



CMPSC 302

WEB DEVELOPMENT



Inline vs. block elements

- * Elements are of two types:

- * Block

- * Inline

- * They are classified as:

- * Block elements:

- * Automatically take up/start on new line

- * Expand to fit all available width

- * Inline elements:

- * Do not take up or start a new line—they are, instead “in line” with content

- * Take up only as much space as-is necessary



Block elements

`<address>`

`<article>`

`<aside>`

`<blockquote>`

`<canvas>`

`<dd>`

`<div>`

`<dl>`

`<dt>`

`<figure>`

`<fieldset>`

`<footer>`

`<form>`

`<h1>-<h6>`

`<header>`

`<hr>`

``

`<main>`

`<nav>`

`<noscript>`

``

`<p>`

`<pre>`

`<section>`

`<table>`

`<tfoot>`

``

`<video>`



Inline elements

<a>

<abbr>

<acronym>

<bdo>

<big>

<button>

<cite>

<code>

<dfn>

<i>

<input>

<kbd>

<label>

<map>

<object>

<output>

<q>

<samp>

<script>

<select>

<small>

<sub>

<sup>

<textarea>

<time>

<tt>

<var>



What does this mean for us?

- * Block elements are usually *organizational*

 - * `<section>`, `<div>`, `<h1>`...

- * Inline elements are typically *functional* or *stylistic*

 - * `<a>`

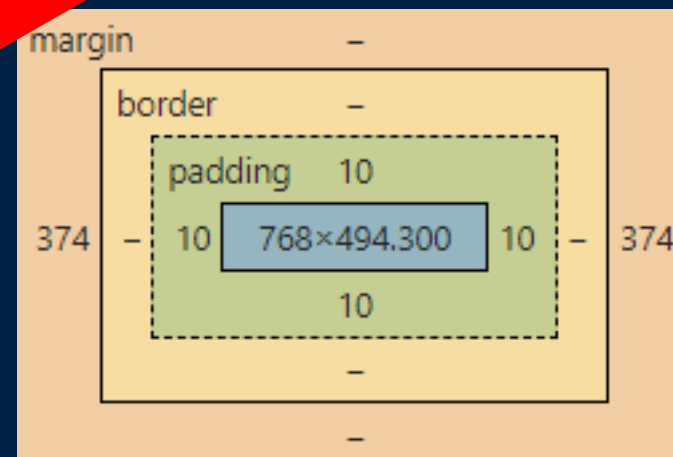
 - * ``, `<i>`

 - * ``, ``

- * They have implications for understanding how style will affect them



Not really a detour: the CSS Box Model





Even inline elements are a box

Lorem ipsum

Lorem ipsum dolor sit amet, consectetur adipiscing elit. *Sed tincidunt tellus orci*, a porta lectus vestibulum ac. Nam ut ex eu elit feugiat cursus. Curabitur et tempus tellus. Mauris **volutpat** magna ac leo volutpat congue. Fusce vulputate enim libero, placerat malesuada elit euismod eget. Aliquam eget lacus tristique, fermentum velit a, consectetur elit. Nam ultrices dolor ut *auctor lacinia*.

Integer ut consequat quam, ac sollicitudin odio. **Fusce efficitur sem lectus**, quis commodo dolor tincidunt iaculis. Aenean laoreet ex ut neque tristique dignissim. Mauris accumsan suscipit nunc vitae tempus. Fusce augue magna, dapibus ut leo ac, viverra bibendum quam. Proin interdum ipsum quis quam iaculis, eget bibendum est dignissim. Fusce iaculis nisi placerat lectus consectetur semper vel vitae ligula. In ultrices enim *ut dapibus hendrerit*. Duis accumsan, metus eu ullamcorper gravida, nisi purus lobortis odio, a semper magna sem sed turpis.

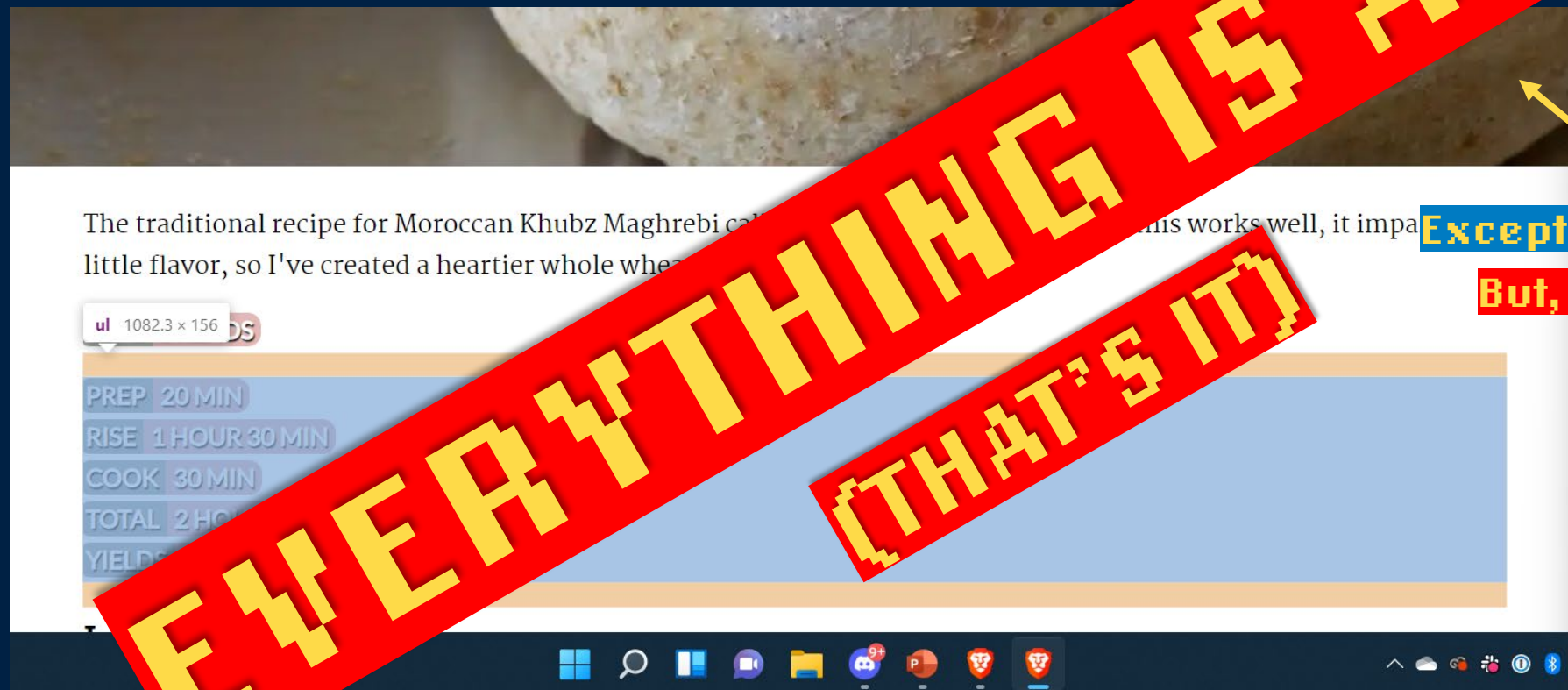
More information

For more information about "Lorem Ipsum..."

em 189.08 × 27.2



One more time with feeling



Except this, which is a bread

But, the image IS A BOX!



Professor, it looks like you have style

It's true.



But also...

- * We can style these boxes using *selectors*

- * Which, in this case, are really just:

- * Element tags

- * Element attributes

Again, just a “way of speaking”; a “language”

- * We use selectors in a special *syntax* and file called “CSS”:

- * **C**ascading

- * **S**tyle

- * **S**heets

- * They’re called “cascading” because that’s essentially what they do:
read from top to bottom

- * They are *additive*—they combine to form a “composite” style



``phasizing the point

- * To style some text, we're going to use the `` tag (new, but not really)
 - * `` stands for **emphasis**
- * By browser stylesheet default (your browser automatically uses one whether you like it or not):
 - * `` is italic type
- * Use this in `lorem-ipsum.html` to emphasize 2 random parts of a sentence
 - * Enclose the part you'd like to emphasize in `...`
- * Use `` in at least 2 places as well



Detour: Lorem Ipsum

- * Tradition started by some printer in the 15th century
- * Comes from *De finibus bonorum et malorum* (The Extremes of Good and Evil) by my guy Cicero
- * Used by designers for a long time as a placeholder text. “Greek” (put placeholder text in a document to see how it would look) and “Latin” (to see how it would look)
- * There are many variations of Lorem Ipsum, but the most common one is the “standard” Lorem Ipsum. (You are now the designer you’ve always wanted to be.)

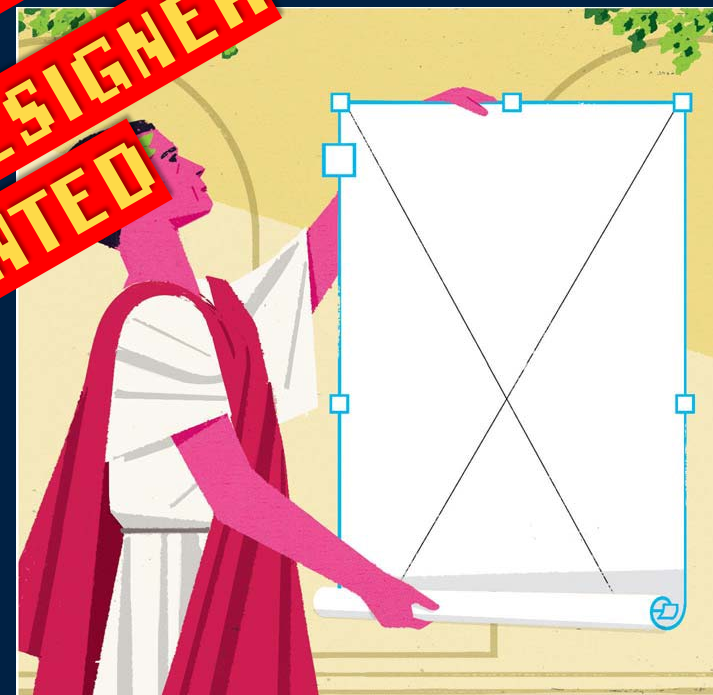


Image via PrintMag



So why `` and not `<i>`?

- * Also, why `` and not ``? (They *look* equivalent)
- * One tag in each pair is *semantic*
 - * ``, for example, means that text is a strong emphasis
 - * ``, for example, means something similar, but is probably “nicer”
 - * `` literally just means “bold this ish”
 - * `<i>` also means “italics, plz”
- * Only use `` or `` when really emphasizing something
 - * Else, use the decorative `` or ``
- * In terms of “lorem ipsum...” I have no idea what it means anyway



Styling it

- * If `` and `` are really just semantic, it may not necessarily mean that italics and heavier-weight type (i.e. bold) are the way
- * We can *change* how these tags work in our `style/lorem-style.css` file.
- * Let's make them different backgrounds and not italic or bold



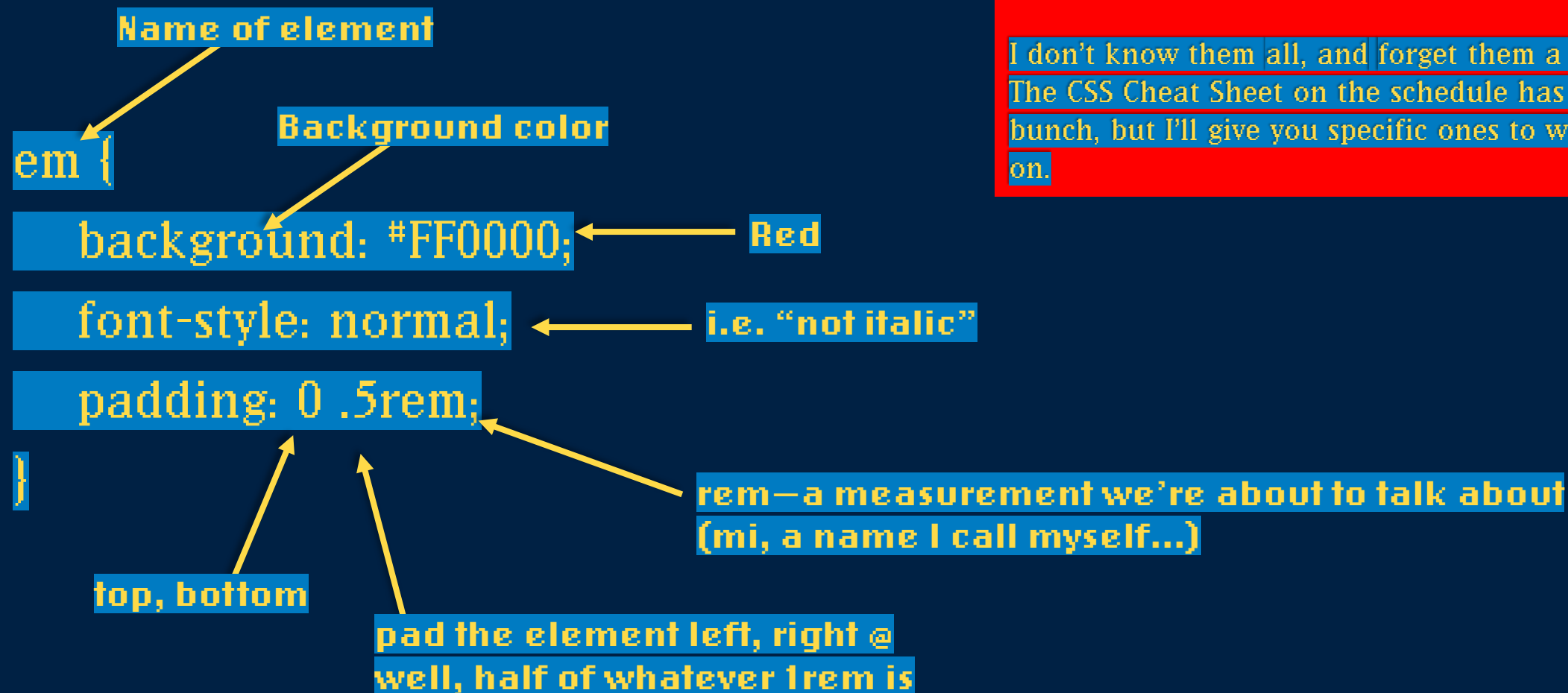
Styling it

```
em {  
  background: #FF0000;  
  font-style: normal;  
  padding: 0 .5em;  
}
```

```
strong {  
  background: #222222;  
  font-weight: 400;  
  padding: 0 .5em;  
  color: #FFFFFF;  
}
```



What the what?



There are a *ton* of CSS "properties" like:

background

font-style

Padding

I don't know them all, and forget them a lot. The CSS Cheat Sheet on the schedule has a bunch, but I'll give you specific ones to work on.



Brief look at sizing units

* In HTML there are several sizing units:

* pixels (px)

* points (pt)

* relative percentage (%)

* em (em)

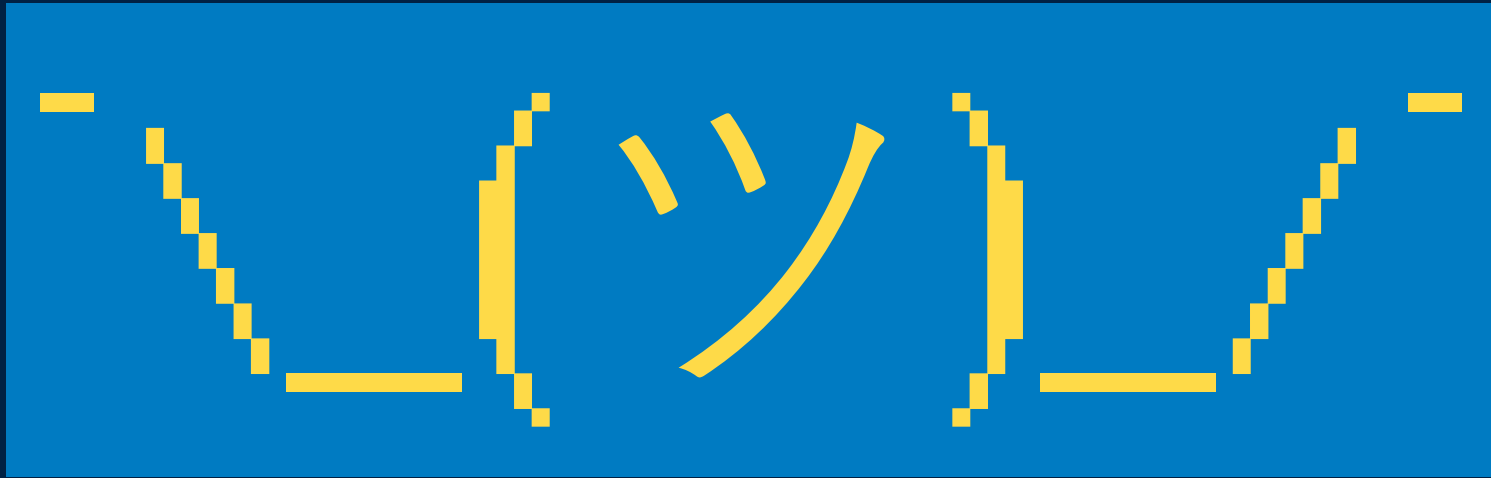
* Relative to whatever type size the element it's contained in has

* rem (rem)

* Relative to the entire document (base element—<html>)



Which do I use?



It depends.



For now...

- ✧ We're going to use **rem** to remain true to our root type size
- ✧ Look at our root element in lorem-style.css:

```
html {  
  font-size: 14pt;  
}
```

- ✧ This means our `` size is **7 pt**



CSS properties we're going to use this week

font-size the type size

font-family the type face (i.e. Times New Roman)

background the background color

height the height of an element

color the color of the type face

border-bottom the border on the bottom of an element

transform the ability to enact different transformations

padding the “stuffing” inside of an element

ENOUGH CHIT-CHAT

(LET'S DO STUFF)