

Lecture 12

408/508 *Computational
Techniques for Linguists*

Today's Topics

- More html tags
- X11 colors
- UTF-8
- Introduction to CSS
- inline style
- Tabs
- Homework 5

HTML

- inline styling applied to text elements:
- e.g. (inline text) vs. <div> (block text)
 - <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/span>
 - ` ... `
 - font-size: Xpx
 - font-family: name, name ...
 - color: name (or hex RGB) e.g. #00CC00
 - background-color: name (or RGB)
 - text-align: left|right|center
 - **note:** serif, sans-serif, monospace are generic font families
 - **note:** X11 color names are okay,
http://en.wikipedia.org/wiki/X11_color_names

X11 Colors

- https://en.wikipedia.org/wiki/X11_color_names#Color_name_chart

	Name ↕	Hex (RGB) ↕	Red (RGB) ↕	Green (RGB) ↕	Blue (RGB) ↕	Hue (HSL/HSV) ↕	Satur. (HSL) ↕	Light (HSL) ↕	Satur. (HSV) ↕	Value (HSV) ↕
	Alice Blue	#F0F8FF	94%	97%	100%	208°	100%	97%	6%	100%
	Antique White	#FAEBD7	98%	92%	84%	34°	78%	91%	14%	98%
	Aqua	#00FFFF	0%	100%	100%	180°	100%	50%	100%	100%
	Aquamarine	#7FFFD4	50%	100%	83%	160°	100%	75%	50%	100%
	Azure	#F0FFFF	94%	100%	100%	180°	100%	97%	6%	100%
	Beige	#F5F5DC	96%	96%	86%	60°	56%	91%	10%	96%
	Bisque	#FFE4C4	100%	89%	77%	33°	100%	88%	23%	100%
	Black	#000000	0%	0%	0%	0°	0%	0%	0%	0%
	Blanched Almond	#FFEBCD	100%	92%	80%	36°	100%	90%	20%	100%
	Blue	#0000FF	0%	0%	100%	240°	100%	50%	100%	100%
	Blue Violet	#8A2BE2	54%	17%	89%	271°	76%	53%	81%	89%
	Brown	#A52A2A	65%	16%	16%	0°	59%	41%	75%	65%
	Burlywood	#DEB887	87%	72%	53%	34°	57%	70%	39%	87%

UTF-8

```
<head>
...
<meta charset="utf-8">
...
</head>
```

- The `<meta>` [HTML](#) element represents [metadata](#) that cannot be represented by other HTML meta-related elements, like [<base>](#), [<link>](#), [<script>](#), [<style>](#) or [<title>](#).
- `charset`
 - This attribute declares the document's character encoding. If the attribute is present, its value must be an ASCII case-insensitive match for the string "utf-8", because **UTF-8 is the only valid encoding for HTML5** documents. `<meta>` elements which declare a character encoding must be located entirely within the first 1024 bytes of the document.

HTML

https://developer.mozilla.org/en-US/docs/Web/HTML/Inline_elements

HTML (**Hypertext Markup Language**) elements historically were categorized as either "block-level" elements or "inline-level" elements. Since this is a presentational characteristic it is nowadays specified by CSS in the [Flow Layout](#). Inline elements are those which only occupy the space bounded by the tags defining the element, instead of breaking the flow of the content. In this article, we'll examine HTML inline-level elements and how they differ from [block-level elements](#).

■ An inline element does not start on a new line and only takes up as much width as necessary.

HTML

Conceptual differences

In brief, here are the basic conceptual differences between inline and block-level elements:

Content model

Generally, inline elements may contain only data and other inline elements. You can't put block elements inside inline elements.

Formatting

By default, inline elements do not force a new line to begin in the document flow. Block elements, on the other hand, typically cause a line break to occur (although, as usual, this can be changed using CSS).

- Block:
 - lists
 - headings:
 - `<h1> ... <h4>`
 - `<p>`
 - `<div>`
 - general container

HTML

- Tables:

- `<table> ... </table>`
- `<tr> ... </tr>`
- `<th> ... </th>`
- `<td> .. </td>`
- Attributes:
 - `border="size"`
 - `colspan="number"`
 - `style="...;..."`
 - `border: width style color`
 - `border-width: top right bottom left`
 - `border-style: top right bottom left`
 - `border-color: top right bottom left`
 - `border-collapse: collapse|separate`
 - `padding: size`
 - `text-align: left|center|right`
 - `vertical-align: top|middle|bottom`
 - `width, height`
- Newer stuff:
 - `<thead> ... </thead>`
 - `<tbody> .. </tbody>`

table row

table heading element

table data (one cell)

e.g. 1px

e.g. 2 (span next two columns)

also border-left, border-top, border-right, etc.

e.g. type=solid|dotted|dashed|double|none

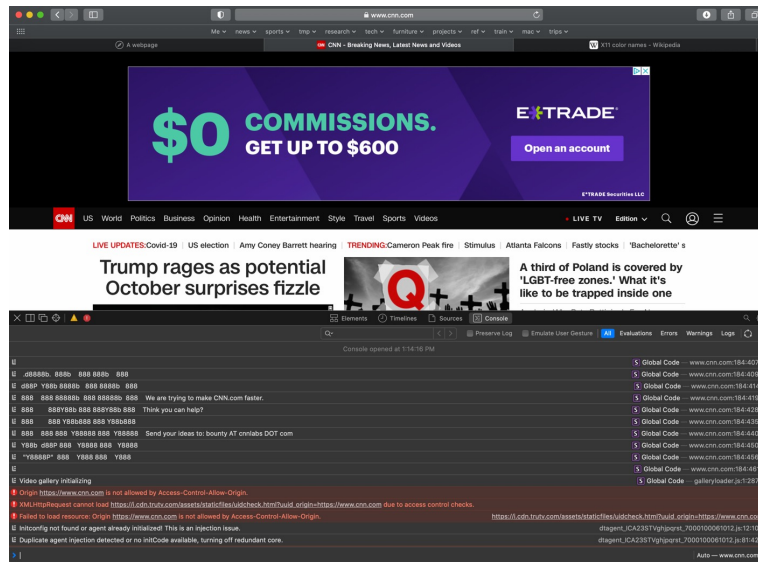
e.g. 100px or 100%

HTML

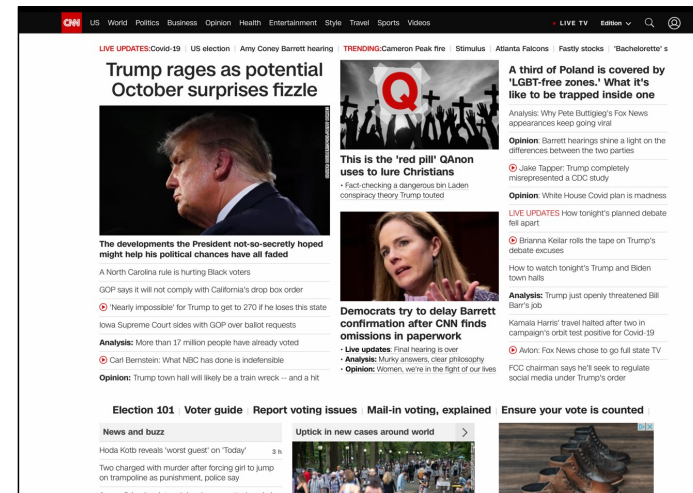
- General "chunks" of html:
 - `<div style="..."> ... </div>`
 - a division (block-level)
 - ` ... `
 - small chunks (inline)
- (optional) **unique identifier** `id=`
 - used to refer to the "chunk" in CSS or DOM
 - `<div style="..." id="..."> ... </div>` a division
 - ` ... ` small chunks of inline text

CNN Webpage

Look at the errors on loading cnn.com!



Look at the bounty offered to developers! ↑



CSS

```
body {  
  background-color: lightblue;  
}  
  
h1 {  
  color: white;  
  text-align: center;  
}  
  
p {  
  font-family: verdana;  
  font-size: 20px;  
}
```

Selector	Example	Example description
<i>.class</i>	.intro	Selects all elements with class="intro"
<i>#id</i>	#firstname	Selects the element with id="firstname"
<i>*</i>	*	Selects all elements
<i>element</i>	p	Selects all <p> elements
<i>element,element</i>	div, p	Selects all <div> elements and all <p> elements
<i>element element</i>	div p	Selects all <p> elements inside <div> elements
<i>element>element</i>	div > p	Selects all <p> elements where the parent is a <div> element

<https://www.w3schools.com/css/>

CSS

- Inline (could go inside `<head>` ... `</head>`):
 - `<style> ... </style>`
- Good way to see what's possible.
 - Google, e.g. `css table style`
 - https://www.w3schools.com/css/css_table.asp

CSS Tables

[< Previous](#) [Next >](#)

The look of an HTML table can be greatly improved with CSS:

Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Berglunds snabbköp	Christina Berglund	Sweden
Centro comercial Moctezuma	Francisco Chang	Mexico
Ernst Handel	Roland Mendel	Austria
Island Trading	Helen Bennett	UK
Königlich Essen	Philip Cramer	Germany
Laughing Bacchus Winecellars	Yoshi Tannamuri	Canada
Magazzini Alimentari Riuniti	Giovanni Rovelli	Italy

[Try it Yourself »](#)

Tabs

- Read:

- https://www.w3schools.com/howto/howto_js_tabs.asp

Tabs

Tabs are perfect for single page web applications, or for web pages capable of displaying different subjects:

London Paris Tokyo

London

London is the capital city of England.

Tabs

Sandiway Fong

About me

Research

Software

Classes

UA-related

Cycling

Running

Trains

Coffee

Picture courtesy of Leslie Hawthorne Klingler.

My work intersects computer science and linguistics.

Principally, I'm interested in the computational modeling of linguistic theory.

I work at the [University of Arizona](#) (UA).

I direct and teach in the in-person Master's Program in Human Language Technology ([HLT](#))

for the [Department of Linguistics](#).

I'm also a member of the UA [Cognitive Science Program](#) and [SLAT](#).

I have a long-term collaboration with linguistics faculty at Osaka Kyoiku University ([OKU](#)) in Japan.

I work primarily with professors [Jason Ginsburg](#) and [Hiroshi Terada](#).

(The two universities have a formal [academic exchange agreement](#).)



Tabs

Sandiway Fong

- About me
- Research
- Software
- Classes
- UA-related
- Cycling
- Running
- Trains
- Coffee

Picture courtesy of Leslie Hav

Recent research projects, talks and papers.

On the nature of FormSet. [Abstract](#)
Fong, S. & M. Oishi.
FIND-Workshop 2023, December 11-12, Göttingen, Germany.


The Strong Minimalist Thesis (SMT): Form Copy (FC) and the Serial Verb Construction (SVC).
Fong, S., J. Ginsburg, M. Matsumoto & H. Terada.
In *Proceedings of Japanese/Korean Linguistics 30* (JK30), March 11th–13th 2023.
Simon Fraser University, BC, Canada, pp.411-421, CSLI.
[Link](#)

On the computational modeling of English relative clauses.
Fong, S. & J. Ginsburg. (in press)
Open Linguistics, vol. 9, no. 1, 2023. Article number: 20220246, De Gruyter Open Access.
[Link coming soon!](#)

Project: Minimalist Machine: computer implementation of a **generative** theory in the Minimalist Program (MP).
[See](#) theory implemented and many examples of derivations.
(Joint work with Jason Ginsburg of Osaka Kyoiku University.)

Parsing with Box theory: to be updated soon.

Linguistic Theory and Deep Learning.
Generative AI: [ChatGPT and Natural Language Understanding](#)



Tabs

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width,
initial-scale=1">
<style>
body {font-family: Arial;}
/* Style the tab */
.tab {
  overflow: hidden;
  border: 1px solid #ccc;
  background-color: #f1f1f1;
}
/* Style the buttons inside the tab */
.tab button {
  background-color: inherit;
  float: left;
  border: none;
  outline: none;
  cursor: pointer;
  padding: 14px 16px;
  transition: 0.3s;
```

```
  font-size: 17px;
}
/* Change background color of buttons on hover */
.tab button:hover {
  background-color: #ddd;
}
/* Create an active/current tablink class */
.tab button.active {
  background-color: #ccc;
}
/* Style the tab content */
.tabcontent {
  display: none;
  padding: 6px 12px;
  border: 1px solid #ccc;
  border-top: none;
}
</style>
</head>
```


Tabs

```
<body>
<h2>Tabs</h2>
<p>Click on the buttons inside the tabbed menu:</p>
<div class="tab">
  <button class="tablinks" onclick="openCity(event,
'London')">London</button>
  <button class="tablinks" onclick="openCity(event,
'Paris')">Paris</button>
  <button class="tablinks" onclick="openCity(event,
'Tokyo')">Tokyo</button>
</div>
<div id="London" class="tabcontent">
  <h3>London</h3>
  <p>London is the capital city of England.</p>
</div>
<div id="Paris" class="tabcontent">
  <h3>Paris</h3>
  <p>Paris is the capital of France.</p>
</div>
<div id="Tokyo" class="tabcontent">
  <h3>Tokyo</h3>
  <p>Tokyo is the capital of Japan.</p>
</div>
```

```
<script>
function openCity(evt, cityName) {
  var i, tabcontent, tablinks;
  tabcontent = document.getElementsByClassName("tabcontent");
  for (i = 0; i < tabcontent.length; i++) {
    tabcontent[i].style.display = "none";
  }
  tablinks = document.getElementsByClassName("tablinks");
  for (i = 0; i < tablinks.length; i++) {
    tablinks[i].className = tablinks[i].className.replace("
active", "");
  }
  document.getElementById(cityName).style.display = "block";
  evt.currentTarget.className += " active";
}
</script>
</body>
</html>
```

Homework 5

- Build yourself a webpage
- Have these elements (*at least*):
 - base64 picture, other images can be URLs
 - Tabs (*like mine*)
 - write something for each tab
 - style: something different from the defaults, e.g. font, color, background etc.
- Due Sunday midnight
 - send PDF
 - also include your .html file as an attachment
 - Subject: 408/508 Homework 5 *YOUR NAME*