Lecture notes

Lecture 1

Web Main Components

- Client with Web Browser
- Internet
- Web Server

Internet

- Transmission Control Protocol/Internet Protocol (TCP/IP)
  - Same for all computers
  - Allows for data exchange between computers
  - Tolerates unreliable sub-networks
  - Guarantees proper data transmission, since the phisical network can't
  - Layers
    - Application (HTTP, telnet, ftp, email, VoIP)
      - Ability to access services of other layers
      - Defines protocols for data exchange
    - Transport (TCP, UDP)
      - Responsible for end-to-end message delivery
    - Internet/Network (IP)
      - Routing messages
      - All routers run on the IP protocol
    - Physical (Ethernet, WiFi, ATM, x.25, Frame Relay)
      - Translates message into physical representation
      - Puts messages on the wire/wireless network/fiber-optic wire
- Addressing Schemes (where to go)
  - Unique address required
  - Internet Protocol (IP) address
  - Ex.: 205.46.117.104 (each between 0 and 255)
- Domain names (human understandable)
  - DNS: The phonebook of the internet
  - Naming system
  - Associates domain names (human readable) with IP addresses
  - Between client and server
  - 1985
- Routing Traffic Across the internet (how to go)
- The World Wide Web (WWW)
  - Geneva, 1989
  - System of interconnected hypertext documents
  - A distributed hypertext system
  - Set of common communication protocols
  - Multi-platform
  - Hypertext: method of organizing information
    - Gives the reader control over the order in which the information is presented
- HTML
  - Markup: structure
  - Publishing language of the World Wide Web
  - HTML 5 newest
  - `<body>` Everything inside is shown in browser window
    - `<head>`
      - Information about the page
      - `<title>`, charset, etc.
    - `<title>`
      - Shown in the top of the browser
  - Attributes
    - Additional information to elements (name=value)

Lecture 2

HTML structure

```html
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Title of my webpage</title>
  </head>
  <body>
    <p>Hello there</p>
  </body>
</html>
```

- `<!DOCTYPE html>`: Document Type Definition (DTD)
- `<meta charset="utf-8">`
  - Character encoding
  - Can also have:
    - `<meta name="description" ... >`
    - `<meta name="keywords" ... >`
    - `<meta name="author" ... >`
    - Etc.
- Tags can be one-sided or two-sided
- `<!- comment -->`: The comment tag
- HTML treats white-space as a single blank space
  - Exception: the `<pre>` tag
- Inline elements
  - Displayed without starting a new line
  - `<b>`, `<td>`, `<a>`, `<img>`, etc.
- Block elements
  - Start and ends with new line
  - Ex.: `<h1>`, `<p>`, `<ul>`, `<table>`, etc.
- Grouping tags
  - `<span>` (inline)
  - `<div>` (block)
- Semantic Markup (logical elements)
  - `<q>`/`<blockquote>`
  - `<abbr>`
  - `<address>`
- Lists
  - ordered `<ol>`
  - unordered `<ul>`
  - definition `<dl>`
    - `<dt>` for the term
    - `<dd>` for the definition of the term (below, not inside)
  - Can be nested
  - Styled list: `<ol style="list-style-type: upper-roman>`
- Links
  - `<a href="http://exmaple.com>`
  - When linking to page within the same site, you can use a relative URL
  - Can link to elements with id: http://www.sundance.org/#jumbotron
- Images
  - _src_: path to image
  - _alt_: text description, for accessibility and indexing purposes
  - Specifying height/width nice so that the browser can render the rest of the page while leaving room
    - Either as attributes or CSS
    - Measure images in pixels
- Figure captions
```html
  <figure>
    <img src=".." alt="..." />
    <br />
    <figcaption>some text</figcaption>
  </figure>
```

- JPEGs or GIFs best
- Tables
```html
  <table>
    <tr>
      <th scope="col">Saturay</th>
      <th scope="col">Sunday</th>
    </tr>
    <tr>
      <td>1</td>
      <td>1</td>
    </tr>
    <tr>
      <th scope="row">Total sales</th>
      <td>1</td>
      <td>1</td>
    </tr>
  </table>
```

  - Use colspan="2" on row to span two columns
  - Use rowspan="2" on row to span two rows
  - Tags for accessibility, indexing and desgn purposes:
    - `<thead>`
    - `<tbody>`
    - `<tfoot>`
- Site maps
  - Website guide
    Should not require further user interaction
  - When to use
    - Large site
    - Large archive, isolated pages, etc.
    - New site and few external links
    - Rich media content
  - When not to use
    - Small site
    - Comprehensive linking
    - Few media files

URLs

- Protocol
  - http://www.example.com/path/to/myfile.html?key1=value1&key2=value2…
    - http:// indicates protocol (http default)
    - www.exmaple.com is the domain name
    - /path/to/myfile.html is path to resource on server
    - ?key1=value1&key2=value2 are extra query parameters (key/value pairs separated with &)
- GET method
  - Request data from a specified resource
  - Query strings
  - Can be cached
  - Remain in browser history
  - Can be bookmarked
  - Can be distributed and shared
  - Have lenght restrictions (max URL length)
  - Can be hacked
- POST requests
  - Sent as a separate message
  - Never cached
  - Used for sensitive data
  - No max length
- URL anchor
  - …to/myfile.html?key1=value1&key2=value2#SomewhereInTheDocument
    - Anchor to part of th resource itself

Lecture 3 - CSS

- XHTML is a stricter version of HTML
  - Recommended for accuracy reasons
- The Problems with HTML
  - No consistency
  - Can omit attributes, etc. (poorly written code)
  - Coding design
    - `<font>`
  - Compatibility
    - Cross-platform compatibility has been degraded (different browsers, etc.)
- Solution: Separate Structure from Appearance
  - CSS (Cascading Style Sheets)
  - CSS1 in December 1996
  - Types
    - External style sheet (included in `<head>`)
    - Internal style sheet
    - Inline style
- How does CSS work?
  - Web browsers assume a default set of values for each property
  - CSS defines an extensive set of presentation properties
  - A CSS style sheet changes that default rendering
  - Use !important to override
  - Rules with same origin conflict --> more specific applies
  - Equally specific -> last is chosen
  - Style prioritization: inline > embedded > external
- Inheritance
  - `<h1>The headline <em>is</em> important!</h1>`
  - If h1 has color blue, em gets color blue
  - Only some styles are inherites (font-family, text-alignment, etc.)
  - Others are not (padding, etc.)
- Multiple declarations
  - Several `h1` declarations will be merged
- Grouping
  - Comma-separated selectors will give all the same declaration
- Selecting elements
  - `*` - universal
  - `h2` - type selector
  - `.element` - class selector
  - `#elementID` - ID selector
  - `p a` - descendant selector
  - `div[style]`, `input[type="text"]` - attribute selector
  - `one > two` - immediate child selector
  - `h2+p` - adjacent sibling selector, same parent
  - `h2~p` - general sibling selector (does not have to be adjacent)
  - `a:visited`, `a:hover` - pseudo-class
    - Selects an element based on a state the element is in
  - `p::first-letter`, `p::before` - pseudo-element
    - Used to add special effects to some selectors
    - Represents elements that are not really part of the rendered HTML
    - Creates new virtual elements
- Specificity Value
  - Style attribute > ID > Class, pseudo-class, attribute > Elements

Lecture 4 - CSS properties

- Colors
  - Properties
    - color
    - background-color (color or "transparent")
    - background-image (URI)
  - Color Values

    - Named
    - RGB
    - Hex
    - HSL with optional opacity value
- Fonts
  - Properties
    - font-family (specify order of preferenct)
    - font-style (normal, italic or oblique)
    - font-variant (normal, small-caps)
    - font-weight (normal, bold, bolder, lighter)
    - font-size
      - Absolute
        - mm, cm, in, pt, pc
      - Relative
        - Percentage
        - em (1em == 100% == font-size of parent element)
        - rem
      - Can use pixels or keywords
- Text
  - text-indent (amount of indentation using absolute length or percentage)
  - text-align (left, center, right, justify)
  - text-decoration (non, underline, overline, line-through)
  - etc.
- The Box Model
  - Margin
  - Border
  - Padding
  - Content
- Overflowing Content
  - What to do if the content is largen than the box itself?
  - hidden, overflow, scroll, etc.
- Borders
  - border-width: 1px 4px 12px 4px (clockwise starting from the top)
  - Border styles:
    - solid, dotted, dashed, double, groove, ridge, inset, outset
    - Ex.: border-style: dotted solid double (top inline bottom)
- Padding
  - Space between the content and its border
- Margin
  - Gap between boxes
- Change inline/block
  - `display: inline` - turn a block element into an inline one
- Hiding
  - `visibility: hidden` hides the element but it leaves a space where it would have been
- Border Images
  1. The url of the image
  2. Where to slice the image
  3. What to do with the straight edges
     1. Stretch
     2. Round
- Box Shadows
  - Horizontal offset
  - Vertical offset
  - Blur distance
  - Spread of shadow
  - `p.one { box-shadow: -5px -5px #777777; }`
- Position
  - position: static
    - Default
    - Each block element sits on top of the next one
  - position: relative
    - Moves an element in relation to where it would have been in normal flow
    - top, left, right, bottom, etc.
  - position: absolute
    - Box is taken out of normal flow
  - position: fixed
    - Position in relation to the browser window
- Overlapping
  - relative, fixed, absolute --> boxes can overlap
  - Later element sits on top of older
  - Specify order with z-index
- Floating elements
  - Allows you to take an element in normal flow and place it as far to the left/right as it goes
- Table properties
  - width
  - padding
  - letter-spacing, font-size
  - border-top/bottom
  - :hover
- Display style
  - none
  - block
  - inline
  - inline-block
  - inline-table
  - inherit
  - list-item
  - run-in
  - table
  - table-caption
  - table-cell
  - table-column
  - table-column-group
  - table-footer-group
  - table-header-group
  - table-row
  - table-row-group

Lecture 5

Flexbox

- Parent properties
  - `display: flex` enables flex context for all direct children
  - `flex-direction: row (default) | row-reverse | column | column-reverse;`
  - `flex-wrap: nowrap (default) | wrap | wrap-reverse;`
  - `flex-flow: column wrap;` - shorthand for `flex-direction` and `flex-wrap`
  - `justify-content: flex-start (default) | flex-end | center | space-between | space-around | space-evenly | start | end | left | right ... + safe | unsafe;` - alignment along the **main axis**
  - `align-items: stretch | flex-start | flex-end | start | end | self-start | self-end + ... safe | unsafe;` - alignment along the **cross axis**

Grid