

# Advanced CSS

# Our Goals

- Working effectively with your editor
- Understand CSS positioning and display
- Writing semantic HTML
- Have a brief introduction to Web Accessibility

# Atom is already pretty good...

But we can use packages to make it better:

- Emmet
- Open in browser
- Open recent
- Atom Beautify
- Minimap
- JS Hint
- Todo show
- JavaScript snippets
- Autodetect indentation
- Linter
- Linter - HTML
- Linter - CSS
- Linter - JS
- Color Picker

# What is Zen coding (Emmet)?

- It automates the creation of HTML for us
- It came from [here](#), it used to be called Zen coding

# What is Zen coding (Emmet)?

- It automates the creation of HTML for us
- It came from [here](#), it used to be called Zen coding
- Most of us are sick of writing HTML, too many angle brackets and too many quotes

# What is Zen coding (Emmet)?

- It automates the creation of HTML for us
- It came from [here](#), it used to be called Zen coding
- Most of us are sick of writing HTML, too many angle brackets and too many quotes
- It's an abbreviation expander, to use it:
  - You type in a set of keywords and symbols
  - Then press tab or **<CTRL> + E**

# What is Emmet?

- It was built by [Sergey Chikuyonok](#)
  - For [Smashing Magazine](#)
- It was Zen Coding, but now everyone knows it as Emmet

# Doctypes and Full Pages

```
!  
html:5
```



# Creating Elements

```
div
```

```
p
```

```
a
```

```
h1
```

```
div.className
```

```
div#idName
```

```
div.className#idName
```

# Adding extra information

[ ] - for attributes

{ } - for text

```
h1{Hello World}  
a[href="http://ga.co"]  
a[href="http://ga.co"]{General Assembly}
```

# Creating Siblings

+ - for siblings

```
h1+a  
header+main+footer
```

# Creating Children

> - for children

```
header>h1  
div>p>lorem
```

# Creating Elements

\* - for children

```
p*3  
ul>li*3
```

# Creating Groups

() - for groups

```
(header>h1)+(main>p)  
(main>p*3)+(footer>ul>li*3)
```

# Use it as much as possible!

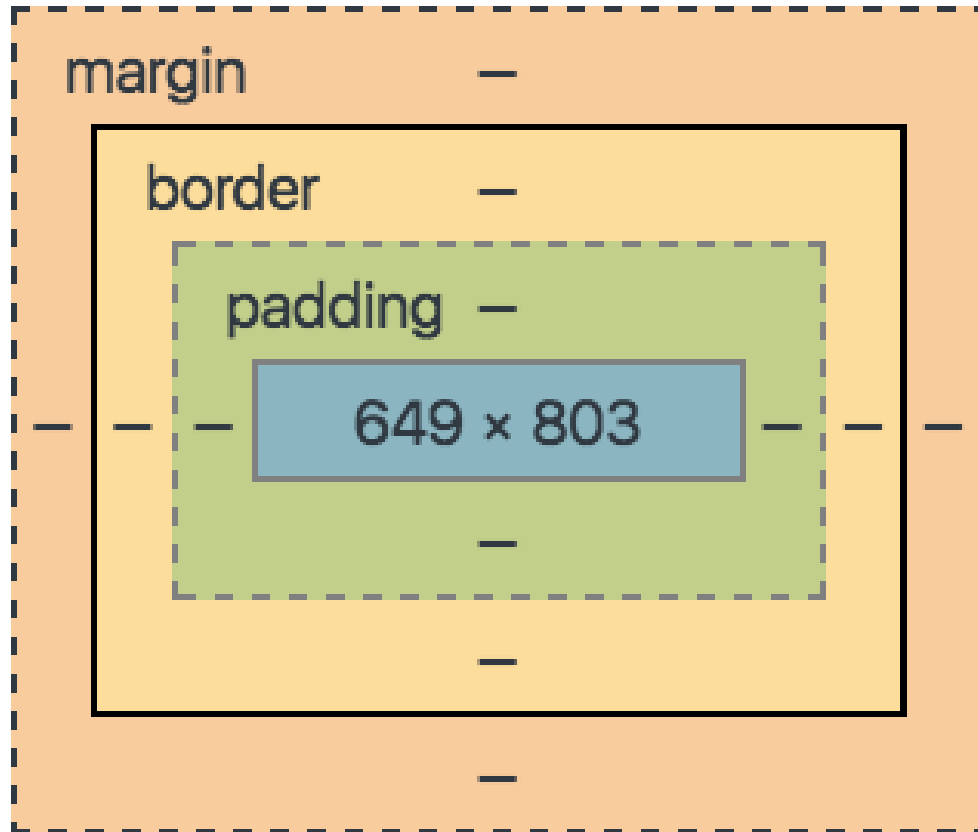
- [Emmet Documentation](#)
- [Emmet Cheat Sheet](#)

# Atom Shortcuts

- [Atom Editor Cheat Sheet](#)
- [Atom Keyboard Shortcuts](#)



# Box Model



# Display

The ***Display*** property specifies the way that an element interacts with other elements:

- Whether they can have other elements sitting next to them
- Whether you can set heights or widths

# Display

- ***inline***

- The element can have other elements sitting next to it, but you can't set widths and heights

- ***block***

- The element can't have other elements sitting next to it, and you can set widths and heights

# Display

- ***inline-block***

- The element can have other elements sitting next to it, and you can set widths and heights

- ***none***

- The element is invisible and doesn't take up any space on the page

# Position

The ***Position*** property specifies exactly what you are probably imagining. You can move elements based on:

- Where the element is meant to be

# Position

The ***Position*** property specifies exactly what you are probably imagining. You can move elements based on:

- Where the element is meant to be
- Where the element is in the *entire document*

# Position

The ***Position*** property specifies exactly what you are probably imagining. You can move elements based on:

- Where the element is meant to be
- Where the element is in the *entire document*
- Where the element is in the *browser window*

# Position

The ***Position*** property specifies exactly what you are probably imagining. You can move elements based on:

- Where the element is meant to be
- Where the element is in the *entire document*
- Where the element is in the *browser window*
- *It can remove the elements from the document flow*



# Position: Relative

Based on where the element is meant to be in the document flow. It's good for:

- Making elements overlap

# Position: Absolute

You can specify the position of an element based on the entire document, but it won't scroll with the page. Good for:

- Precise positioning
- Difficult alignment

# Position: Fixed

You can specify the position of an element based on the browser window, and it will scroll with the page. Good for:

- Creating headers
- Precise positioning
- Difficult alignment

# Variadic Attributes

Shorthand to apply a number of properties

# Variadic Attributes

Shorthand to apply a number of properties

```
h1 {  
  /* Applies to all four sides */  
  margin: 1em;  
  
  /* vertical | horizontal */  
  margin: 5% auto;  
  
  /* top | horizontal | bottom */  
  margin: 1em auto 2em;  
  
  /* top | right | bottom | left */  
  margin: 2px 1em 0 auto;  
}
```

# Custom Fonts

- [Google Fonts](#)
- [Font Awesome](#)
- Custom Fonts
- [Fontello](#)
- [Icomoon](#)

# Google Fonts

- Go through [here](#) and Add the fonts that you want to your Collection
- Once you have selected all your fonts, click Use (bottom right)
- Choose the styles that you would like, and the character set
- Choose @import, and copy and paste the code into the top of your CSS file that it shows
- Reference the font with the code provided

# Font Awesome

- Go [here](#)
- Put this in the head of your HTML page
  - `<link rel="stylesheet"  
href="https://maxcdn.bootstrapcdn.com/font-  
awesome/4.6.1/css/font-awesome.min.css">`
- Go through [here](#) and click on the icons that you want
- That will show you the HTML that you need



# Custom Fonts

Have the files in your folder and reference them

- You can use [this tool](#) to get all the different types

```
@font-face {  
  font-family: 'GT Pressura';  
  src: url('fonts/GTPresurra.eot');  
  src: local('GT Pressura'),  
        url('fonts/GTPressura.eot#iefix'),  
        url('fonts/GTPressura.eot') format('truetype'),  
        url('fonts/GTPressura.otf') format('opentype'),  
        url('fonts/GTPressura.woff') format('woff'),  
        url('fonts/GTPressura.woff2') format('woff2'),  
        url('fonts/GTPressura.svg') format('svg');  
}
```

# Writing Semantic HTML

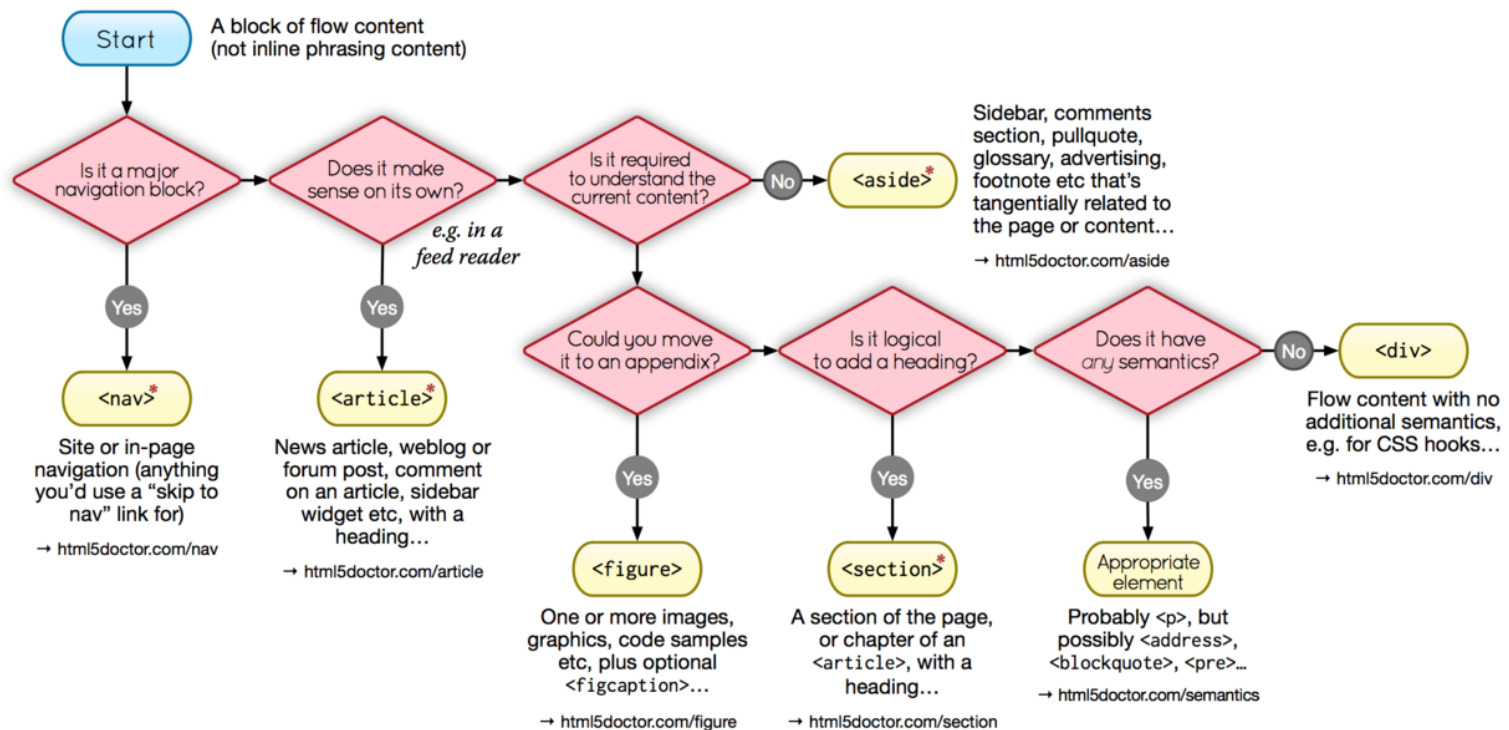
- Giving meaning to a subject
- It aids how both humans and machines interpret our page
- Great for "SEO"
- It's very hard to get used to
- Not always seen as a positive use of time



# HTML5 Element Flowchart

## Sectioning content elements and friends

By @riddle & @boblet  
[www.html5doctor.com](http://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](http://html5doctor.com/outline)

2011-07-22 v1.5  
For more information:  
[www.html5doctor.com/semantics](http://www.html5doctor.com/semantics)

# Important Links

- [Let's talk about semantics](#)
- [Our pointless point of semantic value](#)
- [Pursuing semantic value](#)

# Web Accessibility

# Web Accessibility

- Semantic

# Web Accessibility

- Semantic
- Perceivable

# Web Accessibility

- Semantic
- Perceivable
- Operable



# Web Accessibility

- Semantic
- Perceivable
- Operable
- Understandable

# Web Accessibility

- Semantic
- Perceivable
- Operable
- Understandable
- Robust

# Web Accessibility

- Semantic
- Perceivable
- Operable
- Understandable
- Robust
- Valid - [HTML](#) and [CSS](#)

# Good links

- [The Accessibility Cheatsheet](#)
- [The Web Accessibility Basics](#)
- [ARIA on MDN](#)
- [Web Content Accessibility Guidelines](#)
- [The Web Accessibility Initiative](#)

**Your homework**