HTML

HTML

- HyperText Markup Language

- Not a programming language

- Stored in text files (just like Perl)

A basic page

```html
<html>
  <head>
    <title>My web page title</title>
  </head>
  <body>
    Your HTML content here
  </body>
</html>
```

A kosher page

```html
<?xml version="1.0" encoding="utf-8"?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://
www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>

    <title>An XHTML 1.0 Strict standard template</title>

</head>

<body>

    <p>... Your HTML content here ...</p>

</body>

</html>
```

Why use web Standards?

- Accessibility
  - To robots
  - To people
- Stability

<Tags />

Most tags open and close
- Tags must be nested properly

| Right | ```
<strong>
   <em>
       Strong and emphasis
   </em>
</strong>
``` | Wrong | ```
<strong>
   <em>
       Strong and emphasis
   </strong>
</em>
``` |
|---|---|---|---|

- Some tags stand alone

  `<br /> <hr />`

- Some tags take attributes

  `<img alt="My dog" src="rover.gif"/>`

  `<a href="theonion.com">The Onion</a>`

- Elements consist of start and end tags flanking content

XHTML tags

http://www.w3schools.com/tags/

Text tags

- Heading tag

`<H1>`This is a top level heading`</H1>`
`<H6>`This is the bottom level heading`</H6>`

- Paragraph tag

`<p>`This is definitely a paragraph`</p>`

- Line Break

This is just two lines`<br />` With a hard break.

- Emphasis and Strong

That's `<em>`*exactly*`</em>` what I mean - I am `<strong>`**sick**`</strong>` of this slide

- Comment Tag

`<!-- This is a comment. You won't see this on the web-->`

Tables

```
    <table border="1">

<tr>
<th> Column 1 heading</th>
<th> Column 2 heading</th>
<th> Column 3 heading</th>
</tr>

<tr>
<td>Row 2, cell 1</td>
<td colspan="2">Row 2, cell 2, also spanning Row 2, cell 3</td>
</tr>

<tr>
<td rowspan="2">Row 3, cell 1, also spanning Row 4, cell 1</td>
<td>Row 3, cell 2</td>
<td>Row 3, cell 3</td>
</tr>

<tr>
<td>Row 4, cell 2</td>
<td>Row 4, cell 3</td>
</tr>

</table>
```

output:

Column 1 heading	Column 2 heading	Column 3 heading
Row 2, cell 1	Row 2, cell 2, also spanning Row 2, cell 3	
Row 3, cell 1, also spanning Row 4, cell 1	Row 3, cell 2	Row 3, cell 3
	Row 4, cell 2	Row 4, cell 3

http://htmldog.com/guides/htmlintermediate/tables/

Lists

```
<ol>
    <li>First things first</li>
    <ul>
        <li>Who you know</li>
    </ul>
    <li>Not</li>
    <ul>
        <li>What you know</li>
        <li>What you can do with it</li>
    </ul>
</ol>
```

output:

1. First things first
   - Who you know
2. Not
   - What you know
   - What you can do with it

Links

- Relative

```
<a href="myDirectory/index.html">Go down a directory</a>
<a href="../index.html">Go up a directory</a>
```

- Absolute

```
<a href="/">Go to the root</a>
<a href="http://nytimes.com">Go to the NY Times</a>
```

- Anchors

```
<a href="#theEnd">Go to the end</a>
<h1 id="theEnd">This is the end</h1>
```

Images

```
<img src="images/Early.png" alt="An Appalachian Mud Squid">
```

Forms

```
<form name="input" action="html_form_submit.pl" method="post">
```

- POST vs GET

  GET   = Data is in the URL

  POST = Data is in the message body

Text Fields

```
<form name="input" action="handleMyForm.pl" method="get">
    First name:
    <input type="text" name="firstname" />
    <br/>
    Last name:
    <input type="text" name="lastname" />
    <input type="submit" value="Submit" />
</form>
```

output:

First name: [   ]
Last name: [   ] (Submit)

Radio Buttons

```
<form name="input" action="handleMyForm.pl" method="get">
    <input type="radio" name="sex" value="male"/> Male
    <br/>
    <input type="radio" name="sex" value="female"/> Female
    <br/>
    <input type="submit" value="Submit" />
</form>
```
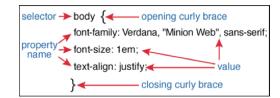
output:

○ Male
○ Female
Submit

xHTML + CSS = Web

Cascading Style Sheets

- Help separate **content** from **appearance**

  - One style sheet can be applied to hundreds of web pages

  - Change styles in just one location

How CSS works

- Statements consist of

  - Selectors

  - Declarations

    - Properties: Values (units)

```
selector → body {← opening curly brace
              font-family: Verdana, "Minion Web", sans-serif;
property → font-size: 1em;
  name
              text-align: justify; ← value
          } ← closing curly brace
```

http://westciv.com/wiki/CSS_Guide:_How_do_style_sheets_work

CSS: Where do I put it?

- Embedded in the <head> of each page

  `<head><style type="text/css"> </style></head>`

- Linked in the <head>

  Advantages: templating, speed

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

- Inline (avoid this)

  `<p style="color: red">text</p>`

CSS Selectors

- HTML selectors - raw tags in the style sheet)
- Class selectors

  use .className in style sheet
  use class="className" in HTML

- ID selectors

  use #idName in style sheet
  use id="idName" in HTML

Divs and Spans

- Divs

    - Use <div id="myDiv"> </div> to define block elements. Useful for both formatting and positioning.

    - The id is unique. It refers to one element

- Spans

    - Use when you want to apply a class to some text inline

    - This is my sequence

        ```
        <span class="dna">ACTGATCTAGCT</span>
        ```

BlueprintCSS

CSS framework

- grid

- "sensible typography"

- stylesheet for printing

Do Not Reinvent the Wheel

http://www.freecsstemplates.org

Where does my website go?

- On Mac OS X

  - Personal web: ~/Sites

  - Main web: /Library/Webserver/Documents

- Linux: /var/www/html or /var/apache2/htdocs

- XP Home: C:\Program Files\ApacheGroup \Apache\htdocs

- Could be elsewhere. Don't give up!

Naming your html files

- .html .htm

- Why index.html is special
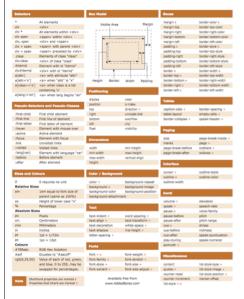
Resource: HTML

- HTML Dog

  http://htmldog.com

- W3C tags

  http://www.w3schools.com/tags

Resources: CSS

Cheat sheet:
http://www.addedbytes.com/download/css-cheat-sheet-v2/pdf/

CSS tutorial
http://westciv.com/wiki/Main_Page

Two column style sheet and tutorial
http://www.456bereastreet.com/lab/
developing_with_web_standards/csslayout/2-col/

Tools of the Trade

- Web Developer Plugin for Firefox

- CSS editors

  - MacRabbit CSSEdit

  - SimpleCSS

  - TopStyle (Windows

Scraping

- We can parse web pages like any other text.

```perl
#!/usr/bin/perl
# A quick script to parse PFAM search results
use strict;
use warnings;

my $search = shift;
system ("wget -q -O pfam_search.txt 'http://pfam.sanger.ac.uk/
search/keyword?query=" . $search . "'");

open (FILE, "<", "pfam_search.txt") or die "Cannot open file: $!
\n";

# create a hash of the ids in our web page
# We use a hash as an easy way to eliminate duplicates
my %pfam_ids;
while (my $line = <FILE>) {
    if ($line =~ /(PF\d+)/) {
        $pfam_ids{$1}++;
    }
}

# Print the IDs on one line
print join ("\t", $search, sort keys %pfam_ids), "\n";
```