

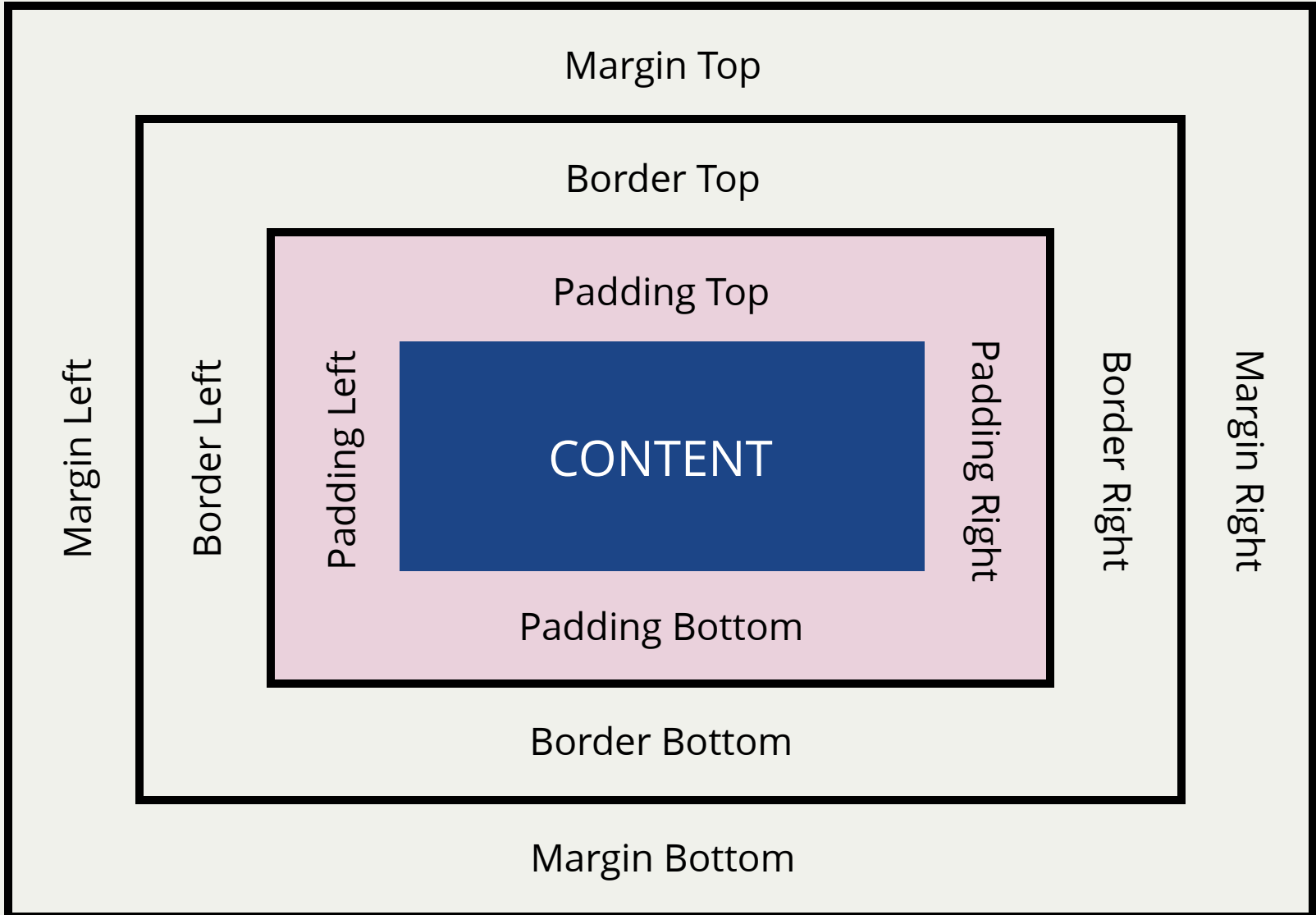


- **Draw** the DOM tree for web pages containing elements like header and footer
- **Experiment** with Display Properties
- **Apply** header and footer, single column and multi column layouts to build web pages
- **Experiment** with, and **predict**, floats and clearing

- Review
- HTML5 Structural Elements
- Display Properties
- Floats and Clears
- Lab Time: Layout Challenge



- What is the Document Flow?
- What do we mean by box model?
- What makes up the box model?
- What do we use padding and margin for?
- When we specify a width, what do we typically specify?
- How do we solve that problem?



**The pink is the element's background colour*



```
<header></header>
```

Used to contain a group or introductory and navigational aids

Heading elements, logos, search form etc.


```
<nav></nav>
```

Used for a section on the page with navigation links

You can have more than one of these on the page

```
<footer></footer>
```

A footer typically contains information about the company or author, copyright data or links to other documents

```
<section></section>
```

Defines a unique section in the document

e.g. A chapter, content blocks etc.

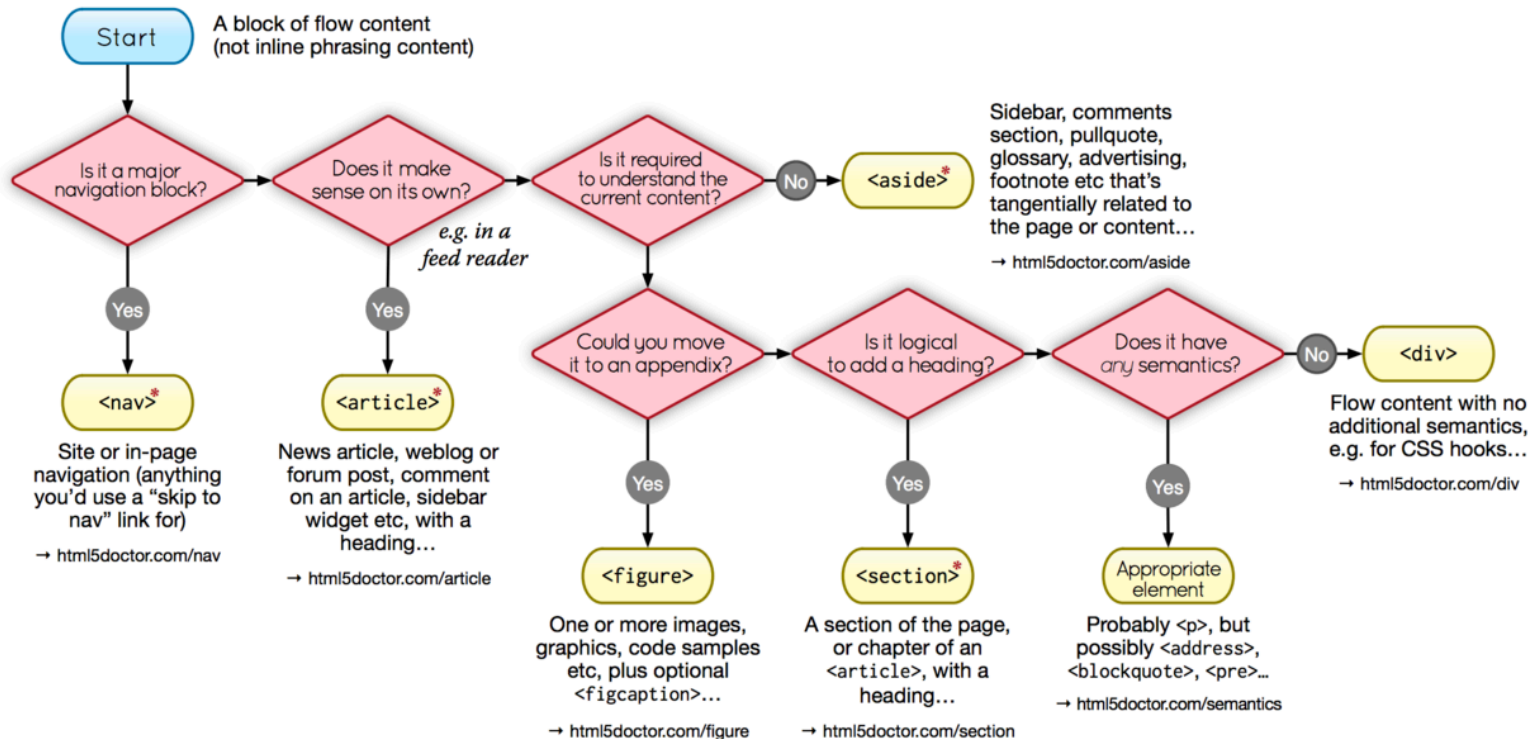


Doctor

HTML5 Element Flowchart

Sectioning content elements and friends

By @riddle & @boblet
www.html5doctor.com

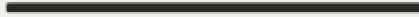


* Sectioning content element

These four elements (and their headings) are used by HTML5's outlining algorithm to make the document's outline
→ html5doctor.com/outline

2011-07-22 v1.5
For more information:
www.html5doctor.com/semantics





Display determines the type of rendering box a particular element uses

- Can other elements sit next to it?
- Can you set widths and heights?
- Does padding and margin work?

Elements have these set by default (the **user agent stylesheet** does this)

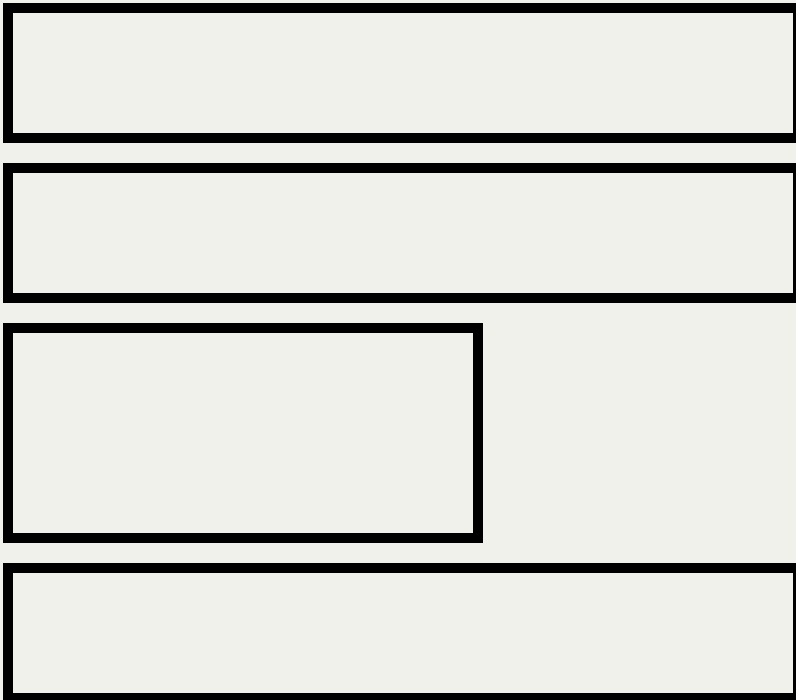
-
- Start on a new line
 - By default, they will stretch out to the right and left as far as they can (unless you set widths or heights)
 - Nothing can sit on the same line as it
 - Widths, heights, margin and padding all work

-
- Does not disrupt the flow of other elements
 - It will sit next to any element it can
 - Padding works (but does not push elements away vertically)
 - Margin works horizontally
 - Widths and heights don't work at all

-
- A mix of inline and block
 - Other elements can sit in the same line
 - You can set widths and heights
 - Margins and padding works perfectly

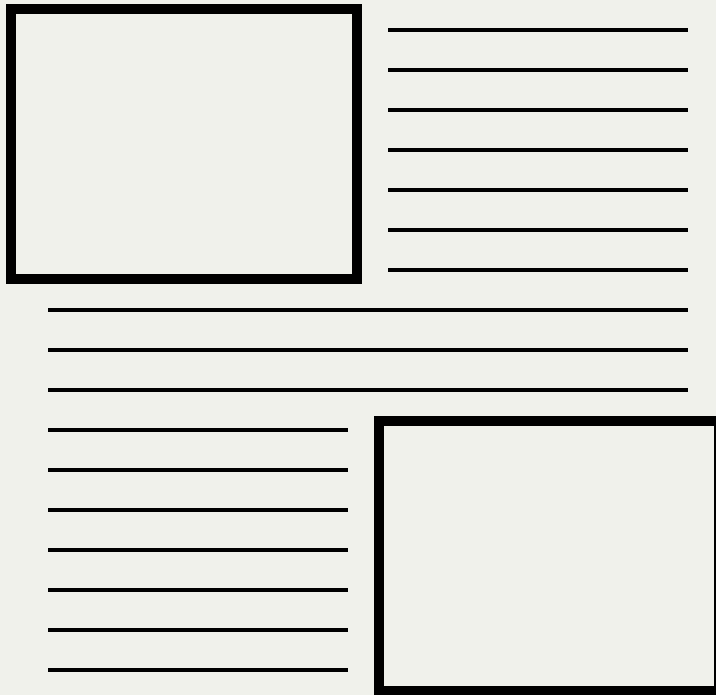
-
- Completely removed from the document flow
 - Invisible
 - It's like the element was never there





- How elements sit on a page
- The natural flow of a webpage

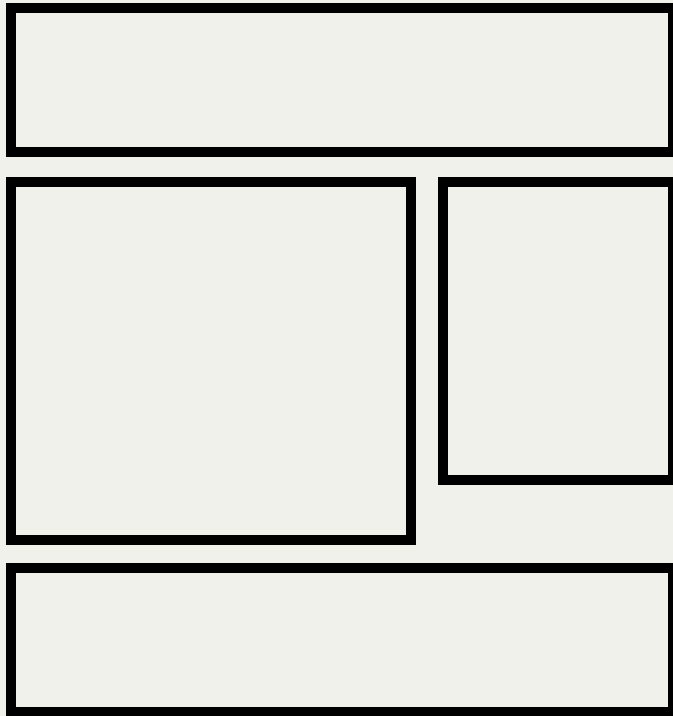
But what if we want to change that?



Elements with the float property:

- Remain in the document flow
- They can be pushed to the left or right
- Other content can flow around them

```
element {  
  float: left | right | none;  
}
```

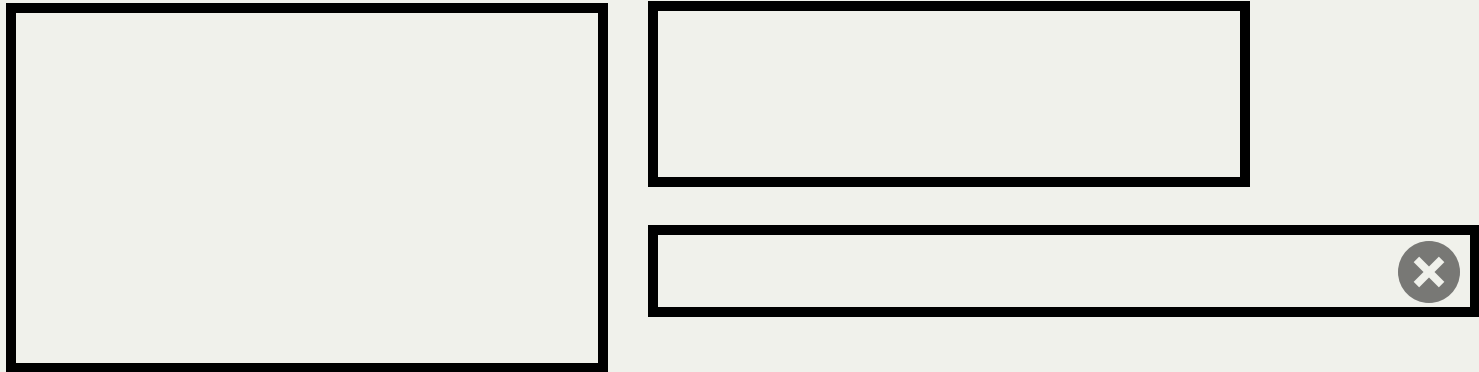


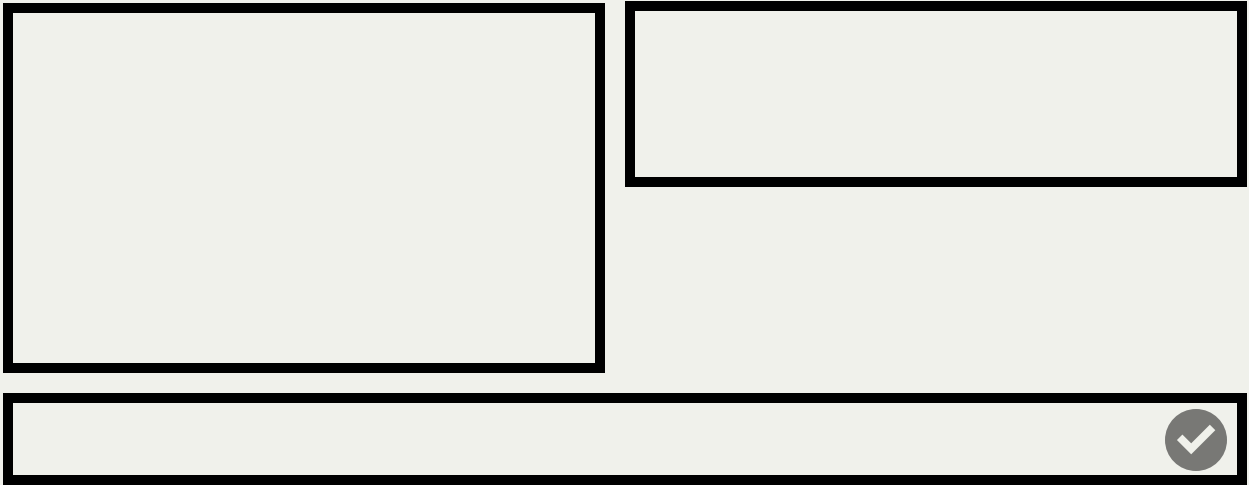
Beyond just wrapping text:

- Building full page layouts
- Smaller parts of the page
 - Navigations
 - etc.



Once you have floated something, everything after it in the document flow will float up and try and fill whatever gap there is

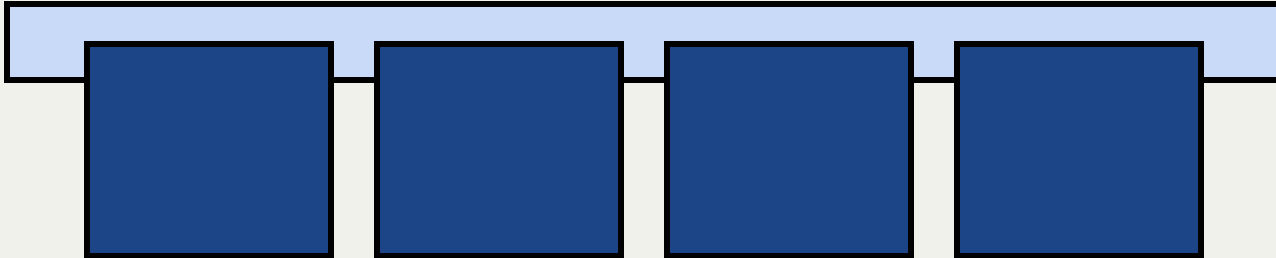




Clearing is drawing a line and saying things can't float past this point

```
element {  
  clear: left | right | both | none;  
}
```

If everything inside an element is floated, that element will have a height of zero



Unless you clear it! ([Solutions here](#))

So how do we clear a float?

-
1. **The Empty Div Method**
 2. **The Overflow Method**
 3. **The Clearfix Method**

Adding an empty element to your HTML that has **clear both** attached to it. Add it wherever you need to stop the floats

```
...  
<div class="clear"></div>  
...
```

```
...  
.clear {  
    clear: both;  
}  
...
```

Add **overflow auto** or **overflow hidden** to the parent element that contains all of the floats. This will force the element to care about floated elements

Use the clever **:after** CSS pseudo-selector to clear the floats. This pseudo-selector adds content to the page, and you can style that content to have **clear both**

```
.clearfix:after {  
    content: "";  
    visibility: hidden;  
    display: block;  
    height: 0;  
    clear: both;  
}
```

Layout Challenges

- Finish Relaxr
- Muck around with those Layout Challenges
- Review floats, clears and display
 - CSS Tricks
 - Smashing Magazine
 - Floats and Escalators (this is really well done!)



Lesson 4: CSS Floats and Clears

<https://ga.co/fewd32syd>

