

HTML/CSS 3

Uchicago Techteam

Specificity weights

- Determines which style takes precedence
 - $X(\text{num of id selectors}) - X(\text{class}) - X(\text{type})$
- $p \{ \}$: $0 - 0 - 1$
- $.\text{food} \{ \}$: $0 - 1 - 0$
- $\#\text{hotdog} \{ \}$: $1 - 0 - 0$

```
#hotdog {  
  color: red;  
}
```

1 – 0 – 0

```
.food {  
  color: blue;  
}
```

0 – 1 – 0

```
p {  
  color: black;  
}
```

0 – 0 – 1

Combining Selectors

```
<div class="hotdog">
  <p>...</p>
  <p>...</p>
  <p class="mustard">...</p>
</div>
```

```
.hotdog p {
  background: brown;
}
.hotdog p.mustard {
  background: yellow;
}
```

- Multiple selectors are read from right to left
 - Rightmost selector: key selector
 - The rest: pre-qualifiers
 - .hotdog p 0 – 1 – 1
 - Applies style to “p element WITHIN an element of class .hotdog”
 - p.mustard 0 – 1 – 1
 - If there is no space in between, this means “.mustard class that is ALSO p”
 - div a p 0 – 0 – 3
 - “p element WITHIN a WITHIN div”

- How would you create two buttons, one sending “danger” message and one sending “success”?

```
<a href="#" class="btn-danger">Danger!</a>
```

```
<a href="#" class="btn-success">Success!</a>
```

```
.btn-danger {
```

```
border: 1px solid black;
```

```
background-color: red;
```

```
}
```

```
.btn-success {
```

```
border: 1px solid black;
```

```
background-color: green;
```

```
}
```

```
<div class="btn"><a href="#" class="btn-  
danger">Danger!</a></div>
```

```
<div class="btn"><a href="#" class="btn-  
success"> Success!</a></div>
```

```
.btn {  
  border: 1px solid black;  
}
```

```
.btn-danger {  
  background-color: red;  
}
```

```
.btn-success {  
  background-color: green;  
}
```

Layering multiple classes

- Multiple classes can be used as a single attribute value, separated by spaces
- Modularity is important for reusability

```
<a href="#" class="btn btn-danger">Danger!</a>
```

```
<a href="#" class="btn btn-success">Success!</a>
```

```
<a href="#" class="btn-alt btn-danger">Danger!</a>
```

```
.btn {  
    border: 1px solid black;  
}  
  
.btn-danger {  
    background-color: red;  
}  
  
.btn-success {  
    background-color: green;  
}  
  
.btn-alt {  
    ...  
}
```

CSS properties: colors

- With keywords

black, maroon, purple, etc...

- With Hexadecimal colors (most common)

#000000 (= #000, black) ~ #ffffff (= #fff, white)

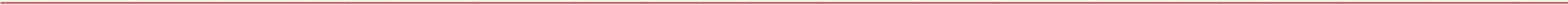
- With RGB/RGBa

rgba(0, 255, 0, 0.80%)

- With HSL/HSLa

hsla(24, 100%, 50%, .5)

**Chrome extensions,
Online sources
(color.adobe.com)
Could help color-picking**

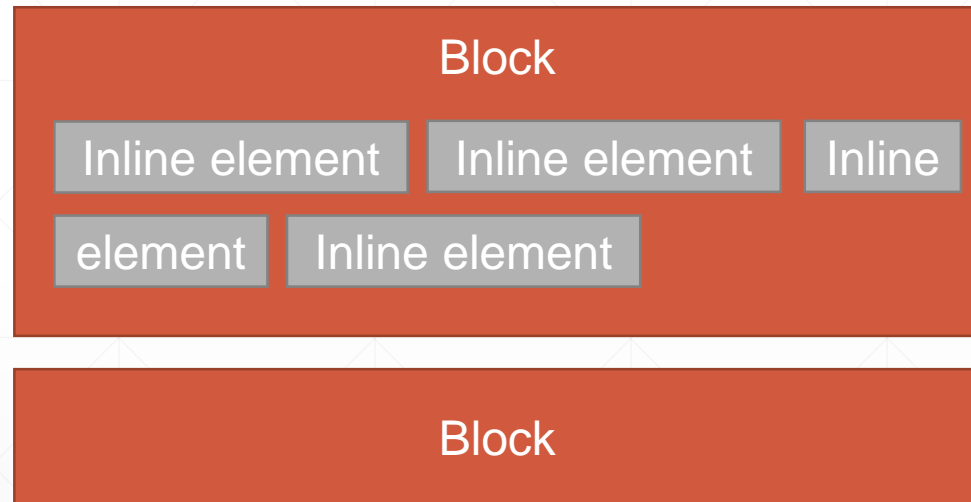


CSS properties: lengths

- Absolute lengths: pixels
 - Disadvantage: not flexible
- Relative length:
 - Relative to parent element: %
 - Relative to element's font size: em

```
p {  
  font-size: 14px;  
}  
  
.col {  
  width: 50%;  
}  
  
.banner {  
  font-size: 14px;  
  letter-spacing: 0.5em;  
}
```

CSS properties: display

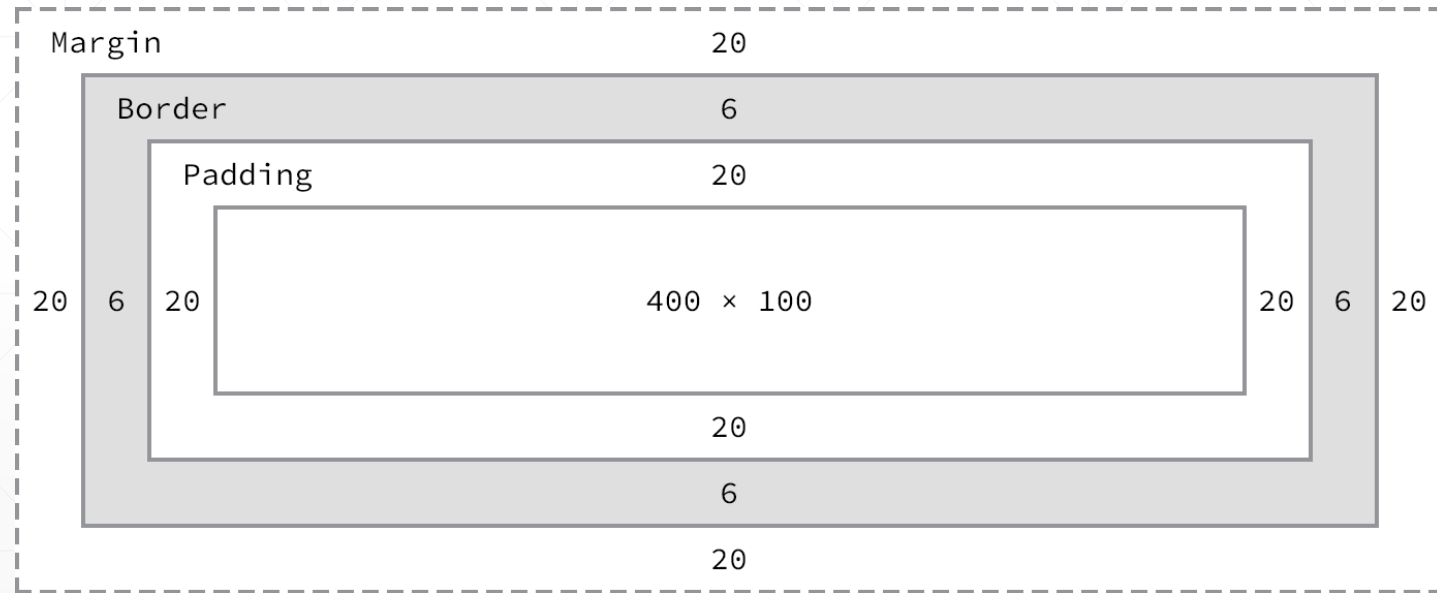


- Blocks begin in new line, spans the entire width
- Inlines fall into normal flow, maintains content width (cannot wrap blocks!)
- Inline-block: block-like styling, inline-like positioning
- None: hides the element

The Box Model

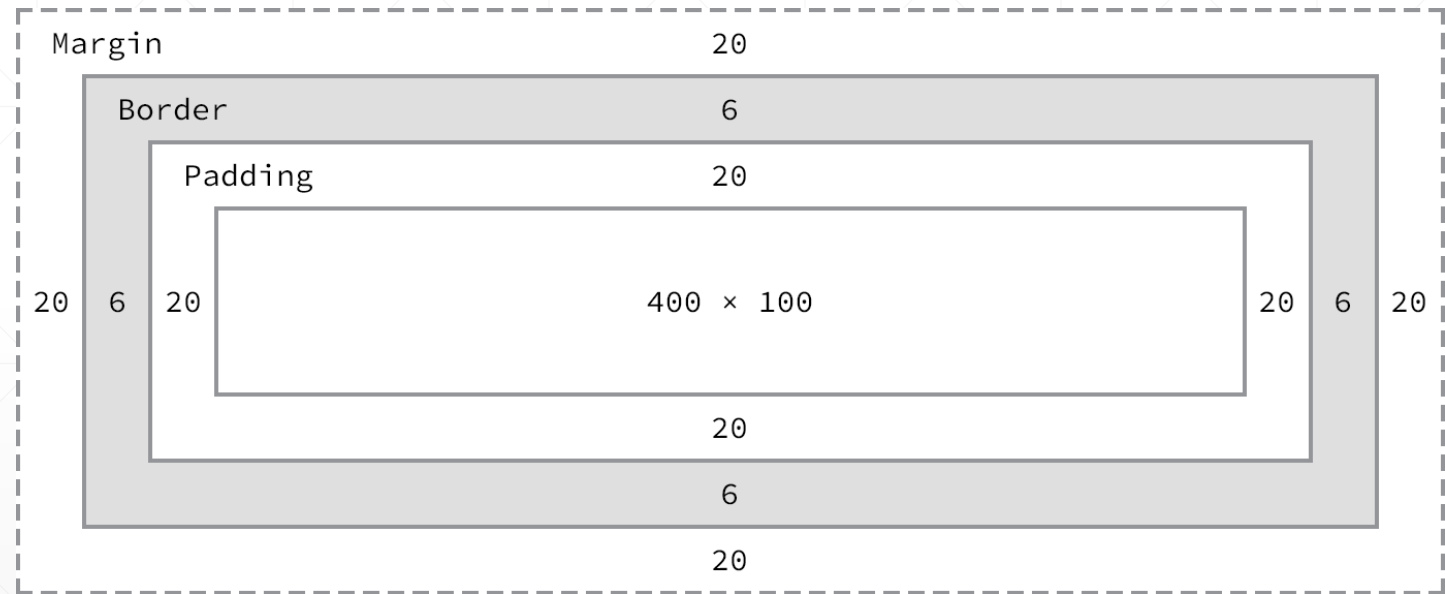
- Every HTML element is regarded as a rectangular box, with:
 - Width & Height
 - Margin
 - Border
 - Padding

Note that the actual size of the element is larger than width * height!



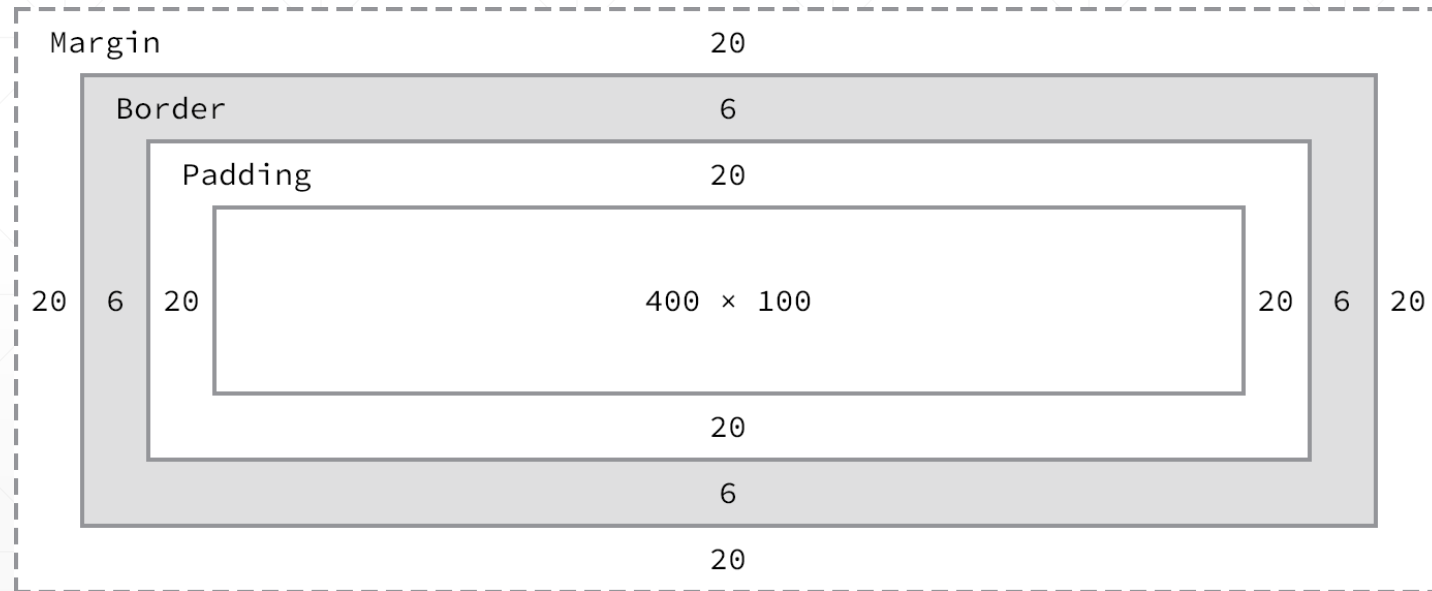
Box Model properties

- Width & Height:
 - Both relative and absolute values can be used.
 - Cannot use for inline
 - Can use for block and inline-block
 - Block's default width: 100%
 - Inline-block's default: content size



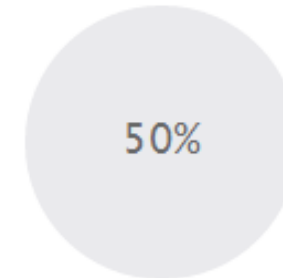
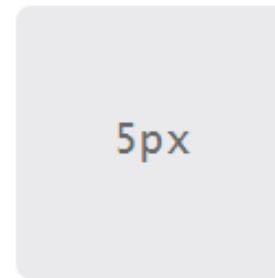
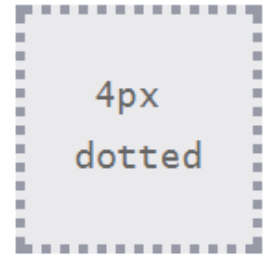
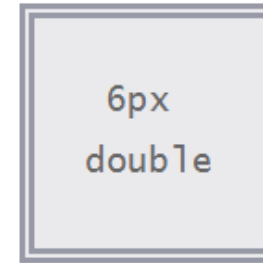
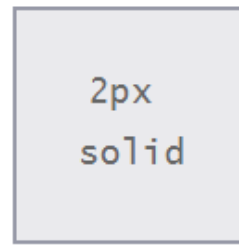
Box Model properties

- Margins & Paddings
 - Spacing between elements
 - For inline element:
 - Margin only works horizontally
 - Padding works for all direction, but vertical padding overlaps with lines above & below
- Ways to declare:
 - Margin: 10px;
 - Margin: 10px 20px;
 - Margin: 10px 20px 15px 30px;
 - margin-top, margin-right, margin-bottom, margin-left



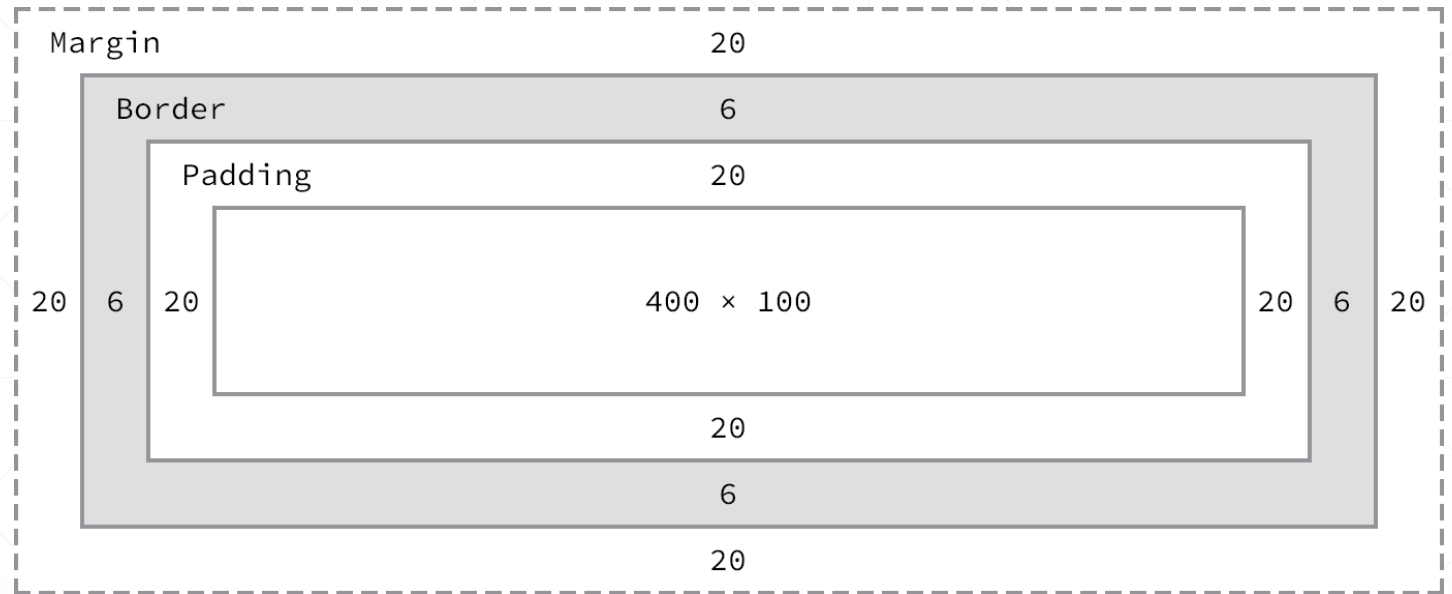
Box Model properties

- Borders:
 - Could specify width, style, color
 - Ways to declare
 - Shorthand vs. longhand
 - Border: 2px dashed yellow;
 - Border-bottom: 1px double black;
 - Border-left-style: dotted;
 - Border-radius is declared separately
 - Border-radius: 50%
 - Border-radius: 15px 50px
 - Border-top-right-radius: 1em;



Box Sizing

- There are three ways of specifying the size of the box with CSS width/height property:
 - Content-box (default)
 - Padding-box
 - Border-box
- Often used in CSS Resets for browser compatibility



```
div {  
  -webkit-box-sizing: content-box;  
  -moz-box-sizing: content-box;  
  box-sizing: content-box;  
}
```

Try on your own!

- (Follow the instructions on <http://learn.shayhowe.com/html-css/opening-the-box-model/#practice-1>)
 - Step 0: First, comment out what you have written last week on your .html and .css (or you can download source code from learn.shayhowe.com/practice/getting-to-know-html.zip)
 - Step 1: Adjust box-sizing
 - Step 2: Create a container class using box model properties
 - Step 3: Apply container class to header and footer
 - Step 4: Add sections with container class
 - Step 5: Create spacing between text-based elements
 - Step 6: Create button classes & styling
 - Step 7: Create a hero class
-

Chrome DevTools

- Easy way to inspect each HTML element and its styling



Typography – font properties

- color: #555
- font-family: “Lato”, “Helvetica Neue”, Helvetica, Arial, sans-serif
 - Downloaded fonts or fonts with spaces must be wrapped with “ “
- font-size: 14px
- font-style: normal/italic/oblique/inherit
- font-variant: normal/small-caps/inherit
- font-weight: normal(400), bold(700), bolder, lighter, inherit
 - Or specify with values 100 ~ 900 (available to a specific font)
- line-height: 22px

shorthand: font-style, font-variant, font-weight, font-size, line-height, font-family

font: italic small-caps bold 14px/22px “Lato”, “Helvetica Neue”, Helvetica, Arial, sans-serif

Let's try

I Am a Builder

Posted by Shay Howe

Every day I see designers and developers working alongside one another. They work intelligently in pursuit of business objectives. They work diligently making exceptional products. They solve real problems and take pride in their work. They are builders.

[Continue...](#)

Typography – text properties

- Text-align: left, right, center, justify, inherit
 - Text-decoration: none, underline, overline, line-through, inherit
 - Text-indent: 20px, -10px, etc.
 - Text-shadow: 3px 6px 2px rgba(0, 0, 0, .3)
 - Text-transform: none, capitalize, uppercase, lowercase, inherit
 - Letter-spacing: -.5em
 - Word-spacing: .25em
-

Let's try

I Am a Builder

Posted by Shay Howe

Every day I see designers and developers working alongside one another. They work intelligently in pursuit of business objectives. They work diligently making exceptional products. They solve real problems and take pride in their work. They are builders.

[CONTINUE...](#)

Thanks!

You can review the materials at <http://learn.shayhowe.com/html-css>