



# WELCOME TO FEWD

WEEK 2 / LESSON 3

Leonie Dunnett

**LESSON 3:**

# **CSS BOX MODEL**

# LEARNING OBJECTIVES

- Apply inline and block attributes to a page
- Define and manipulate the CSS Box Model
- Understand and use CSS resets
- Select nested elements to apply styling

# TODAY'S SCHEDULE

- Review
- Block vs. Inline Elements
- Introducing the Box Model
- Using CSS Resets
- Cross-Browser Compatibility & Testing
- Tags and Boxes **Code Along**
- Selectors and the DOM **Code Along**
- Relaxr Landing Page **Lab**

# REVIEW

- Images
- CSS Basics
- Assignment questions

# **BLOCK VS. INLINE ELEMENTS**

# BLOCK-LEVEL ELEMENTS

eg. `<p>`, `<div>`

- Begin on a new line
- Stack
- Occupy available width
- Can be nested
- Can wrap inline-level elements

# INLINE-LEVEL ELEMENTS

eg. `<span>`

- Don't begin on a new line
- Fall into normal document flow
- Only maintain the contents width
- Will not accept width and height properties
- Can also be nested
- Can't wrap block-level elements

(`<a>` are an exception to this)



# DISPLAY

Most common values:

block

inline

inline-block

none

`display: inline-block;`

Element behaves as a block-level element,  
but will not begin on a new line by default.

There is often space between `inline-block` elements,  
but there are ways to remove this.

# **INTRODUCING THE BOX MODEL**

# EVERY ELEMENT IS A RECTANGULAR BOX

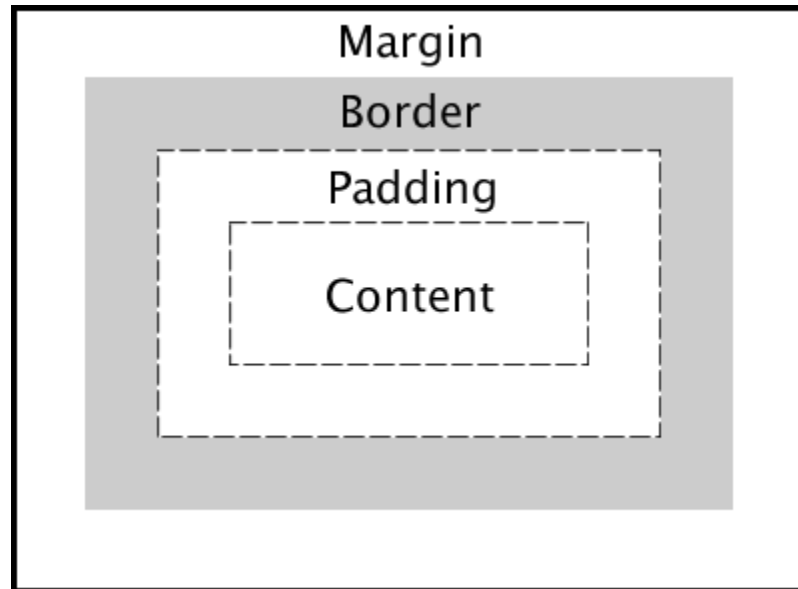
Element 'boxes' can have:

margin border padding width height

...and of course,

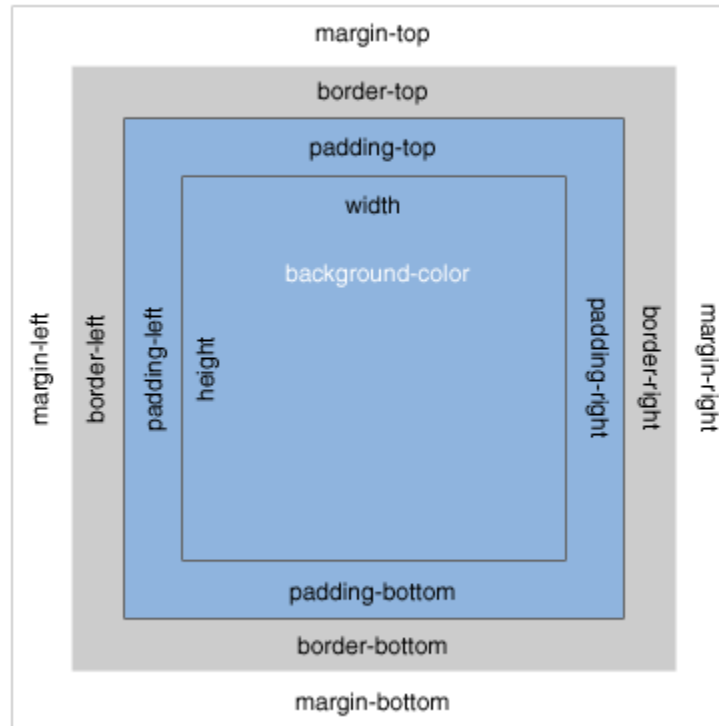
content

# THE BOX MODEL



Describes the physical properties of an elements box.

# THE BOX MODEL



Total width and height is **additive**.

## DEFAULT BOX SIZING

total width = width + padding-left + padding-right  
+ border-left + border-right

total height = height + padding-top + padding-bottom  
+ border-top + border-bottom

Need to add the width, padding, and border values together to get an elements actual full width.

# BOX SIZING PROPERTY - NEW IN CSS3

Now two ways to determine how box-size is calculated:

`box-sizing: content-box;` (default)

Padding and border **added to** total width and height

---

`box-sizing: border-box;` (new way)

Border and padding **included within** width and height



# MARGIN

- Space around outside of box
- Completely transparent in color
- Doesn't work vertically on inline elements
- Not factored into total width

# PADDING

- Space between content and border
- Works vertically on inline elements

# FOUR FORMATS

1. `margin: 10px;` (all sides the same)
2. `margin: 10px 5px;` (top & bottom, left & right)
3. `margin: 20px 5px 10px;` (top, left & right, bottom)
4. `margin: 5px 5px 10px 0;` (top, right bottom, left)

`padding` formats are the same

# BORDERS

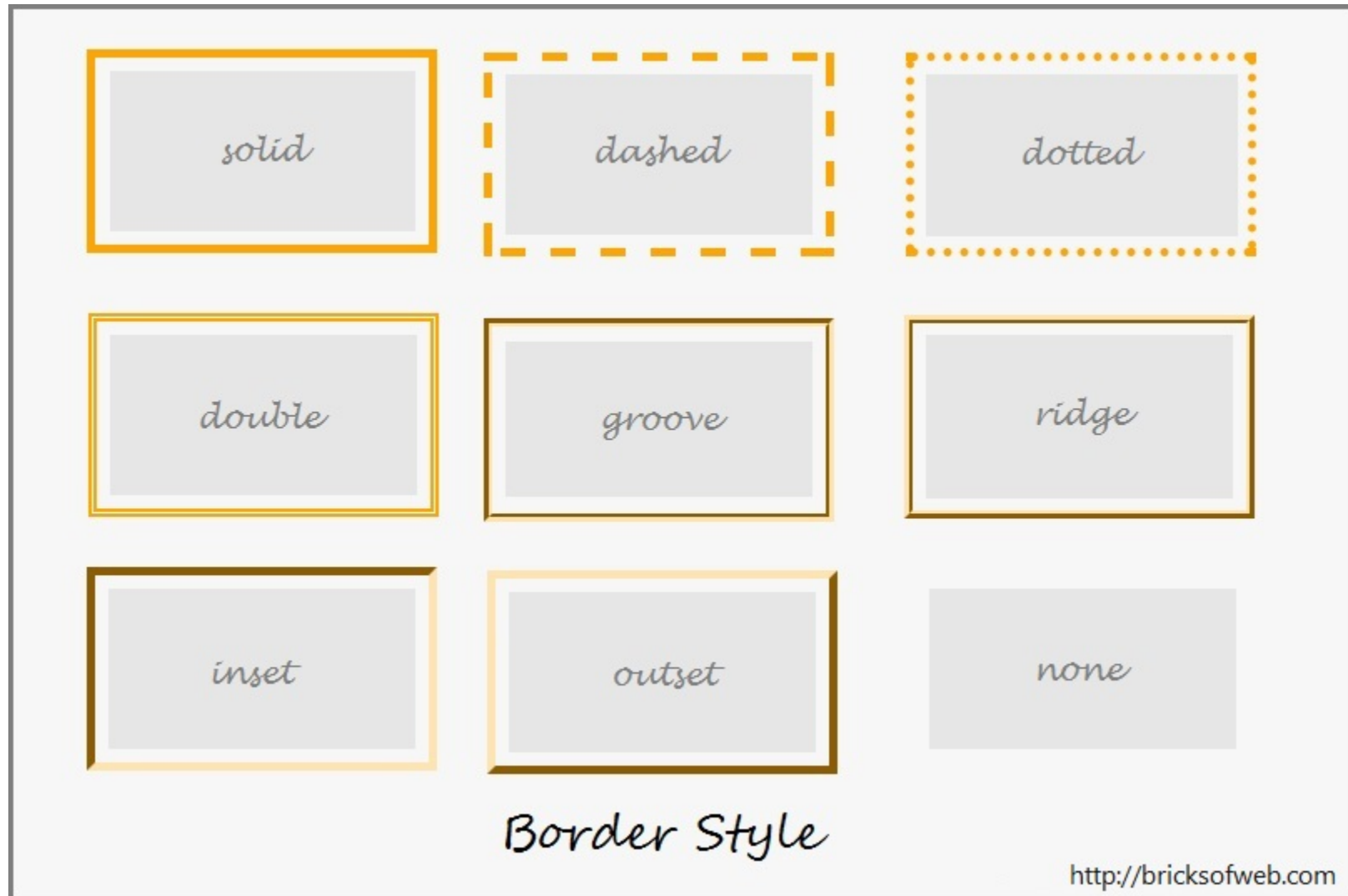
- Borders can have a **width**, **style** and **color**.
- The border shorthand is:

**border: 2px red solid;**

- You can also specify individual properties

eg. **border-width-right: 3px;**

# BORDER STYLES



# BORDER RADIUS

Enables us to round the corners of an element.

Values start at top left of element and go clockwise:

- **Single value** rounds **all four** corners
- **Two values** rounds the **top-left/bottom-right** and **top-right/bottom-left** corners
- **Four values** round the **top-left, top-right, bottom-right,** and **bottom-left** corners

# BORDER DEMOS

Borders on CODEPEN

Border Radius on CODEPEN

# USING CSS RESETS



# CSS RESETS

Browsers apply their own default styles to page elements.  
To standardise these, we need to 'reset' them to the same  
base styles

This is accomplished by applying a CSS Reset.

# CSS RESETS

Generally involve removing any:

sizing, margins, paddings, or additional styles

Its important the the reset style sheet is the first sheet the browser encounters, so that they are working from a common baseline

See: [What is a CSS Reset](#)

# POPULAR RESETS

[Eric Meyer's reset](#)

---

[Normalize.css](#)

(for the more adventurous)

- Set's common styles, rather than removing them
  - Requires a greater understanding of CSS
- 

See also: [2015's Most Popular CSS Resets](#)

# **CROSS-BROWSER COMPATIBILITY & TESTING**

# **BROWSERS RENDER ELEMENTS DIFFERENTLY**

Therefore we must test cross-browser compatibility.

Websites don't need to look exactly the same in every browser, but they should be similar.

Progressive Enhancement vs. Graceful degradation

# BROWSER SUPPORT

Is the **box-sizing** property supported across all browsers?

Let's see:

Can I Use

HTML5 Please

Should I Prefix

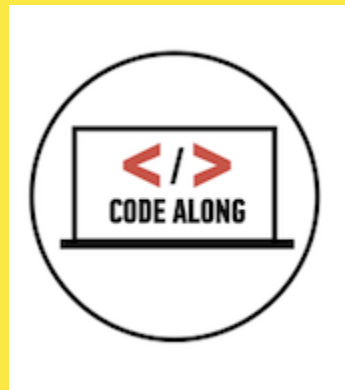
# CSS VENDOR PREFIXES

Outdated example of Box Model Hack:

```
* {  
  -moz-box-sizing: border-box; /* Firefox */  
  -webkit-box-sizing: border-box; /* Safari/Chrome/iOS/Android */  
  box-sizing: border-box; /* IE */  
}
```

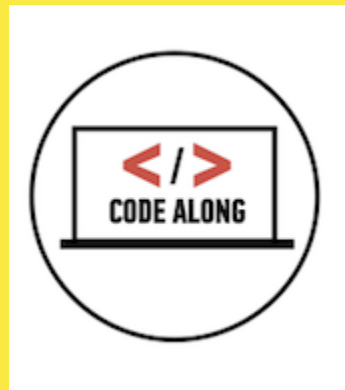
Box Sizing Best Practice:

```
html {  
  box-sizing: border-box;  
}  
*, *:before, *:after {  
  box-sizing: inherit;  
}
```



# TAGS & BOXES





# NESTED SELECTORS

# Nested Selectors in CSS

Often, we may find ourselves wanting to style things *differently* depending on their *context*.

Take emphasis as an example. In typical copy, presented in a roman (non-italic) type, we may wish to add *emphasis* to a word or phrase by using *italics*.

But what if we want to quote a block of text which also emphasizes something?

This is a quote from some famous person who undoubtedly has something *very important* to say.

If we want our block quote to be set in italics by default, then the visual style for emphasis needs to be different, *but only when inside a blockquote*.

This is where *nested selectors* come in handy.



# RELAXR LANDING PAGE

# HOW TO START?

- Draw a DOM (Group exercise)
- Add basic HTML markup
- Add page text
- Begin adding body HTML markup to the page
- Style bit by bit (text, backgrounds, colors, etc)

# REVIEW: LEARNING OBJECTIVES

- Apply inline and block attributes to a page
- Define and manipulate the CSS Box Model
- Understand and use CSS resets
- Select nested elements to apply styling

# **HOMEWORK**

**Review:** Slides and resources

**Code:** Relaxr Landing Page

# RESOURCES

[Box Model explanation](#) via Mozilla

[Collapsing Margins](#) via Mozilla

[Codepen](#) - Testing your own code, looking at cool things other people have done.

[Stack Overflow \(Tour\)](#) - great place to find answers

[HTML Validator](#) - work through html coding errors

[CSS Validator](#) - work through css coding errors