

CSE 383

Module 4

Agenda

HTTP Protocol

HTML

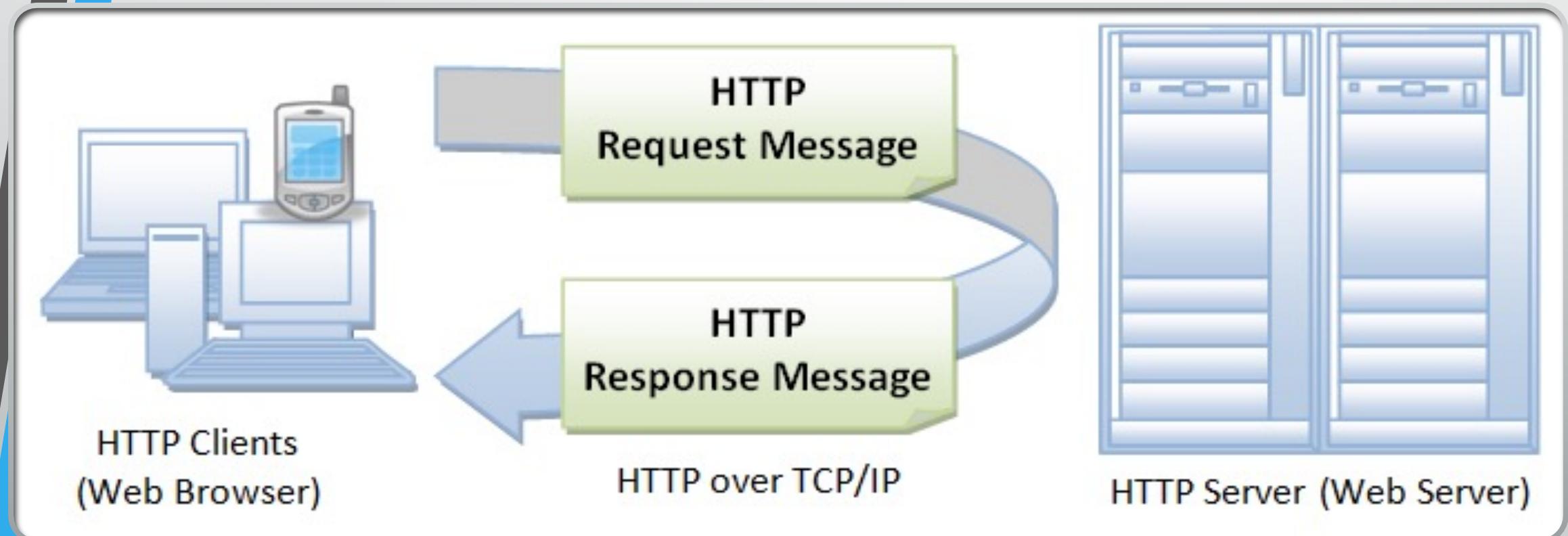
Attendance

CSS

Forms/Bootstrap

Exam1 Week 6

HTTP Protocol





HTTP – Standard Process



Open a TCP Connection



Send an HTTP message



Read the response



Close the connection



HTTP Requests

Request Message consists of:

- REQ LINE
- Headers
- NEWLINE
- Body



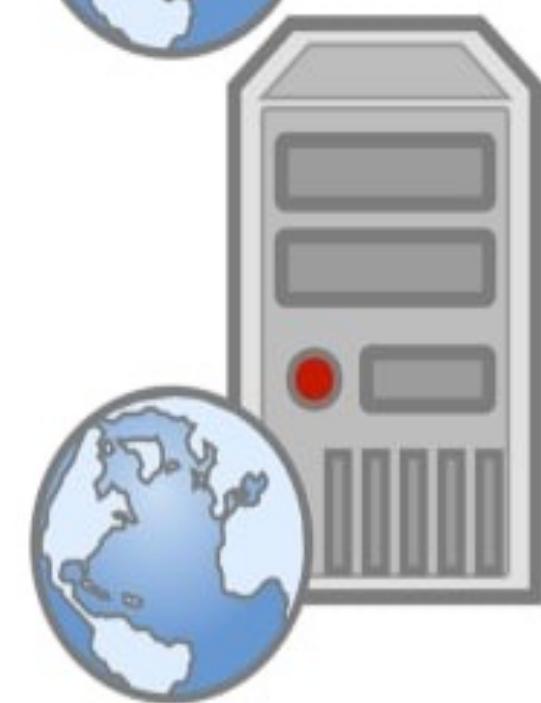
POST

`http://example.php`



GET

`http://example.php?key=value`





HTTP Requests

METHOD PATH VERSION

Valid Methods:

- GET
- POST
- HEAD
- STATUS
- Others

Header is ALWAYS ascii, each line ends in "\r\n".

Headers end with "\r\n" by themselves

GET Example – original protocol 0.9

Linux command to communicate with server

telnet ceclnxo1.cec.miamioh.edu 80

Trying 134.53.148.193...

Connected to ceclnxo1.csi.miamioh.edu.

Escape character is '^['.

→ GET /
← <!doctype html>
← <html lang='en'>
← <head>
← <title>CECLNXo1</title>
...



GET Example – HTTP 1.1

```
telnet ceclnxo1.cec.miamioh.edu 80
```

```
Trying 134.53.148.193...
```

```
Connected to ceclnxo1.csi.miamioh.edu.
```

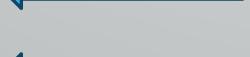
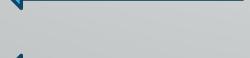
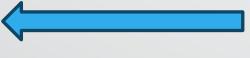
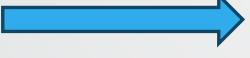
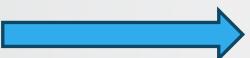
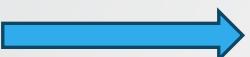
```
Escape character is '^]'.

---


```

```
GET /login?user=kurt&password=SuperSecret HTTP/1.1
```

```
Host: ceclnxo1.cec.miamioh.edu
```



```
HTTP/1.1 200 OK
```

```
Date: Sat, 28 Jan 2023 19:45:13 GMT
```

```
Server: Apache/2.4.41 (Ubuntu)
```

```
Vary: Accept-Encoding
```

```
Content-Length: 791
```

```
Content-Type: text/html; charset=UTF-8
```

```
<!doctype html>
```

```
<html lang='en'>
```



POST

- POST /login HTTP/1.1
- HOST: ceclnx01.cec.miamioh.edu
- Content-Type: text/plain
- Content-Length: 30

- user=kurt&password=SuperSecret

- ← HTTP/1.1 200 OK
- ← Date: Sat, 28 Jan 2023 19:45:13 GMT
- ← Server: Apache/2.4.41 (Ubuntu)
- ← Vary: Accept-Encoding
- ← Content-Length: 791
- ← Content-Type: text/html; charset=UTF-8

- ← <!doctype html>
- ← <html lang='en'>



Response Codes

1xx - Information

2xx - OK

- 200 - OK

3xx - Moved

- 301 - Moved Permanently

4xx - URL error (can't find object)

- 404 - can't find

5xx - Server failure

Content Encoding and Transfer

Content can be compressed to improve transfer efficiency

How much data in payload?

- Close -> Till input stream closed
- Content-Length header -> If we know the amount of data, can put in header
- Chunked -> Send in blocks where the first 4 bytes of each block indicate the length of the data in the block.

Response Headers

- Headers are Key Value pairs in the metadata providing information about the message
- Content-Length is one header
- Content-Type is IMPORTANT -> tells receiver what type of payload is being sent
- Cookies -> persistent data for each request
- Last-Modified -> Date content was last modified
- Server -> What server is handling data
- Content-Encoding -> Is it compressed?

Mime Types

- Hypertext is a loose term. How does receiver know what type of data is coming back?
- Content-Type, in the header, tells receiver
- HTML -> Content-Type: text/html
- Plain Text -> Content-Type: text/plain
- image/jpg, image/png, application/json
- And lots of others.
- Frequently server will set content type based on extension of object, but if program generating response, then only program knows the content type and the program will send the appropriate MIME type





Uniform
Resource
Locator -> URL

HTTP://foo.com:2345/a/b/c.html

Protocol: http

Port: 2345 (80 if not set)

Computer name: foo.com

Directory: /a/b

Filename: c.html





What is a Web Client?

**Web Client is simply
an application that
speaks HTTP to a
server**

Browser (Chrome,
Firefox, Edge, ...)

Any program can use
HTTP and become a
web client



What is a Web Server?

Web Server is simply a server that speaks HTTP

Can return objects or invoke programs

apache is common program

PHP is a language module that runs inside of apache.

Tomcat -> Webserver written in Java, runs java http servlets

Custom programs can use HTTP and become web servers



HTTP and REST

REST -> Representational State Transfer.

It used to be that almost all HTTP requests were to obtain documents

Now HTTP handles programmatic data transfer. REST is standard for this programmatic data transfer.

Used as part of AJAX (Asynchronous Javascript with XML)

Used as program to program data transfer for Remote Procedure Calls



HTTP things to know

- Be able to write HTTP Requests and Responses
- Status Types
- Basic MIME types
- Request Methods
- Given a URL, what would the sequence to obtain the content



Html Review

Structure

- Head
- Body
- Footer

Tags

- H1 - h6
- Div
- Span
- P

Img

- Relative
- Fixed
- local/remote

Links (<a>)



Html Review

Table

- <table><tr><th></th></tr><tr><td></td></tr></table>

Bullet Points (unordered list)

-

Communication

- Get/Post
- Headers
- Body
- Socket

What is HTML5

Validation/Accessibility

Section A

Section B

Section C

CSS

Cascading Style Sheets



CSS

Cascading Style Sheets

- Cascading since final properties depend upon all merged rules.

Inserting css

External style sheet

Internal style sheet

Inline style

Cascading

Closer CSS has a higher priority

- External (low)
- Internal (medium)
- Inline (high priority)

More Specific CSS has a higher priority

- P (specifies all paragraphs)
- P#intro (specifies all paragraphs with the id intro)



class



Class of styles



```
<p class='infoClass'>This is text</p>
```



```
<p class='infoClass'>More Text</p>
```



`p.infoClass {color: red}` -> is applied to BOTH paragraphs



Multiple elements can have same class

Selector Magic

- p → all paragraphs (select by tag)
- .class → all elements of given class
- p.class → all elements of given class of tag p
(paragraph)
- #id → named element (name = id)
- * → All Elements
- div p → all paragraph that are inside a div
- div > p → all paragraph whose direct parent is a div
- p::first-line → select first line of paragraph

• AND LOTS more ->

http://www.w3schools.com/cssref/css_selectors.asp

Display

Display = [none block inline inline-block]

None: Do not display this (not shown)

Block:

- Displays on new line (separated by top and bottom from other elements)
- Starts on a new line and takes up the whole width

Inline:

- Displays with the other elements declared around it (no line separation)
- Height/Width properties have NO effect

Inline-block

- Displays an element as an inline-level block container. The element itself is formatted as an inline element, but you can apply height and width values



Multiple classes

<p class="info important menu">This paragraph has multiple classes controlling its style</p>

p.info {color: red}

p.important {font-size: +2em}

p.menu {background-color: yellow}

Chrome Inspect

DevTools - chrome://new-tab-page/

Elements Memory Console Recorder Sources Network >> ⚠ 1 ✖ 1 ⚙ ⋮

ground: rgba(255, 255, 255, 1.00); --color-new-tab-page-most-visited-foreground: rgba(255, 255, 255, 1.00); --ntp-logo-color: rgba(255, 255, 255, 1.00);">...

> </div> flex

> <dom-if restamp style="display: none;">...

> <dom-if id="customizeDialogIf" restamp style="display: none;">...

> <svg>...

<!--_html_template_end_-->

... </ntp-app> == \$0

<script type="module" src="new_tab_page.js"></script>

<link rel="stylesheet" href="chrome://resources/css/text_defaults_md.css">

<link rel="stylesheet" href="chrome://theme/colors.css?sets=ui,chrome">

<link rel="stylesheet" href="shared_vars.css">

<script type="module" src=".lazy_load.js"></script>

</body>

</html>

html body ntp-app

Styles Computed Layout >>

Filter :hov .cls + ↻

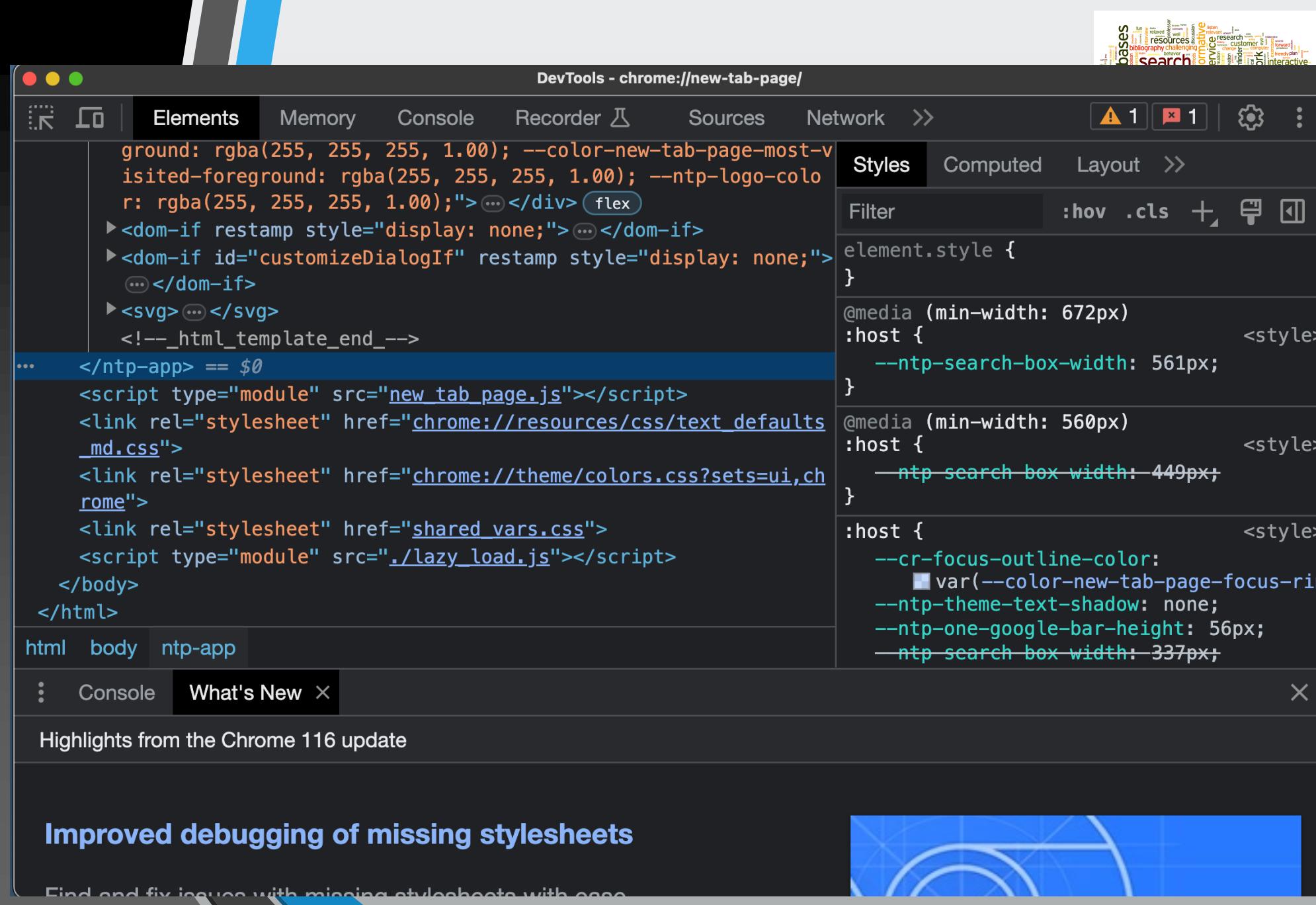
element.style { } @media (min-width: 672px) :host { --ntp-search-box-width: 561px; } @media (min-width: 560px) :host { --ntp-search-box-width: 449px; } :host { --cr-focus-outline-color: var(--color-new-tab-page-focus-ring); --ntp-theme-text-shadow: none; --ntp-one-google-bar-height: 56px; --ntp-search-box-width: 337px; }

Console What's New X

Highlights from the Chrome 116 update

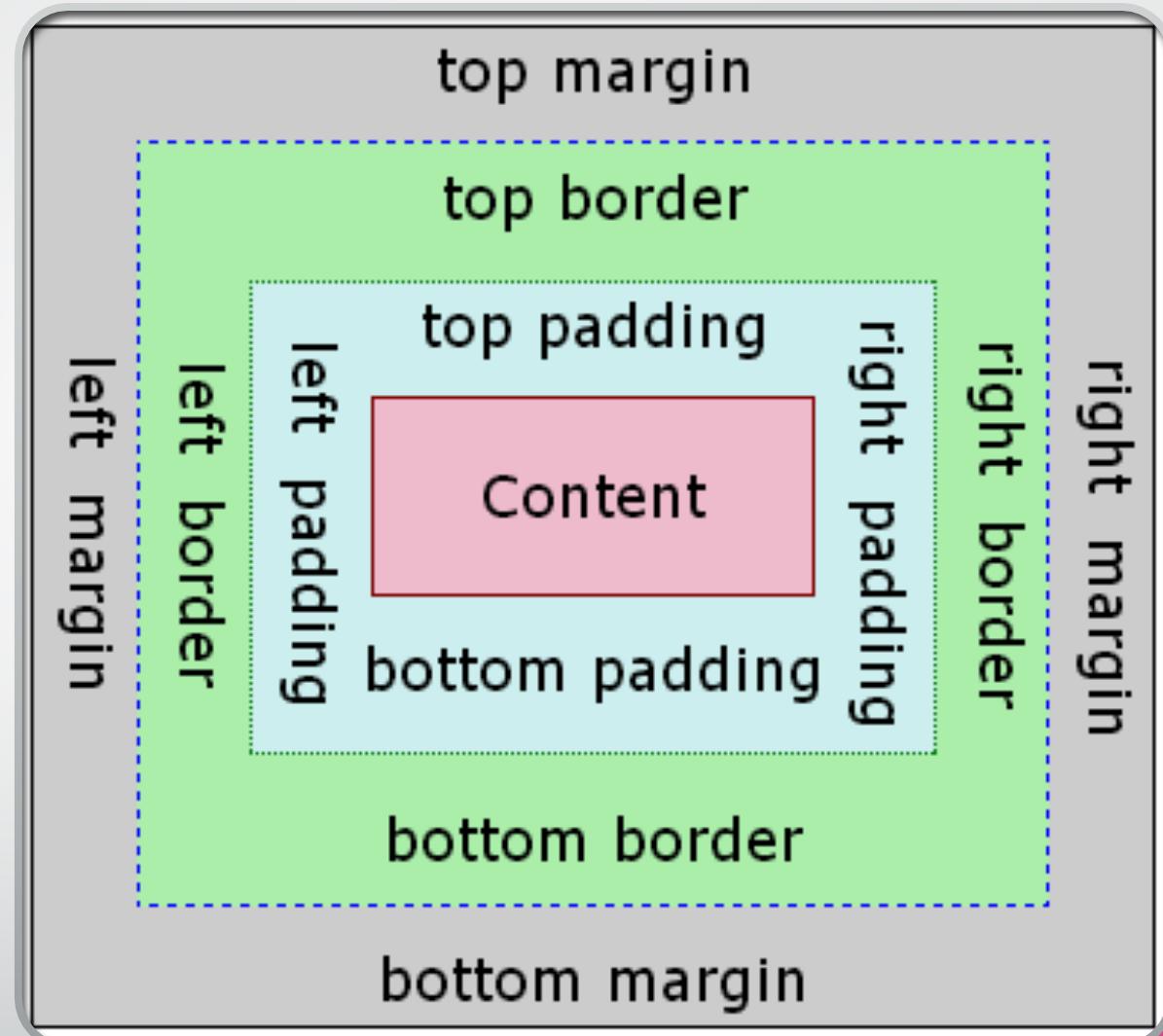
Improved debugging of missing stylesheets

Find and fix issues with missing stylesheets with ease.

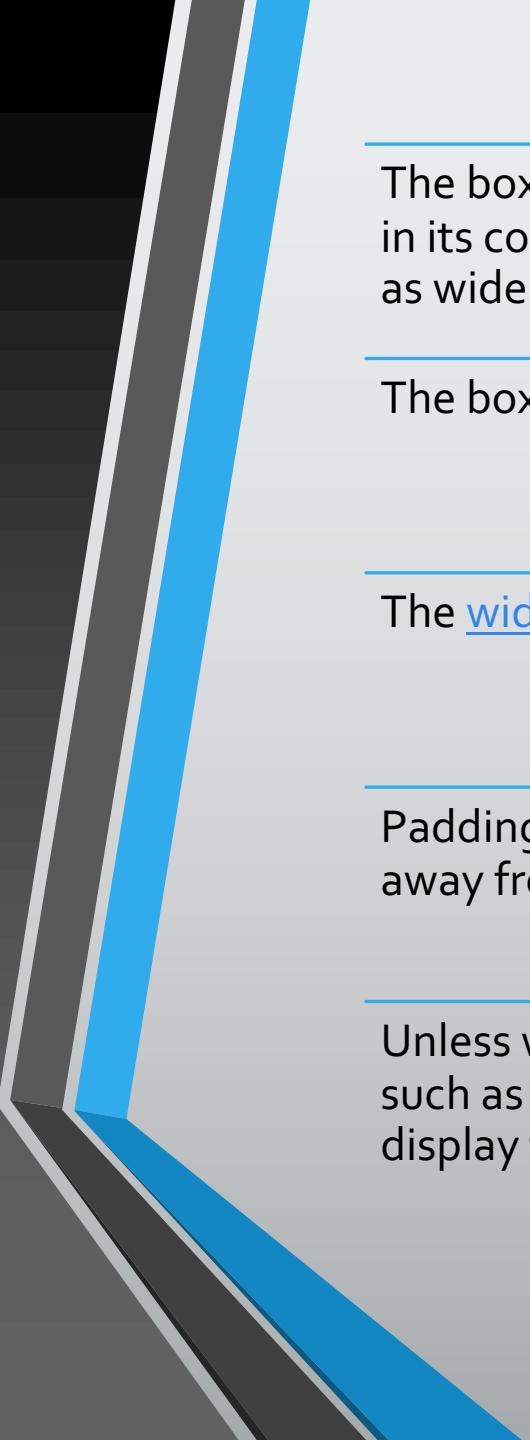


Box Model

- How browser defines area around element.
- Allows style sheet to set size of each area.
- padding-left: 10px
padding: 10px (sets top,right,bottom,left all to 10px)
- Display Types: Block or Inline



Box Model Layout Block Type



The box will extend in the inline direction to fill the space available in its container. In most cases this means that the box will become as wide as its container, filling up 100% of the space available.

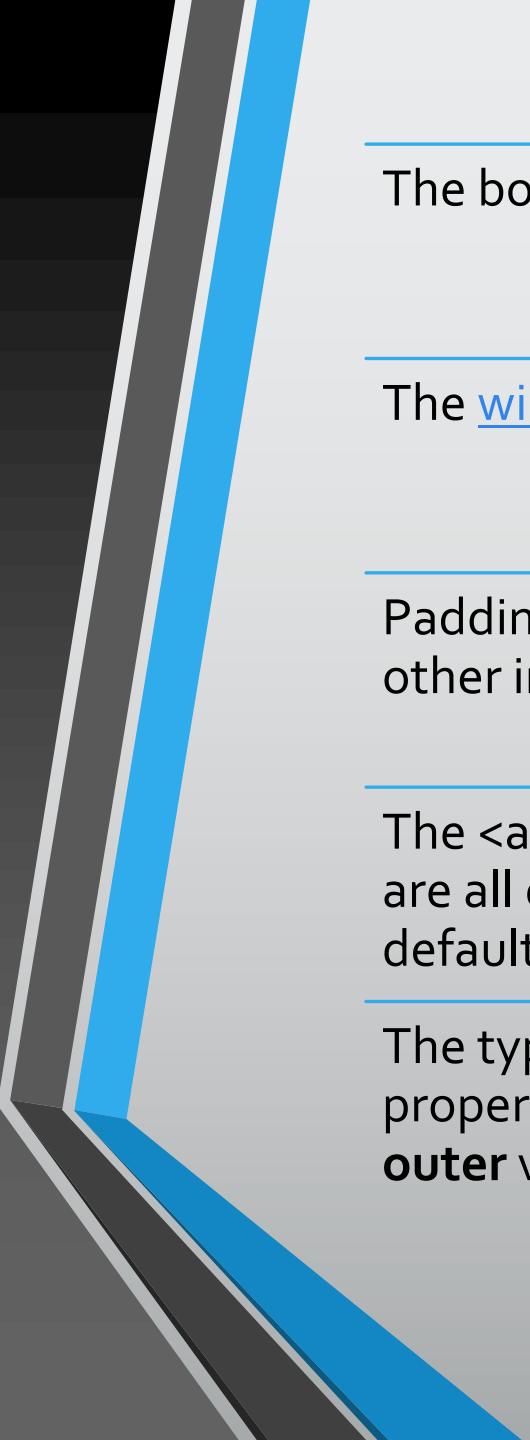
The box will break onto a new line.

The [width](#) and [height](#) properties are respected.

Padding, margin and border will cause other elements to be pushed away from the box

Unless we decide to change the display type to inline, elements such as headings (e.g. `<h1>`) and `<p>` all use block as their outer display type by default.

Box Model Layout Inline Type



The box will not break onto a new line.

The [width](#) and [height](#) properties will not apply.

Padding, margin and borders will apply but will not cause other inline boxes to move away from the box.

The `<a>` element, used for links, ``, `` and `` are all examples of elements that will display inline by default.

The type of box applied to an element is defined by [display](#) property values such as `block` and `inline`, and relates to the **outer** value of `display`.

Display

Defines "how" an element is displayed.

Can be used to force an element to be a box (eg: span is not a box by default)

can be used to control visibility. Then via code we can display/hide elements.

#error {display: none} -> element will NOT be displayed, layout **NOT** affected

#error {visibility: hidden} -> element not displayed **will** affect layout

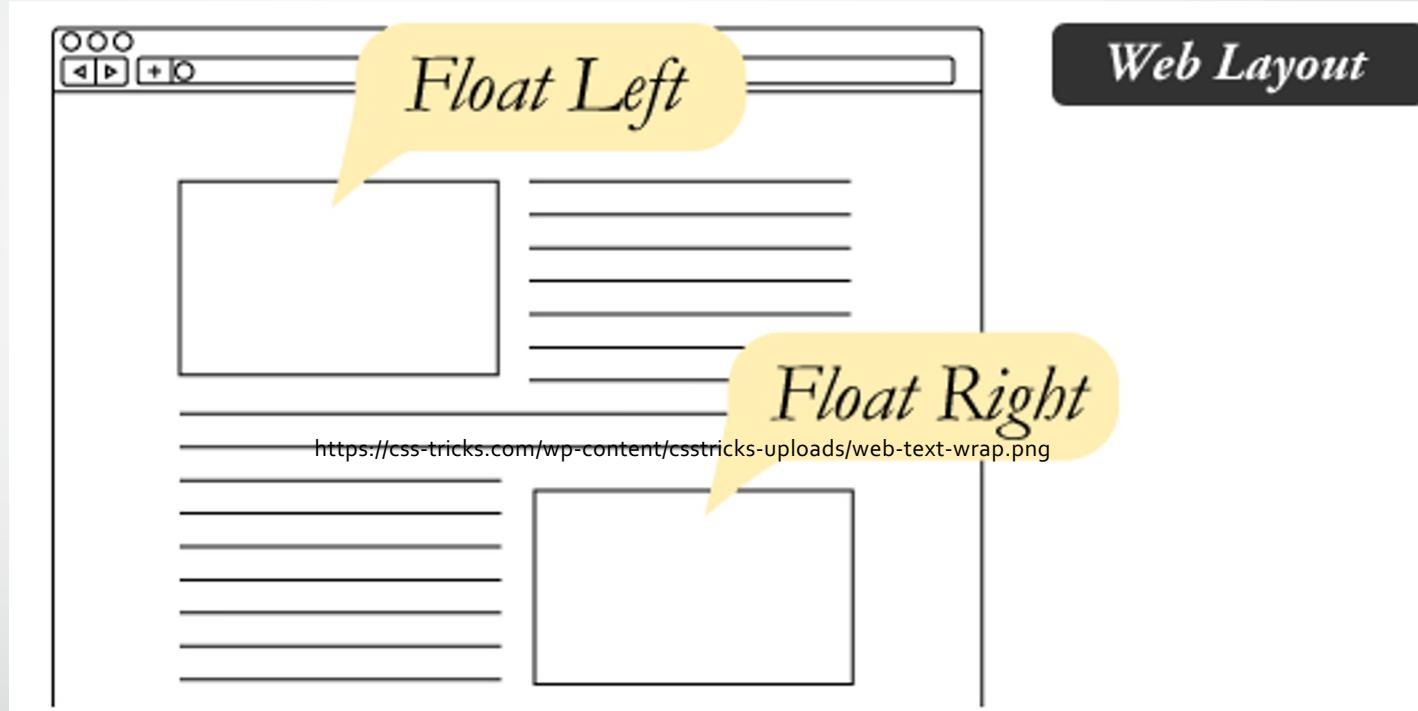


Float

Basic use is to "float" text around an image.

Frequently used to create screen layouts.

Clear: property to force browser to define areas where content is not allowed to Float into.



Centering Text

- .center {
- margin: auto;
- width: 50%;
- border: 3px solid green;
- padding: 10px;
- } -> MUST HAVE WIDTH set
- http://www.w3schools.com/css/css_align.asp

<https://css-tricks.com/centering-css-complete-guide/>



CSS for this course

- You need to be conversant with basic CSS
- This is not a course in web design but web application design.
- You need to understand:
 - main selectors
 - Some pseudo-class and pseudo-element selectors
 - Box Model
 - Float
 - Syntax and use (inline, internal, external)
 - Basic CSS commands (colors, font size, borders, alignment)

Forms & Bootstrap

Forms

It used to be the heartbeat of information gathering was forms.

User input areas to get data from user.

This has changed dramatically with the advent of JavaScript (look at GMAIL, it does not use forms) but forms still account for a lot of data gathering for Web Applications.

Form -> Area to gather user input (with inputs) and SUBMIT the data to a server for processing.



Initial Request

Name	<input type="text"/>
Age	<input type="text"/>
State	<input type="text"/>

Lookup page
Return Form

url

html

url +
Key/Value

Lookup page
Return Form

html

new page

Web 1.0

Form Based Applications



Form Example

```
<form method='POST' action='userdata.php' id='myForm'>  
  
<div class="form-group">  
  
    <label for="username" >User Name:</label>  
  
    <input type="text" class="form-control" name = "username"  
          id="username">  
  
    </div>  
  
<input type='hidden' name='uid' value='johnsok9'>  
  
<input type='submit' name='Submit' Value='GO'>  
  
</form>
```

Notes

Hidden fields

```
<input type='hidden' name='uid' value='johnsok9'>
```

Used to send data to an application but not have it visible on page.

We frequently put a hidden value on the field for the command. If a user submits a form with the enter button, then the submit button is NOT sent since the submit was not clicked.

eg:

```
<input type='hidden' name='cmd' value='AddUser'>
```

Hidden Field

DO NOT PUT CONFIDENTIAL INFORMATION IN HIDDEN FIELD

While users do not immediately see the value, it is in the source

AND USERS CAN CHANGE IT!!!!!!!!!!!!!!



Text Input NAME field required

```
<input type='text' name='username' id='for-css' value='default' required>  
  
<input type='password' name='password'>  
  
<textarea rows='10' cols='10'...>DEFAULT TEXT</textarea>  
  
<input type='submit' value='Submit Form'>
```

- ID is for referencing the DOM element.
- NAME is for the browser to use to create the KeyValue Pair.
- If No name present, the browser will not send the field.

Radio

```
<input type='radio' name='division' id="div1" value='CEC' checked>CEC<br>
```

```
<input type='radio' name='division' id="div2" value='CAS'>CAS<br>
```

....

Same names will be grouped together

Checkbox

```
<input type="checkbox" id="Miami" name="miami" value="yes"> Affiliated  
with Miami<br>
```

```
<input type="checkbox" id="current" name="current" value="yes"> Current  
Student<br>
```

Button

```
<input type="button">
```

Input attributes

readonly

disabled (value not sent when submitting form)

Size

maxlength -> CAUTION CAUTION

Live Form Examples

<https://ceclnxo1.cec.miamioh.edu/~johnsokg/cse383/lecture4/>

BootStrap

A popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first web sites.

Set of design elements

Grid, Typography, Images, Alerts....

Getbootstrap.com



Reference Material

Pseudo Class

A pseudo-class is used to define a special state of an element.

For example, it can be used to:

- Style an element when a user mouses over it
- Style visited and unvisited links differently
- Style an element when it gets focus

```
/* mouse over link */  
  
a:hover {  
    color: #FF00FF;  
}
```

Pseudo Element

A CSS **pseudo-element** is a keyword added to a selector that lets you style a specific part of the selected element(s).

For example, `::first-line` can be used to change the font of the first line of a paragraph.

```
/* The first line of every <p> element. */  
p::first-line {  
    color: blue;  
    text-transform: uppercase;  
}
```

Simple Layout with DIVs

- <http://ceclnx01.cec.miamioh.edu/~johnsok9/383/css1/div.html>
- Notes:
 - In #MainPage I set width manually and then use margin: auto to tell browser to make both left and right margins equal, hence centering page
 - I make Banner, h1 and h2 all use centered text on one line
 - Footer has a thin border applied

Css examples courtesy of Dr. Campbell

Layout with Menu

<http://ceclnx01.cec.miamioh.edu/~johnsok9/383/css1/menu.html>

Example of printing - hiding banner

<http://ceclnxo1.cec.miamioh.edu/~johnsokg/383/css1/divPrint.html>

FlexBoxes

Powerful layout tool

Display Property can be used to hide, set block level layout

display property: flex defines a flex "container"

Container has properties -> flex-direction, flex-wrap, flex-flow, and others

reference: https://www.w3schools.com/css/css3_flexbox.asp

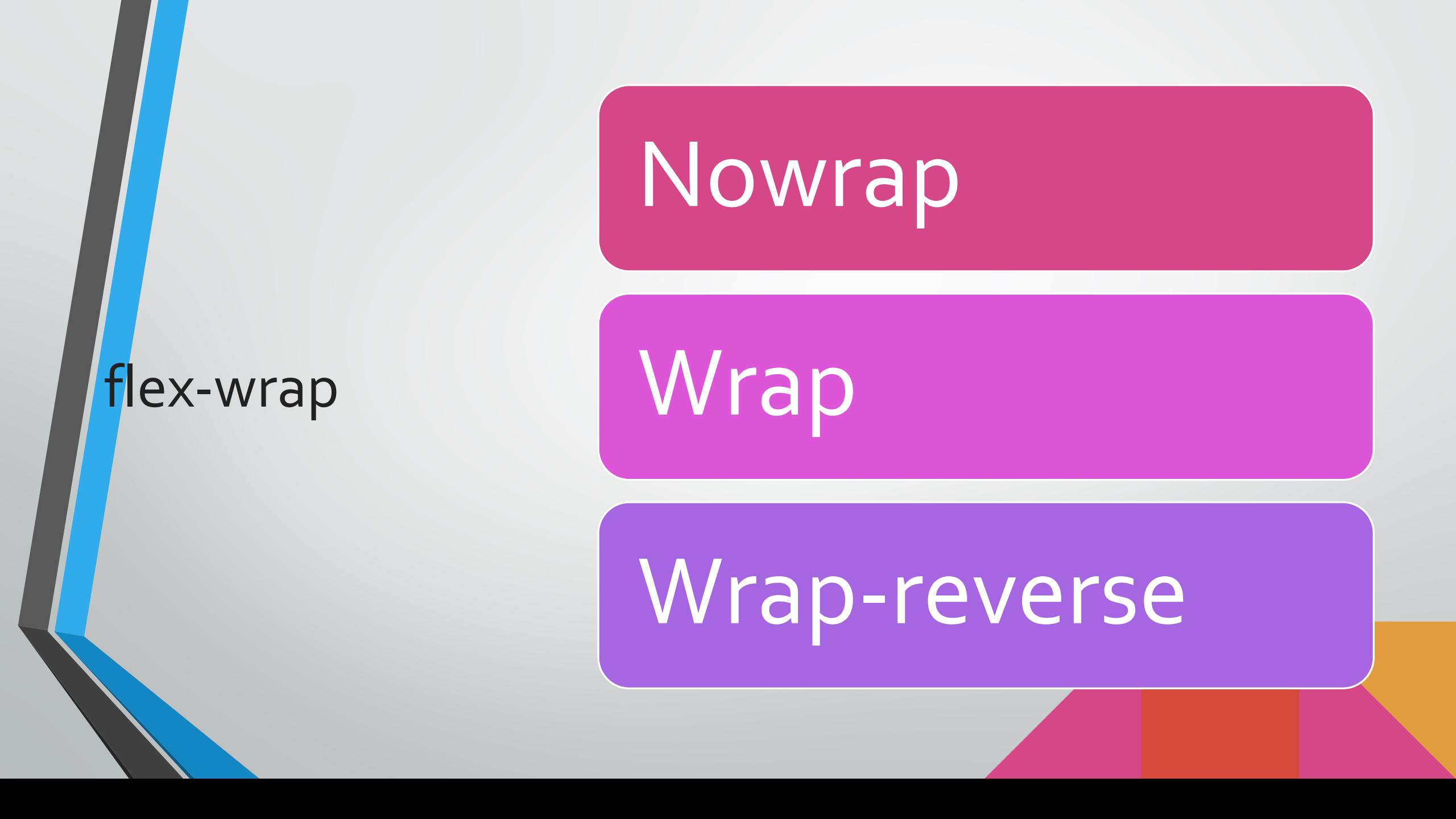
flex-direction

Row

Row-reverse

Column

Column-reverse



flex-wrap

Nowrap

Wrap

Wrap-reverse

Flex Example

- <http://ceclnx01.cec.miamioh.edu/~johnsok9/383/css1/flex.html>
- Uses flex box column for main layout
- content uses flexbox row
- Great tutorial on flexboxes –
<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

Children

Direct
children

select only immediate children of div tags

div > p

div p selects all p's inside a div

specific
children

this is kinda cool, select the nth child of a tag

li:nth-child(2) -> select the second child in the list

li:nth-child(2n) -> select all even rows

li:
nth-
child(2n){

background-
color: grey;

}

Positioning

Normally browsers layout blocks in a "flow" pattern

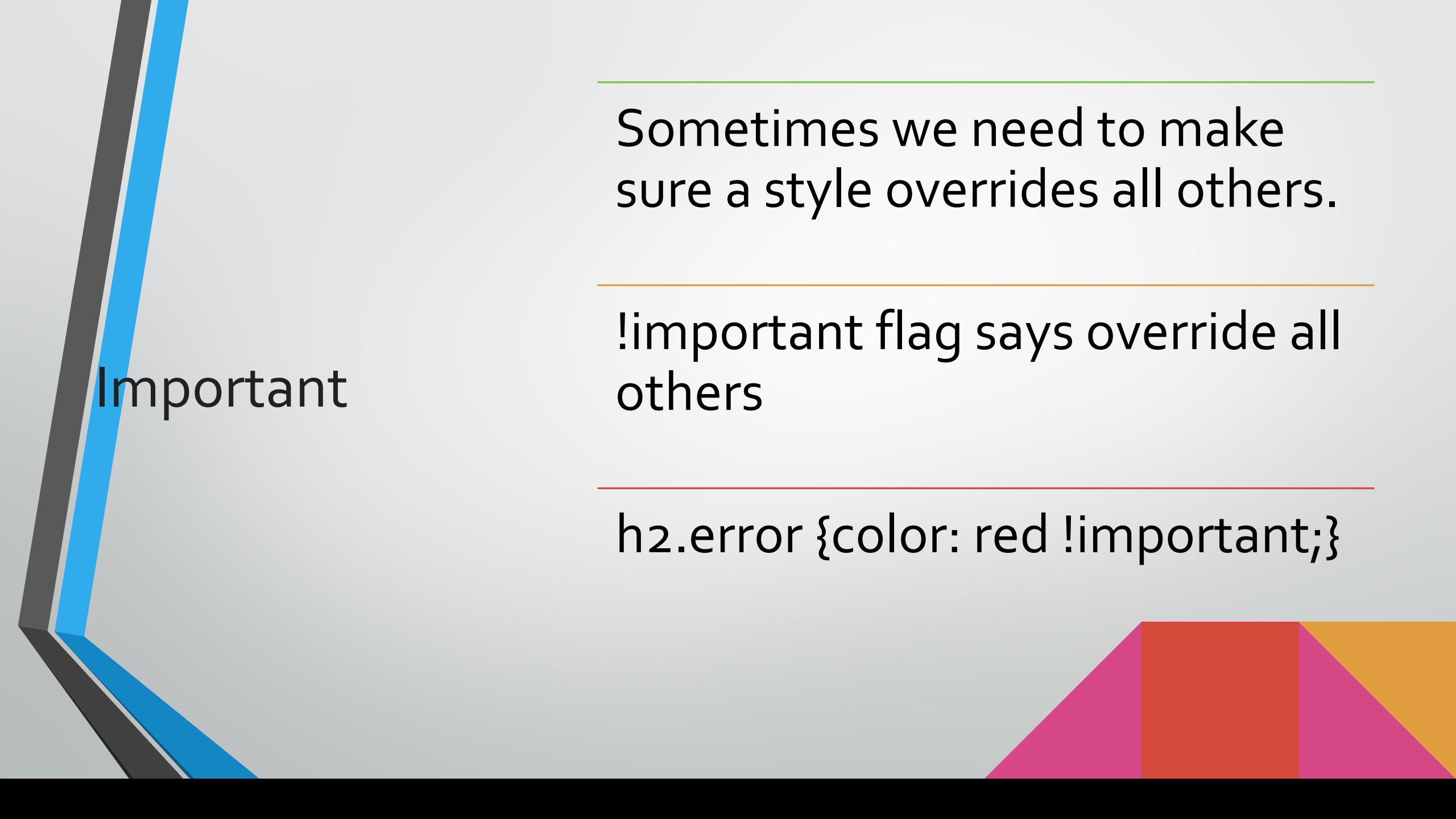
Static -> Default

Relative -> adjust relative to normal position. Eg: move left 5 pixels

Absolute position tells the browser to put this object relative to the DOM

Fixed position tells the browser to put this object relative to the screen.

Example: <http://ceclnx01.cec.miamioh.edu/~johnsok9/383/css1/position.html>



Important

Sometimes we need to make sure a style overrides all others.

!important flag says override all others

h2.error {color: red !important;}

Hover Effect

Change styles when mouse goes over:

```
.info:hover {  
    background-color:red;  
    transition: all 1.5s ease;  
    font-size: +3em;  
}
```

<http://ianlunn.github.io/Hover/>

Link states

Default links changes color when visited, active,etc

a:link -> controls links not clicked on

a:visited -> controls links that were visited

"Browsers limits the styles that can be set for a:visited links, due to security issues. Allowed styles are:

- color
- background-color
- border-color (and border-color for separate sides)
- outline color
- column-rule-color
- the color parts of fill and stroke " (w3schools)

Before and after

This mixes control between html and css. It can be done – but it can get confusing very fast!

```
h2:before {  
    content: "Read: ";  
}  
  
and  
  
h2:after {  
    content: "Read More";  
}
```



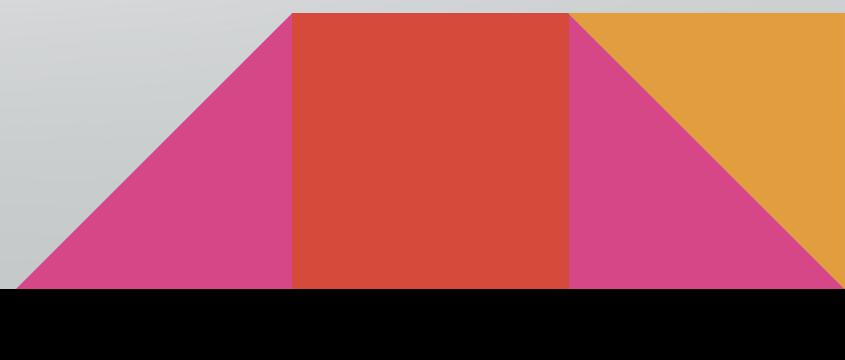
Force
upper or
lower case

text-transform:

uppercase

lowercase

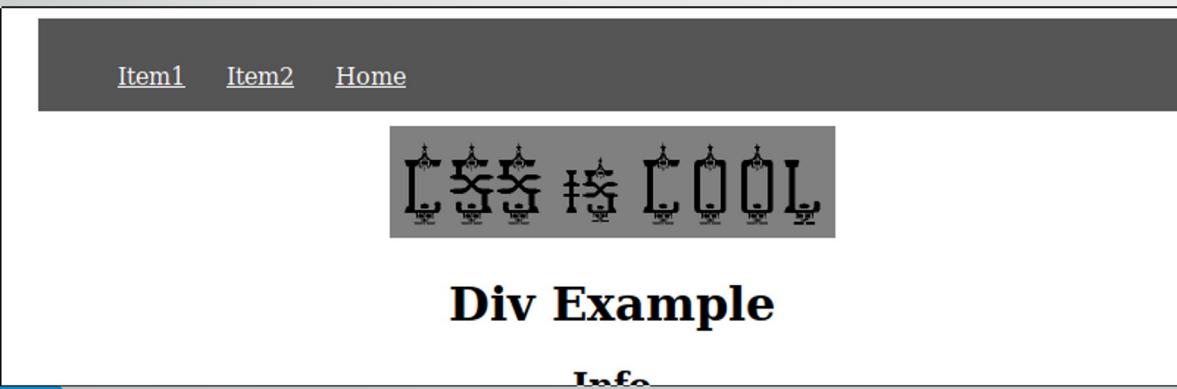
capitalize



Navigation/Menu

<http://ceclnxo1.cec.miamioh.edu/~johnsok9/383/css1/menu.html>

```
<div id='menu'>
  <ul>
    <li><a href='http://miamioh.edu' target='_blank'>Item1</a></li>
    <li><a href='http://miamioh.edu' target='_blank'>Item2</a></li>
    <li><a href='http://miamioh.edu' target='_blank'>Home</a></li>
  </ul>
</div> <!-- end div menu-->
```



```
#menu {
  margin:10px 10px 10px 10px;
  padding:0;
  overflow:hidden;
  background-color: #555;
}
```

```
#menu li {
  float: left;
  color: white;
  text-align: center;
  display: block;
  padding: 14px 14px;
}
```

```
#menu li a:link {
  color: white;
```

```
#menu li a:visited {
  color: white;
```

```
#menu li a:hover{
  color: #faa;
}
```

Responsive Design

- Responsive design using ONLY html and CSS
- http://www.w3schools.com/css/css_rwd_intro.asp

Printing and Media Type

- http://www.w3schools.com/css/css3_mediaqueries.asp
- I use this for creating Printable Output -> hide menu, ads, that type of material

CSS and Print

- <https://www.smashingmagazine.com/2015/01/designing-for-print-with-css/>
- *Skim this article to get sense what can be done.*
- It is possible to use CSS for more quality printing
- @page rule -> specify page sizing and details
- [Understand Page Margins ->](#)



End of Presentation