7.0	17.0	27.0	37.0	47.0	57.0	67.0	77.0	87.0	97.0
ROW 8	8.0	18.0	28.0	38.0	48.0	58.0	68.0	78.0	88.0	98.0
ROW 9	9.0	19.0	29.0	39.0	49.0	59.0	69.0	79.0	89.0	99.0
ROW 10	10.0	20.0	30.0	40.0	50.0	60.0	70.0	80.0	90.0	100.0

Tables: Not for Layout

Before we had better techniques, tables were often used to layout a webpage, but this is now strongly discouraged.



We will discuss better ways to do layout when we cover CSS.

The table element

The TABLE element contains all the elements that make up a table - the header, rows, and columns. This example shows a table of monthly savings.

```
<table>
  <thead>
   <tr>
    <th>Month </th>
    <th>Savings </th>
   </tr>
  <tbody>
   <tr>
    <td>January </td>
    <td>$100 </td>
   </tr>
</table>
```

Month	Savings
January	$100

The table header

Most tables start with a header row with the names of the columns. The `thead` element starts the header area, the `tr` element creates a row, and `th` elements create cells in the row.

```
<table>
  <thead>
   <tr>
    <th>Month </th>
    <th>Savings </th>
    </tr>
    ....
</table>
```

Month	Savings

The table body

Tables can consist of any number of data rows after the header. The `tbody` element begins the body (data) area, the same `tr` element creates a row, and `td` elements create data cells in each row.

```
<table>
  ....
  <tbody>
   <tr>
    <td>January </td>
    <td>$100 </td>
   </tr>
</table>
```

Month	Savings
January	$100
February	$50

The table caption

One way to provide additional info about the purpose and use of a table is via the `caption` element, which goes right under the `table` element.

```
<table>
  <caption>Monthly Savings (in USD) for 2010</caption>
  ...
</table>
```

Monthly Savings (in USD) for 2010

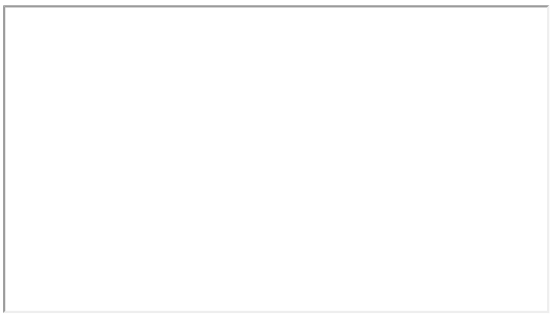
Month	Savings
January	$100
February	$50

Iframes

The iframe element

You can use HTML to embed other webpages in your own page using an `iframe` element pointing at a URL:

```
<iframe src="https://www.google.com"
  width="500" height="300"></iframe>
```



Iframe embeds

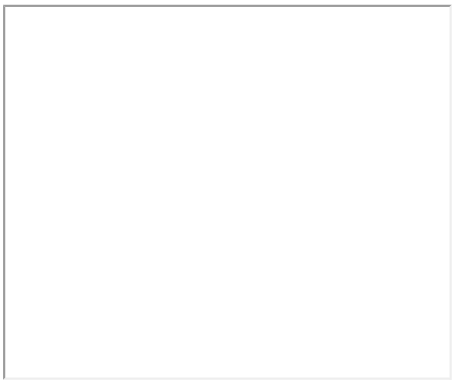
The `iframe` element makes it easy for websites to become embeddable on other websites, and for users to share content that they've created on a site.

Most websites don't naturally look good when embedded at small sizes, so websites create special versions that are designed for embedability, and the provide users with those URLs.

Iframe embeds: Docs

In its "Publish" dialog, Google docs provides this HTML for embedding a presentation on your site:

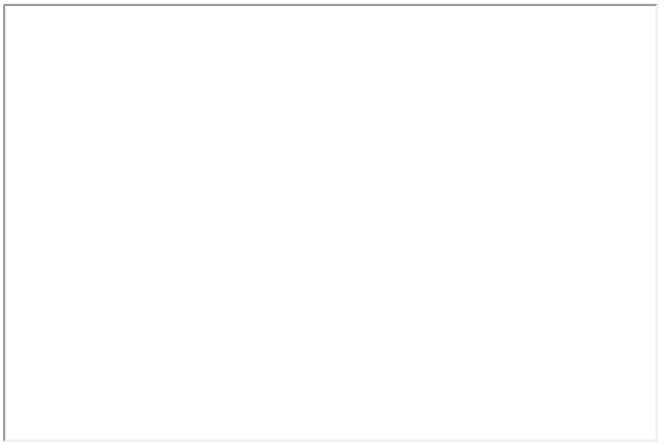
```
<iframe src="https://docs.google.com/present/embed?id=dggjrx3s_2119zdj9k4cf"
  width="410" height="342"></iframe>
```



Iframe embeds: Calendar

In settings, Google Calendar provides this HTML for embedding a calendar:

```
<iframe src="http://www.google.com/calendar/embed?src=developer-calendar%40google.com"
  width="600" height="400"></iframe>
```



Multimedia

The object element

You can use `object` to embed non-HTML content in your page, like a quicktime movie or a flash file. Typically, users need a "plug-in" to view that content.

The `object` tag should have one or more `param` child tags that give info to the browser on what to embed.

```
<object width="300" height="150">
  <param name="movie" value="beatingheart.swf"/>
</object>
```

The embed element

The `embed` tag is similar to the `object` tag, but is sometimes used in cases where the `object` tag doesn't work.

The `embed` tag should specify `src` and `type` attributes.

```
<embed type="application/x-shockwave-flash"
 src="beatingheart.swf"/>
```

object and embed

The `object` tag can include content that is executed if the browser doesn't understand the object tag. This content could be text, another `object` tag, or an `embed` tag.

```
<object classid="clsid:02BF25D5-8C17-4B23-BC80-D3488ABDDC6B"
 codebase="http://www.apple.com/qtactivex/qtplugin.cab"
 height="400" width="500">
 <param name="src" value="rbc.mov" >
 <param name="autoplay" value="true" >

 <embed
  type="video/quicktime"
  pluginspage="http://www.apple.com/quicktime/download/"
  height="400" width="400"
  src="rbc.mov" autoplay="true"
 />
</object>
```

The object element (Example)

The code from the previous slide produces:

Embeds: Youtube

Youtube gives user an embed code that includes an `object` and an `embed` tag.

```
<object width="480" height="385">
  <param name="movie"
     value="http://www.youtube.com/v/BHtGcZ9XqjY"/>
  <param name="allowFullScreen"
     value="true"/>
  <param name="allowscriptaccess"
     value="always"/>

  <embed src="http://www.youtube.com/v/BHtGcZ9XqjY"
         type="application/x-shockwave-flash"
         allowscriptaccess="always"
         allowfullscreen="true" width="480" height="385"/>

</object>
```

Embeds: Youtube (Example)

The code from the previous slide produces:

Multimedia in HTML5

In the past, developers had to use browser plugins to embed multimedia on their page.

In HTML5, you can embed audio or video using native HTML tags, and if the browser supports the tags, it will give users controls to play the file.

The `audio` and `video` tags are both very new, so they only work in the most recent versions of modern browsers.

Video file formats

A video file, like an ".avi" file, is really a container for multiple related files that describe a video, like:

- a video track
- one or more audio tracks with synchronization markers
- metadata (title, album art, etc).

The most popular video formats are:

- MPEG4: .mp4, .m4v
- Flash Video: .flv
- Audio Video Interleave: .avi
- Ogg: .ogv
- WebM: .webm

Video codecs

A video track is compressed and stored in a particular way, and a "codec" describes how to turn the stored data into the series of images that you see.

The most relevant video codecs are:

- H.264 (aka MPEG-4 Advanced Video Coding): Patent-encumbered.
- Theora: Royalty-free.
- VP8: Patented, but licensed royalty-free.

Audio codecs

An audio track is also compressed, stored, and decoded according to a codec. Most audio codecs support two channels of sound, some support more.

The most relevant audio codecs are:

- MP3: Patent-encumbered.
- AAC (Advanced Audio Coding): Patent-encumbered. Used in Apple products.
- Vorbis: Not patented.

Browser support

Each browser supports a different combo of formats and codecs, due to patents issues. This is the current situation:

Codecs/container	IE	Firefox	Safari	Chrome	Opera	iPhone	Android
Theora+Vorbis+Ogg	·	3.5+	·	5.0+	10.5+	·	·
H.264+AAC+MP4	·	·	3.0+	5.0+	·	3.0+	2.0+
WebM	·	·	·	6.0+	10.6+	·	·

The Theora/Ogg and WebM formats are both open-source and patent-free, so they are the best to use if you don't want to worry about licensing.

Creating multimedia files

There are various command-line utilities for turning uncompressed video into web-ready video. The most user-friendly option is FireFogg, for Firefox.

The video element

The `video` element embeds a video player for a particular video file.

You can use these attributes to customize the player: `poster`, `preload`, `autoplay`, `loop`, and `controls`.

```
<video src="chrome_japan.ogv" controls width="200"></video>
```

Multiple media sources

As we saw, different browsers support different formats. Thankfully, HTML5 lets us specify multiple sources for the `video` and `audio` elements, so browsers can use whichever works for them.

```
<video height="200" controls="">
  <source src="chrome-japan.webm" type="video/webm"/>
  <source src="chrome-japan.mp4" type="video/mp4"/>
  <source src="chrome_japan.ogv" type="video/ogg"/>
</video>
```

Fallback options

There are some browsers that don't support these new elements. In that case, you can provide a fallback option via a browser plug-in, like Java or Flash, and put the fallback inside the element. For example, here's a Flash fallback:

```
<video src="video.ogv" controls>
    <object data="flvplayer.swf" type="application/x-shockwave-flash">
      <param value="flvplayer.swf" name="movie"/>
    </object>
</video>
```

The audio element

The `audio` element is used for embedding an audio player inside a page for a particular audio file.

You can use various attributes to customize the player: `preload`, `autoplay`, `loop`, and `controls`.

```
<audio src="Tromboon-sample.ogg"
  controls="true" autobuffer="true"></audio>
```