HTML & CSS

Norwalk Community College
Taurus Wright
06/05/13 – 07/10/13

Get Exercise Files

- Goto http://tauruswright.tripod.com
- Click the 'HTML' link
- Click the exercises link for today's class

Internet Explorer

- Choose Open
- Click Extract All Files
- Browse to the Desktop
  - Click OK
- Click Extract

Firefox

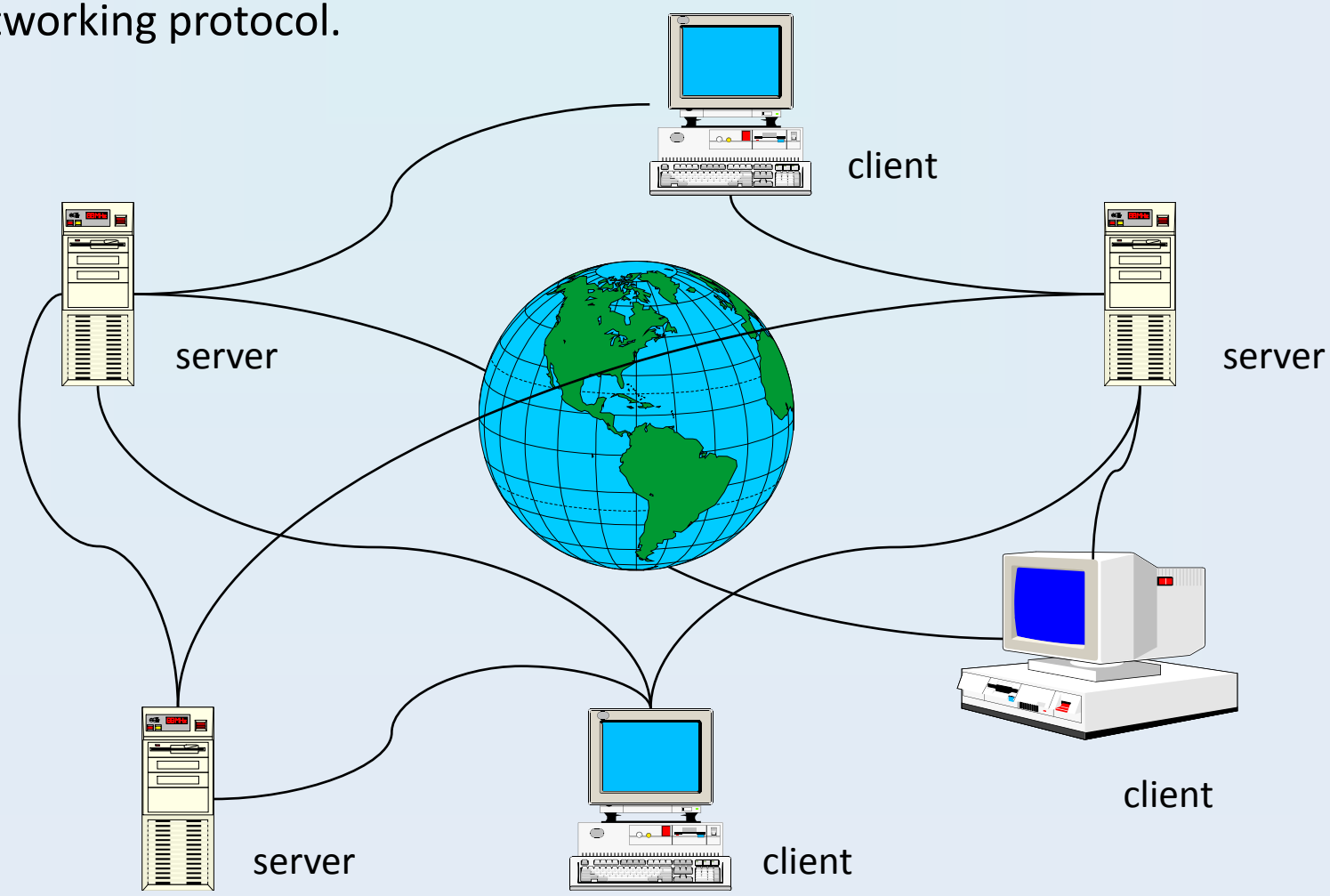
- Choose Open with Windows Explorer
- Click Extract All Files
- Browse to the Desktop
  - Click OK
- Click Extract

Agenda

- Getting Started with HTML
- Images & Linking & bears…Oh my!
- We've got Style → CSS
- Layout & Positioning…'nuf said
- Communication is Key→ Forms
- Wrapping it up with a Big Pretty Bow

The Internet

- World-wide network of computers connected by the TCP/IP networking protocol.



HTML

- Hypertext Markup Language
  - Tim Berners-Lee - 1990
  - Contain “tags” = “markups”
  - Non-platform specific files
  - Simple text files
    - create using any text editor
- Example
  <h1>Welcome to the Home Page</h1>

What is HTML?

- HTML is a language for describing web pages.
- HTML stands for **H**yper **T**ext **M**arkup **L**anguage
- HTML is not a programming language, it is a **markup language**
- A markup language is a set of **markup tags**
- HTML uses **markup tags** to describe web pages
- HTML markup tags **structure** web pages

http://www.w3schools.com/html/html_intro.asp

HTML Documents = Web Pages

- HTML documents **describe web pages**
- HTML documents **contain HTML tags** and plain text
- HTML documents are also **called web pages**

http://www.w3schools.com/html/html_intro.asp

Web Browsers

- **Internet Explorer, Firefox, Safari**, etc. are web browsers
- Web browsers **read** HTML documents
- Web browsers **display** HTML documents as **web pages**.
- The web browser uses HTML tags to **interpret page content**.

Creating HTML Documents

- To create HTML documents/web pages you need…
  - A text editor
  - A web browser
- If you want to be fancier you can use
  - A WYSIWYG HTML editor
    - A.K.A. Dreamweaver, MS Expressions
- .HTM or .HTML File Extension?
  - You can use either . There is no difference.

HTML Tags

- HTML markup tags are usually called HTML tags
  - Sometimes they are called elements
- HTML tags are keywords surrounded by **angle brackets** like <html>
- HTML tags normally **come in pairs** like and
- The first tag in a pair is the **start tag,** the second tag is the **end tag**
- Start and end tags are also called **opening tags** and **closing tags**

http://www.w3schools.com/html/html_intro.asp

Building Blocks of HTML Syntax

- <…>
- </…>
- =
- “….”
- Opening - <tags>, <elements>
- Closing – </tags>, </elements>
- Attributes=“Values”
- <tag attribute=“value”>Blah blah blah</tag>

Global Structure

```
<html>
   <head>
        <title>In title bar</title>
   </head>
   <body>
   </body>
</html>
```

Nesting

Headings

- Block level element
- Bold and size change
- <h1> </h1> thru <h6> </h6>

Paragraphs

- Paragraphs

- Line breaks

- Block quotes
  <blockquote> </blockquote>

HTML Attributes

- HTML elements can have **attributes**
- Attributes provide **additional information** about an element
- Attributes are always specified in **the start tag**
- Attributes come in **name/value pairs** like: name="value”
- Values are always **enclosed in quotes** “…”

Horizontal Rule

- Visual divider
- <hr /> → empty tag
- Attributes…
  - width = distance across page
  - size = thickness
  - color = change color
  - align = alignment on page

Colors

- Hexadecimal: a base 16 method of counting used to describe data to your computer. It uses values …
- 0 thru 9, A thru F
  (zero is the lowest, F is the greatest)
- Your monitors are RGB devices. R=red, G=green, B=blue Each color receive 2 digits of data in that order. (i.e.: FF0000 is the hexadecimal name for red because
  R=FF, G=00, B=00)
- Web safe colors: the 216 colors that are the same on the 3 major computer platforms, (PC, Mac, and Unix). The consist of the the values, 00, 33, 66, 99, CC, FF
- If all six digits are the same you get a shade of gray
- When you define a hexadecimal color it is preceded by a pound '#' sign
- DevX: Project Cool VisiBone

Back

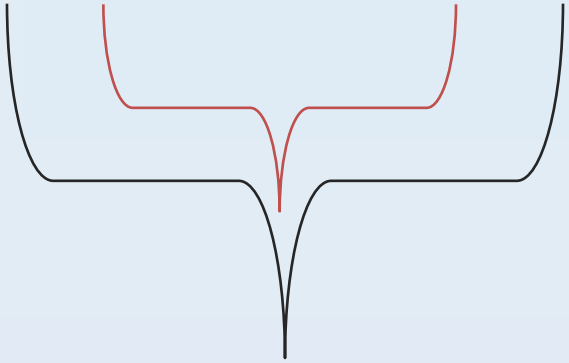
Nested Tags

Improperly Nested

****HTML 4.01**** is fun to learn

Properly Nested

****HTML 4.01**** is fun to learn



Back

Lists

- Ordered →
- Unordered →
- List items →
- Example:

```
<ol>
<li>apples</li>
<li>oranges</li>
</ol>
```

Notate Code

- Comments
- Are not visible
- Example:
  <!-- your text here -->

Character Entities

- Symbols and special characters
- Examples…
  -   ➔ non-breaking space
  - © ➔ copyright
  - | ➔ vertical bar
- ISO-Latin 1 References
  - http://www.htmlhelp.com/reference/charset/
  - http://www.w3schools.com/tags/ref_entities.asp

Formatting Elements

<abbr> (Abbr.)

<address> (*Address*)

<cite> (*Citation)*

 (*Emphasized text)*

<samp> (Sample text)

<sub> (Subscript)

<acronym> (Acronym)

<blockquote> (Blockquote)

<code> (Computer code text)

<q> (“Quote text”)

 (**Strong text)**

<sup> (Superscript)

Deprecated Tags and Attributes

- Deprecated tags are tags that will not be supported in future versions of HTML.
  - This is very important for XHTML, HTML5 and validating
- The following tags and attributes should be avoided:

Tags	Description
	Deprecated. Defines centered content
<font> & <basefont>	Deprecated. Defines HTML fonts
<s> & <strike>	Deprecated. Defines strikethrough text
<u>	Deprecated. Defines underlined text
Attributes	**Description**
align	Deprecated. Defines the alignment of text
bgcolor	Deprecated. Defines the background color
color	Deprecated. Defines the text color

HTML Attributes

- HTML elements can have **attributes**
- Attributes provide **additional information** about an element
- Attributes are always specified in **the start tag**
- Attributes come in **name/value pairs** like: name="value”
- Values are always **enclosed in quotes** “…”

HTML Hyperlinks (Links)

- A hyperlink (or link) is a word, group of words, or image that you can click on to jump to a new document or a new section within the current document.
- The cursor will turn into a little hand when you rollover a link.
- Links are specified in HTML using the <a> tag.
- The <a> tag can be used to:
  - To create a link to another document, by using the href attribute
  - To create a bookmark inside a document, by using the name or id attributes

Hyperlinks

- href = link destination
  - local = another page → path to file
  - external = another site → URL of site
  - anchor = specific location on page → name of anchor
  - email = opens email software with address filled in → email address
  - File = a document or image → path to file

<a> Tag

- Syntax → *Link trigger*
- Common attributes
  - href → hyperlink reference
  - target → where to open the reference
    - _top, _parent, _blank, _self
  - name → to name bookmarked location

Remote Links

- Links to another site
  href="http://www.site.com"
- Open in new browser window
  - Target links
    target="*targetname*"
    - _blank = new browser window
      target="_blank"

Email Links

- Opens user's default email program
- Fills in the "To:" field
- mailto:*email@address.com*

Example:

Creating Email Links with Default Content

- To create mailto links with default content
  - Use a '?' at the end of the specified address.
- For additional default fields
  - Use the ampersand '&' between field names

Example

Contact Us

Anchors

- Links to a location on a page
- 2 steps…

1.
   Names the location on the page
2. Trigger
   Links to the anchor

Images

- Images are defined with the **<img>** tag.
- The **<img>** tag is empty, which means that it has no closing tag.
- To display an image you need to use the **src** attribute.
- The **value** of the src attribute is the URL of the image you want to display.
- **Syntax for defining an image:**
  -
- Common Attributes
  - src →source
  - alt → alternate text

Image Links

- wrap link around image
- Example:
  <img src="…">
- Remove image border
  <img src="…" border="0">

Image Maps

- An image that links to more than one destination
  - Creates hotspots = clickable regions
- <map name="…"> </map>
  - <area> = to determine shape, coordinates, and destination
    - shape attribute
      - rect ➔ default, upper-left and bottom-right points
      - circle ➔ radius, center and edge points
      - poly ➔ points that create an outline
    - coords = x,y coordinates for indicated points
    - href = destination
- usemap attribute in <img … usemap="#..."">

CSS Introduction

- Style Sheet: any rule or sequence of rules that affect the appearance of a document
- Cascading Style Sheet (CSS)
  - A core technology of HTML,
  - Controlling the design and layout of Web documents
- Can control a document in ways that HTML can’t.

CSS Advantages

- Separate structure from design
- Make changes that HTML can’t
  - Remove underline from hyperlinks
  - Change line spacing
  - Give any element a background/border
  - Indent any element

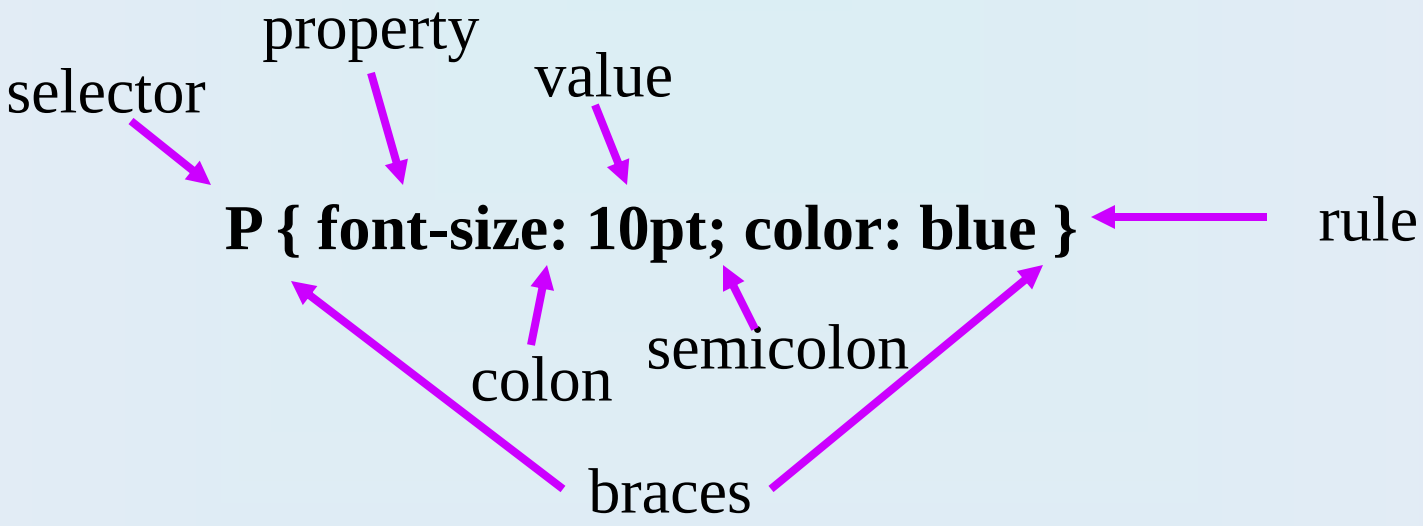
Style Sheet Types

Inline	Inserted directly into any HTML tag by using the style attribute <p style="color: red">
Embedded	Placed inside the head section of an HTML document using the style tag <style> P {color:red} </style>
Linked	An external .css document accessed using the link tag <link rel= type= href="name.css">

Building Blocks of CSS Syntax

- {…}
- :
- ;
- Blah-blah
- Selectors (element, class, ID)
- Properties:values;
- Selector {property:value;}

Style Sheet Syntax



Rule: the combination of a selector and declaration

Selector: item that is being styled

Declaration: the properties and values

- **Properties must be separated with a semicolon**
- **The amount of white space you insert doesn't matter**
- **The case (usually) doesn't matter**

Creating Rules - Embedded

1. Type style tags in head section
   <style > </style>
2. Type rules within style tags
   p {color: red;}
   a {text-decoration: none;}
3. Example

   <style type="text/css">

   p {color: red;}
   a {text-decoration: none;}

   </style>
4. More properties
   http://www.w3schools.com/cssref/default.asp

Applying Colors

- Color Methods
  - Names
    ex: P {color: peachpuff }
  - Hexadecimal
    ex: P {color: #FF00FF; }
    - Shorthand
      ex: p {color: #F0F; }
  - RGB
    ex: P {color: rgb(100,50,55); }
    P {color: rgb(40%,12%,67%); }
- Properties
  - Color = text color
  - Background-color = color of element's background

Units of Measure

Absolute

- Best for pages that will print

Relative

- For scalable documents

No space b/n number and unit

Inches	**= in**
Centimeters	**= cm**
Millimeters	**= mm**
Points	**= pt**
Picas	**= pc**

Em	**= em** height of font
X-height	**= ex** elements x char.
Pixels	**= px** screen resolution
Percent	**= %** of element

Text Properties

- Text-align – (used to align any selector)
  Left, center, right, justify
- Text-indent
  (first-line)
- Line-height
- Text-transform
  (capitalize, lowercase, uppercase, normal)
- Text-decoration
  (underline, overline, line-through, none, [blink: only Nav.])
- Letter-spacing
- Word-spacing

Font Properties

- Font-family
  (font names, generic names)
- Font-size
  (units, xx-small – xx-large, smaller, larger)
- Font-weight
  (bold, bolder, lighter, normal, 100 – 900 [whole values])
- Font-style
  (italic, oblique, normal)
- Font-variant
  (small-caps, normal)
- Vertical-align
  (only on inline elements ex:) , <b>, <i>
- The font shorthand must follow the order of…
  {style weight size/height family; }

Choose Fonts

- Serif vs. Sans-Serif
  - Sans-serif is better than serif
- Common Fonts

Arial	Comic Sans MS	Times New Roman
Arial Black	Courier New	Georgia
Trebuchet MS	Impact	Verdana

- Web Fonts

Verdana	Georgia
Comic Sans MS	Trebuchet MS

http://www.w3schools.com/css/css_font.asp

Background Images

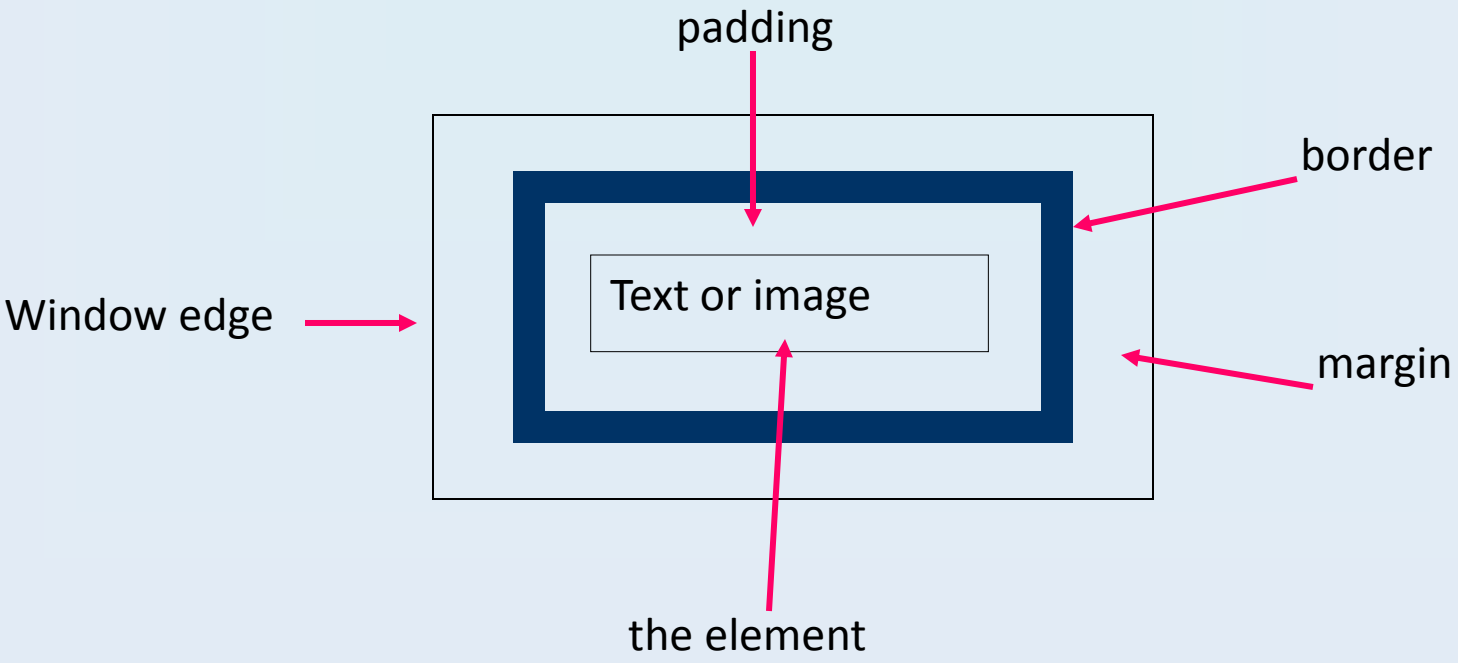
- Background-repeat:
  - repeat | repeat-x | repeat-y | no-repeat
- Background-image:
  - url(filename.xxx)
- Background-attachment:
  - fixed | scroll
- Background-position :
  - if you have used background-repeat: no-repeat;
  - keywords (in pairs) | x%, y%;
- Background ➔ shorthand (any order)
- Bound background-image with width and height properties

Link Pseudo-Classes

- Color hyperlinks at various stages
  - a:link – not yet used
  - a:visited – clicked links
  - a:hover – when moused over
  - a:active – when mouse button is down
- Should be listed in the above order to avoid browser confusion

Formatting with the Box Model

- Box model: elements consist of a box which includes the element itself, and its optional padding, border, and margin



Margins and Padding

- **Margin**
  - Space b\n element and its parent (typically <body>)
  - (margin-right, -left, -top, -bottom), or margin shorthand
  - Margins collapse vertically (greater of the two is applied)
- **Padding**
  - Space b\n elements content and box
    - Box: the natural space allotted an element's contents
  - (padding-right, -left, -bottom, -right), or padding shorthand
  - Table padding must be set on the <td>
- **Shorthand examples**
  - {margin: 10px; } = all four sides
  - {padding: 5px 8px} = top/bottom, left/right
  - {margin: 2px 5px 10px; } = top, left/right, bottom
  - {padding: 10px 20px 15px 5px; } = top, right, bottom, left

Borders

- Border: lines that display on the edge of the box
- Properties – (-top, -right, -bottom, -left)
  - Border-width: thin, medium, thick, (length unit)
  - Border-style: solid, ridge, double, inset, outset, dotted, dashed, groove,
  - Border-color: names, rgb, hex
  - Border shorthand: order doesn't matter
- Borders around tables and lists can be buggy
- Not all border values are supported, or supported the same way in the browsers
  - Dotted looks like dashed in Explorer
  - No thin or thick in Navigator

Element Dimension

- Dimensions → height and width
- Properties
  - Width
  - Height
- Total element dimension = width + padding + margin
- If you specify a width, but not the height, the browser determines the height according to the content

Floating

- Floating allows you to arrange elements on the page by defining their alignment to adjacent elements
- Properties
  - Float: alignment with other elements (like align attribute of the image tag)
    - (left, right)
  - Clear: prevents wrapping around an element
- Child elements float inside parent's box
- Clear can only be used on floated elements

Overflow

- Ensures that the browser will know what to do with content that exceeds their allotted space
- Property
  - Overflow
  - Values…
    - Scroll ➔display a scroll bar for excess content
    - Hidden ➔cut off excess content
    - Visible ➔content flows normally
    - Auto ➔ browser determines best display method
    - Inherit ➔ uses parent's value
- Scrollable Content
  - Use to apply the overflow property

Customize Lists

- List-style-type = label symbol
  - disc, circle, square, decimal, lower-alpha, upper-alpha, lower-roman, upper-roman, none
- List-style-position = control where new lines are located
  - inside, outside
- List-style-image = use and image for the bullet
  - url(filename.gif)
  - The image needs to be the correct size
  - Images take precedence over types
- Shorthand = {list-style: image type position}
  - To be safe, if you specify an image, also specify a type

Selectors

- 3 types of selectors
  - Element → HTML tags
  - Classes → custom rules for multiple elements
  - ID's →unique styles for one item
- Pseudo-classes denote a specific quality of an element
- Grouping selectors
  - Use a comma to separate them (p.4)
    ex: td, p, div {color: orange}

Class Selectors

- 2 types of classes
  - Element Classes
    - Associated with a specific element
    - div.header {property: value; }
  - Independent Classes
    - Not associated with a specific element
    - .author {property: value; }
- You can apply classes 3 ways
  - Element ➔ <p class="name">
  - Div ➔
    (Div is a block level element. It goes around tags)
  - Span ➔
    <Span is a text level element. It goes around text only)

ID Selectors

- Styles elements that are use the id attribute
  - <input type="text" id = "email"…/>
  - <style type="text/css">
    #email {property: value; }
    </style>
- 2 types of IDs
  - Element ID
    - Associated with a specific element
    - input#header {property: value; }
  - Independent ID
    - Same ID on different pages
    - #logo {property: value; }

Contextual Selectors

- Contextual Selectors
  - To target a specific child element nested in a parent element
  - p b {color: red}

Inline Style

- Single use of a style inserted directly into the elements
- Uses the style attribute
  ex: …
- Use Inline styles to…
  - Test the effect of a style
  - Create an exception to a rule
  - When creating a new class or id would be cumbersome
- You can also apply inline styles with the and tags

The Cascade

- To solve conflicts between styles
- Cascading → hierarchical order in which styles are applied when there is a conflict
- Top to Bottom
- General → Specific
- Hierarchy of Types of Style Sheets
  - In-line
  - Embedded
  - Linked / External

Linked (External) Style Sheets

- A linked style sheet is an external CSS document that is connected to the page using the link tag in the head section of the document
- <link rel="stylesheet" type="text/css" href="name.css" />
  - rel ➔ link relationship
  - Type ➔ type of style sheet
  - href ➔ filename of CSS doc.
- The CSS document is a text document containing style rules only and ends in the ***.css*** file extension

Comments

- /* your comment here */
- Located outside or inside the declaration
  ex:
  - Selector {property: value; /* comment… */ }
  - Selector {property: value; } /* comment… */

Creating Tables

- <table> </table> → create table
- <tr> </tr> → table row
- Cells
  - <th> </th> → table header
    bolds and centers
  - <td> </td> → regular cell

Tables

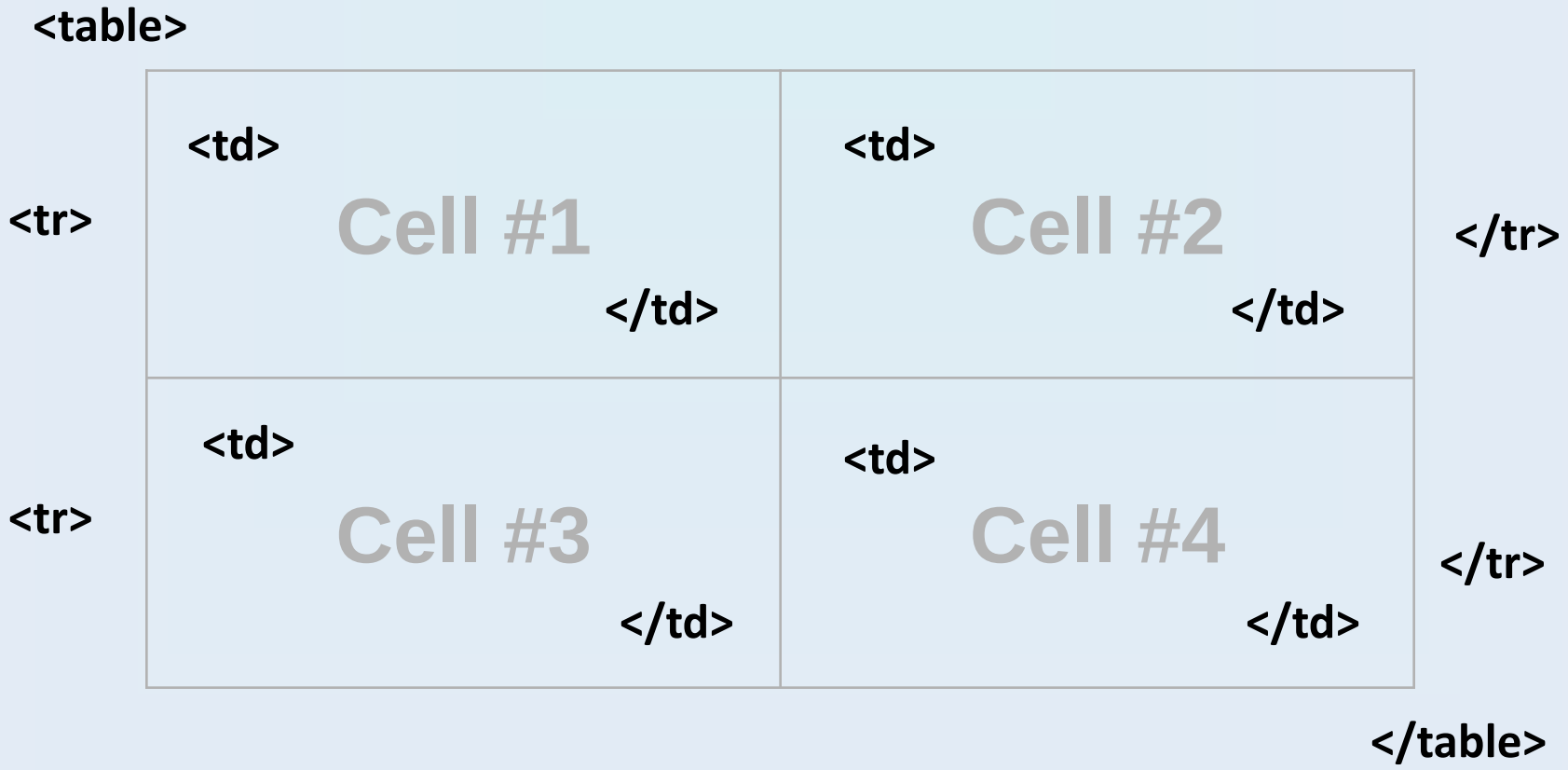

<table>
<tr>
<td>
Cell #1
</td>
<td>
Cell #2
</td>
</tr>
<tr>
<td>
Cell #3
</td>
<td>
Cell #4
</td>
</tr>
</table>

Table Attributes

- Tables
  - border ➔ greater than 1= outer border
  - width ➔ fixed (pixels) or flexible (percents)
  - cellpadding ➔ space within cell around content
  - cellspacing ➔ space between cells
  - summary ➔ description of table's purpose
- Rows and Cells
  - align ➔ left, center, right
  - valign ➔ top, center, bottom
- Deprecated
  - align ➔ use CSS – ok for cells and rows
  - bgcolor ➔ use CSS
  - width ➔ deprecated for cells

Create a Table Based Page Layout

- Fixed → maintains dimensions
  - width = pixel value
- Fluid → changes size
  - width = percent
- Attributes
  - colspan/rowspan
  - width
  - align

Create an Image Quilt

- An group of images that appear to be a solid image
  - Graphics application = slices
- Image appears to download faster
- Guidelines…
  - Set cellpadding and cellspacing to 0
  - Change vertical alignment
  - Remove line breaks within table code

Embed a Table

- To layout data within a cell
- Place <table> tags within a cell

```
<td>
    <table>
    </table>
</td>
```

Review

Merging Cells

- Rowspan → merge down
  - Ex: <td rowspan="#">…</td>
- Colspan → merge right
  - <td colspan="#">…</td>
- Value = # of cells including self

CSS Positioning

- You can precisely declare where you want certain elements to appear on the page.
- Position property - to choose the type of positioning
  - Absolute – from parent element
  - Relative – from it's normal position
  - Fixed – from the top-left corner of the viewport
  - Static – removes positioning, the same as none
- The coordinates of a position are controlled by 4 properties…
  - Top, Left, Bottom, and Right

http://css-tricks.com/absolute-relative-fixed-positioining-how-do-they-differ/

Absolute Positioning

- Absolutely positioned elements are independent of other elements
  - Positioned from parent element
  - Removed from the normal flow
- You will need to know…
  - Top and left coordinates,
  - Width and height
    - Graphics application
    - WYSIWYG
    - Storyboarding

Fluid Layout

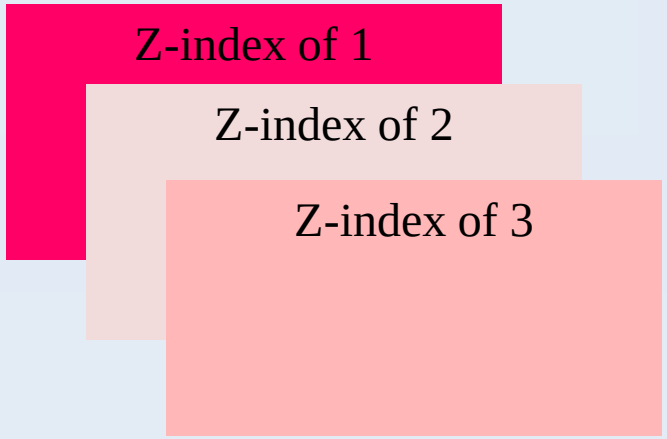
- Will reposition with browser size
- Can be used with fixed layouts
- Use the ‘right’ and ‘bottom’ properties
  - Right: amount of offset between the element’s upper right edge and the right edge of the parent
  - Bottom: amount of offset between the element’s lower right edge and the lower right edge of the parent
- Steps to make an element fluid
  - Set the element’s width using relative units of measure
  - Use left and right margins to control the width of the element
  - Use the right and bottom properties to place the element

Relative Positioning

- Position elements without disturbing other elements in the document
- Elements that follow a relatively positioned item are not disturbed
- Are positioned according their default position in the HTML document
- Will re-flow and change position with browser resizing
- Use position: relative, and the left, right, top, or bottom properties
- Use negative values to move elements up

Z-Index

- Positioning creates layers
- To control the stacking order of overlapping items
- By default, the last element in the document will appear at the top of the stack
- Older Navigator → all items must use z-index in order for the browser to recognize the property
- Use positive numbers



Creating Forms

- Create a Web Form Using…
  - <input>
    - Text fields, radio buttons, check boxes, submit/reset
  - <textarea>
    - Comments and Suggestions
  - <select> and <option>
    - Drop-down menus

Form

- **<form> </form>**
- **Attributes**
  - **action → data destination URL**
  - **method → how data is sent**
    - **get → as a query string (url object), not secure**
    - **post → as a form object, can secure**
  - **enctype → data character encoding**

<form action="" method="" enctype="" >

</form>

Form Elements

Form	Begins a form
Field Type	**Description**
Text	Single line
Text Area	Multiple lines
Radio Button	Only one valid answer
Checkbox	Many valid answers
Drop-Down Box	List of possible choices
Submit	Creates a button to send the info.
Reset	Creates a button to clear the form.

Input

< input type=“text/ password” name=“anything”
size=“#” maxlength=“#” />

< input type=“radio/ checkbox” name=“anything”
value=“choice” checked=“checked” />

< input type=“submit/ reset” name=“*optional***”**
value=“button label” />

Textarea

<textarea

cols=“number of characters wide in pixels”

rows=“number of lines tall”

name=“anything”>

</textarea>

Select & Option

<select name=“anything”>

<option value=“a choice in the menu”>

A Choice in the Menu

</option>

</select>

Fieldsets and Legends

- The <fieldset> element **creates a small box border** around the enclosed form controls to provide a visual cue that the form fields are grouped or related
- The <legend> allows you to **label the group**
- You can modify the fonts, colors, and borders with CSS

Option Groups

- Used to categorize drop-down options
- <optgroup>: creates category
- Label=“” to label the category
- Use CSS to style the groups
- Eventualy…
  - Create collapsible lists
    (currently: only IE for Mac)

Form Labels and access Keys

- <label>
  - Associate text with a form field
    - Click the text to access the field
- Only valid in a <form> block

Implicit and Explicit Labels

- Implicit labels are <label>s that contain their associated form field

  ex: <label>User name: <input type=…></label>
- Explicit labels are <label>s that are tied to a form field using the **for=** and **id=** attributes
- The **tabindex=** attribute lets you assign a tab order to the form fields.
- You can keyboard shortcuts to the <label> element using the **accesskey=** attribute.

Buttons

- <button> element → creates form buttons
- Creates a 3D push able effect
- Is a container: text and elements inside display on the button
- • Can control using CSS
  - Easier to specify (doesn't use <input>)

Frames

- Allow you to display more than one page
  - Navigation
- Elements
  - <frameset>
    - cols & rows
    - Sizing Options
  - <frame>
    - src – page
    - name – for targeting

Sizing Options

Value	Example
Absolute	Cols="50, 100, 200"
Percentage	Cols="25%, 75%"
Proportional	Rows="1*, 4*, 2*"

You can combine values

Modifying Frames

- <frame> attributes
  - noresize → prevents resizing
  - scrolling → creating scrollbars
    - yes, no, auto
  - frameborder → size of border
    - 0 = no border
  - marginwidth → left and right mergins
  - marginheight →top and bottom margins

Targeting Frames

- Default displays content in itself
- To the <frame>
  - Name="frame name"
- To the <a> (on actual page)
  - Target="frame name"
- Default target values
  - _blank → open in a new window
  - _self → open in the same frame as it was clicked
  - _parent → open in the parent frameset
  - _top → open in the full body of the window

Creating Inline (Floating) Frames

- <iframe>: allows you to embed other documents into a page.
- Uses the src= attribute to specify the page to embed
- You can use CSS or attributes to style iframes
- Iframe documents can be any file type a browser can display.

Linking to an Iframe

To target your hyperlinks to the iframe

1. You name the iframe using the name attribute
2. Use the target attribute to have you linked document display in the iframe

Media types

- Create a style sheet for specific media
  - Print friendly page
  - Page layout for mobile devices
  - Page for aural browsers
- Create external style sheet for media or @media property.

http://www.w3schools.com/css/css_mediatypes.asp

Media Queries

- Create layout dependent on browser width
- @ media in an embedded style sheet ***or***
- media query style sheet and link to it
  - @import url("…")
  - <link rel="stylesheet" type="text/css" href="*.css" />
- Not supported in IE 8 and prior
  - http://css-tricks.com/how-to-create-an-ie-only-stylesheet/

http://css-tricks.com/resolution-specific-stylesheets/

Automatically Redirect Web Pages

- <meta> describes web page
  - name or http-equiv
- To automatically load a page
  - Refresh
    - http-equiv="refresh"
  - Content…
    - seconds
    - URL
    - content="60;URL=index.htm"

Adding Multimedia

- Use <a> tag to link to audio file
  - Uses third party program to play
- Embed file
  - <embed />
  - <object></object>
- HTML 5
  - < audio> </audio >
  - <video> </video>

http://www.w3schools.com/html/html_media.asp

< Embed> Tag

- Supported in all major browsers
- Deprecated in HTML 4 and added back in HTML 5
- Example

```
<embed src="Calypso.mp3" />
```

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

< Object> Tag

- Supported in all major browsers
- Used instead of the <Embed> until HTML 5
- You will need to tell the browser what plug-in to use.
- May not work in all browsers

```
<object classid="clsid:02bf25d5-8c17-4b23-bc80-d3488abddc6b" width="250
    "height="250"
    codebase="http://www.apple.com/qtactivex/qtplugin.cab">
    <param name="src" value="../cyclers.mov">
</object>
```

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

Legacy Multimedia Code Example

```
<object classid=“clsid:02bf25d5-8c17-4b23-bc80-d3488abddc6b"
width="250 "height="250"
        codebase=“http://www.apple.com/qtactivex/qtplugin.cab">
        <param name="src" value="../cyclers.mov">
        <param name="autoplay" value="true">
        <param name="controller" value="true">
        <embed src="../cyclers.mov" width="250" height="250"
autoplay="true" controller="true"
pluginspage="http://www.apple.com/quicktime/download/">
        </embed>
</object>
```

HTML – <Audio> Tag

- Supported in Internet Explorer 9, Firefox, Opera, Chrome, and Safari.
  (Internet Explorer 8 and earlier versions, do not support the <audio> tag.)
- ```
  <audio controls="controls">
    <source src="Calypso.ogg" type="audio/ogg">
    <source src="Calypso.mp3" type="audio/mpeg">
  Your browser does not support the audio element.
  </audio>
  ```
- Attributes
  - Autoplay: Specifies that the audio will start playing as soon as it is ready
  - Controls: Specifies that audio controls should be displayed (such as a play/pause button etc).
  - Loop: Specifies that the audio will start over again, every time it is finished
  - Preload: Specifies if and how the author thinks the audio should be loaded when the page loads
  - Src: Specifies the URL of the audio file

http://www.w3schools.com/tags/tag_audio.asp
  ```

HTML - <Video> Tag

Supported in Internet Explorer 9, Firefox, Opera, Chrome, and Safari.
(Internet Explorer 8 and earlier versions, do not support the <audio> tag.)

- ```
  <video width="320" height="240" controls="controls">
    <source src="movie.mp4" type="video/mp4">
    <source src="movie.ogg" type="video/ogg">
    Your browser does not support the video tag.
  </video>
  ```
- Attributes
  - autoplay: Specifies that the video will start playing as soon as it is ready
  - controls: Specifies that video controls should be displayed
  - height: Sets the height of the video player
  - loop: Specifies that the video will start over again, every time it is finished
  - muted: Specifies that the audio output of the video should be muted
  - src: Specifies the URL of the video file
  - width: Sets the width of the video player

http://www.w3schools.com/tags/tag_video.asp
  ```

Other Media Resources

- http://www.w3schools.com/html/html_media.asp
- http://www.w3schools.com/html/html_sounds.asp
- http://www.w3schools.com/html/html_videos.asp
- http://www.w3schools.com/html/html_object.asp

Multimedia Code Examples with HTML 5

```
<audio controls="controls" height="50px” width="100px">
<source src="song.mp3" type="audio/mpeg" />
<source src="song.ogg" type="audio/ogg" />
<object height="50px" width="100px" data="song.mp3" />
< embed height="50px width="100px" src="song.mp3" />
</audio>
```

```
<video width="320" height="240" controls="controls">
<source src="movie.mp4" type="video/mp4" />
<source src="movie.ogg" type="video/ogg" />
<source src="movie.webm" type="video/webm" />
< object data="movie.mp4" width="320" height="240">
<embed src="movie.swf" width="320" height="240">
Your browser does not support video
</object>
</video>
```

Link to YouTube

- iFrame
  - <iframe width="#" height="#" src="*YouTube link*"> </iframe>
- Embed
  - <embed width="#" height="#" src="*YouTube link*" type="application/x-shockwave-flash"> </embed>

DOCTYPE element

- The document type declaration, or DOCTYPE declaration, states which version of HTML is used in the document
- The DTD (document type definition) defines the rules for the markup
- The browsers use the DOCTYPE to instruct them on how to display the document

DTDs in HTML

- There are 3 possible document types…
  - Strict – in strict compliance with the rules of HTML
    - CSS controls all visual formatting
    - Contains no deprecated elements or attributes
    - Excludes any frameset markups
  - Transitional
    - Allows deprecated tags to accommodate older browsers
    - Prevents legacy code from breaking down
  - Frameset
    - For framesets
- Must be declared before the opening <HTML>
- http://www.w3schools.com/tags/tag_doctype.asp

Validation

- To validate is to ensure that your DOCTYPE fits your document
- The DOCTYPE declaration must be error free (keep copies in text documents)
- Some HTML editors, like HomeSite, include validation features, but for the most part, you must validate the document yourself
- W3C's Validator
  - http://validator.w3.org/