HTML & CSS Flashcards

Basic HTML DOM Tree

```
<!doctype html>
  <head>
    <title></title>
    <meta>
    <script></script>
  </head>
  <body>
  </body>
  <!-- Comment -->
```

Headings, Paragraphs, and More

```
  <h1></h1> through <h6></h6>
  <p></p> for paragraphs
  <aside></aside>content
```

CSS Setup & Formatting

```
<style type=”test/css”> OR
<link rel="stylesheet" type="text/css"
  href="mystyle.css"> OR
  inline
tagname {    - all ref’ed elements inherit
  property: value;} - how to format
#id {}    - single id’ed element will
               inherit these
.class {}    - all elements of class will
               inherit
</style>     - closes out the style segment
```

Text Attributes

```
  font-family: typeface;
  font-style: normal-italic-oblique;
  font-weight: normal-bold-###;
  font-size: ##px;
  text-align: left-center-right;
  vertical-align: top-middle-bottom;
  text-transform: uppercase-lowercase-
    capitalize;
  text-shadow: _px _px _px color;
    horizontal vertical blur color
```

Link Formatting

```
<a href=”url”>link text
<img src=file alt=”text”
    style=”float:rightORleft
    width:__px; height:__px;
    border:0;”>
  target=”_blank”</a>
<a href= "mailto:jo@doe.org">Email Jo</a>.
```

CSS Selectors

Examples

* - targets all elements. Universal.

.*class* - targets relevant class

#*id* - targets relevant id.

h1 b - targets h1 level bold formatting

p#intro - paragraph level with ID of “intro”

HTML & CSS Organization

```
<div></div>
```

<span></span> - a “within element” version of <div> and is replacing default font stylings - bold, italic, and others.

<main></main> to be used as a way to delineate the “main” section of the html.

General Element Attributes

```
alignment: left-center-right;
color: rgb/hex/name;
background-image: url('file');
background-repeat: no-repeat;
background-size: cover;
background-attachment: fixed;
padding: number;
background-color: color;
```

CSS Border Attributes

```
border-style: dashed-dotted-double-
  inset-more
border-width: px;
border-color: color;
OR
border: px border-style color
   (above example defines all necessary
    elements to create a border.)
```

List Items

```
<ul></ul> OR <ol></ol>
  <li></li>
<dl>          <!-- definition list -->
  <dt></dt>   <!-- definition term -->
  <dd></dd>   <!-- definition -->
<dl>
(ul or ol can be nested)
```

List Attributes: Ordered / Unordered

```
  list-style:style:disc-circle-
  square-none; (impacts <ul></ul> lists)
  Type="1-A-a-I-i"; (impacts ol lists)
```

Additional List Attributes (pp 333 - 340)

```
  list-style-position:outside/inside;
  List-style-image (can use img for bullet)
  list-style:style:disc-circle-
  square-none; (impacts <ul></ul> li
  border-spacing: px px px px;
  border-collapse: collapse;
```

Text Attributes

|<i></i> *Italics*| <b></b> **Bold** |

|<em></em> *emphasis*| <q></q> “quotes” |

|<strong></strong> bold-like| |

|<blockquote></blockquote> | |

|<abbr></abbr> abbreviations & acronyms||

|<cite></cite> citations | |

|<dfn></dfn> defining instance | |

|<ins></ins> inserted content | |

|<del></del> shows deleted content | |

Text Attributes continued

|<s></s> strikethrough | |

|<sup></sup> superscript | |

|<sub></sub> subscript | |

|<br /> line break in an element | |

|<hr /> a horizontal line element | |

<address></address>
Highlights an address, but the address must still be formatted via html attributes. For additional options, see hCards.

Tables

<table></table>

<tr></tr> Table Row <td></td> Table Data

<th></th> OR <thead></thead>
Table Heading

<tbody></>tbody> Table Body

<tfoot></tfoot> Table Foot

Spanning fields are possible.

Older html may include width & spacing and border & background styling.

Subject 4

content

Boxes

```
width: px OR xx%    height: px OR xx%
min-width: px       max-width: px
min-height: px      max-height: px
overflow: hidden-scroll;
  Instructs what to do if content is
  Larger than the box
```

Margin & Padding

```
padding: px; can be top-left-right-bottom
margin: px;
```

Boxes Border

```
border-width: thin-medium-thick
border style: solid-dash--groove-more
  can be border-top/left/right/bottom
border-color: rgb
  can be designated: top right left
    bottom;
shorthand border:
  border: width style color;
```

Centering Content

```
margin: px auto px auto;
  Will center the box on the screen
```

Visibility

```
visibility: hidden-visible
```

Inline/Block Change

```
display: inline/block/inline-block/none
```

Border Images

```
css element {
   filename: url("url")
      px px px px stretch/repeat/round
   filename: url("url")
      px px px px stretch/repeat/round
   filename: url("url")
      px px px px stretch/repeat/round;
}
```

Border Box Shadows

```
css element {
  -Moz-box-shadow: px px rgb;
  -webkit-box-shadow: px px rgb;
  box-shadow: px px rgb;}
```

Properties:

hor. offset vert. offset

blur distance spread of shadow

Rounded Corners

Border-radius: px, px, px, px

Elliptical Shapes:

See p. 322

Position Control - Schemes

Normal - each block level element appears on a new line.

Relative Position - moves it relation to where it would go on default.

Absolute Position - as in "background". Other elements ignore it's placement.

Fixed Position - absolute to the browser window. "Sticky"

Floating - other elements within the block-level float around it.

Position Control - Schemes (cont.)

Normal Flow - position:static;

Relative - psotiion:relative;

Absolute - position:absolute;

Fixed - position:fixed;

z-index: '*number*'

The higher the number, the higher it's precedence to the top of the screen.

Floating Element (cont.)

```
float: right/left;
```

Other elements within the block flow around it.

Clearing floats

```
.clear {
  Clear: left/right/both/none;}
```

Ensures no other element touches the assigned value.

Clearing float Problem

To ensure proper floating add

```
   overflow: auto;
   width: 100%;
```

so that floats are properly formatted.

MultiColumn Layouts

See pp 375-376.

Layout Options - pp 370-390

Image Attributes

Align: left/right (put it far left/right of element/block)

Align: top/middle/bottom (puts it inline, text will be at its top/middle/bottom)

<figcaption></figcaption> HTML5 tag for photo captions

Color

```
background-color: rbga (0.0,0.0.0.0,0.0);
  red blue green alpha
line height: em;
```

Font Attributes

```
text-align: left/right/center/justify;
vertical-align: text-top/baseline/
   text-bottom;
```

Text Attributes

```
font-family: ‘name’; src: url(‘url’);
  format(‘name’);
line-height: em;
text-align: left-right-center-justify;
vertical-align: text-top/baseline/
   text-bottom;
:first-letter :first-line %:
```

Subject 3

content

Styling Links

```
:link, :visited, :hover, :active:
    All font properties can affect these.
```

Attribute Selectors (see p 292).

Subject 4

content

Creating an Object using Literal

```
var objectName = new Object():
  objectName.propertyName = value;
  objectName.methodName = function () {
    CODE;
}
```

The above format properly creates new objects with properties and values, methods and functions.

Updating an Object

```
objectName.propertyName = value;
```

Object properties can also be updated w square brackets syntax in place of dot notation.

Deleting an Object

```
delete objectName.propertyName;
```

Clearing an Object

```
objectName.propertyName = ‘’;
```

Creating Many Objects: Constructor Notation

```
function objectName (property1,property2) {
  this.property1 = property1;
  this.property2 = property2;
  this.methodName = function() {
    CODE;
}
```

Subject 1

content

HTML Forms (pp 144-175)

How html collects user info.

They live in <form></form>, sent w

get OR post (p151).

Inputs: type = “text” “password” “radio” “checkbox”

<select name=”input”></> <input></> <button></> <input></> type=”hidden”

<label></><fieldset></><legend></>

Subject 3

content

Subject 2

content

Subject 4

content