Day 2: Terminology Review

Much of this lecture taken from the SelecTutorial, http://css.maxdesign.com.au/selectutorial

Terminology

Selector – shown in red
Declaration block – shown in blue

```
p { line-spacing: 1.5em; }
```

Declaration

```
p { line-spacing: 1.5em; }
```

Property (red) and **value** (blue)

```
p { line-spacing: 1.5em; }
```

Grouping selectors

```
h1, h2 { color: red; }
```

Make all H1 AND H2 red.

```
section h1, h2 { color: red; }
```

Make H1's located in <section> and ALL H2 red.

```
section h1, section h2 { color: red; }
```

Make only the h1's and h2's in <section> red

For easier reading, you may wish to write as:

```
section h1,
section h2 {
    color: red;
}
```

Shorthand (http://css.maxdesign.com.au/selectutorial/rules_shorthand.htm)

```
p {
        padding-left: 4em;
        padding-right: 2em;
        padding-top: 1em;
        padding-bottom: 3em;
}
```

Will generally be written as:

```
p { padding: 1em 2em 3em 4em; }
```

You can use one, two, three and four values within a shorthand declaration. For example, the rule below will apply padding to all sides of a box:

```
p { padding: 1em; }
```

The rule below will apply 1em of padding to the top and bottom, and 2em of padding to the left and right of the box.

```
p { padding: 1em 2em; }
```

The rule below will apply 1em of padding to the top, 2em of padding to the left and right, and 3em to the bottom of the box.

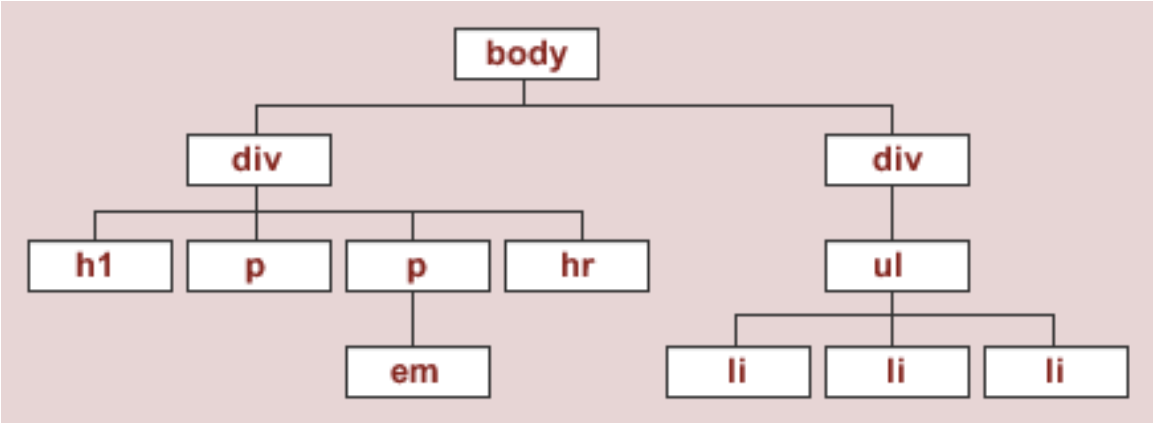
```
p { padding: 1em 2em 3em; }
```

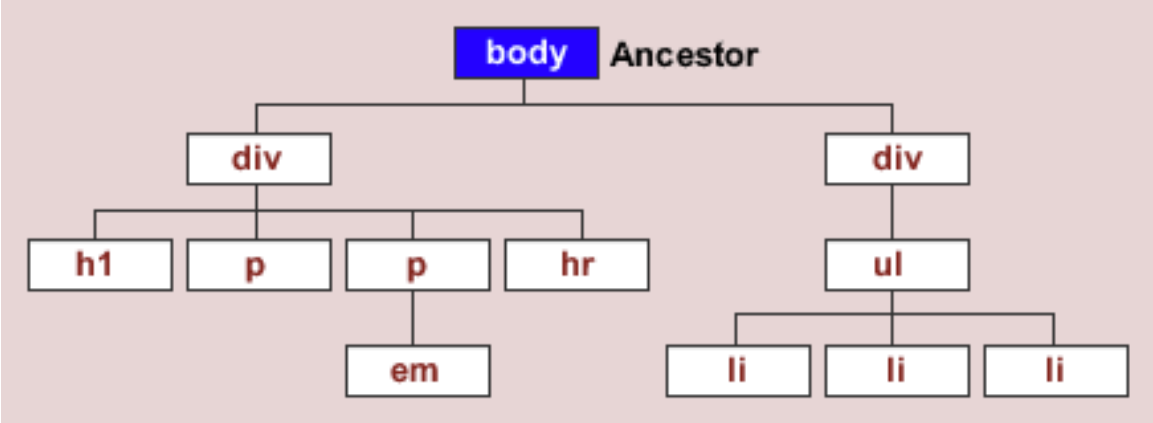
The rule below will apply 1em of padding to the top, 2em of padding to the right, 3em of padding to the bottom and 4em of padding to the left of the box.

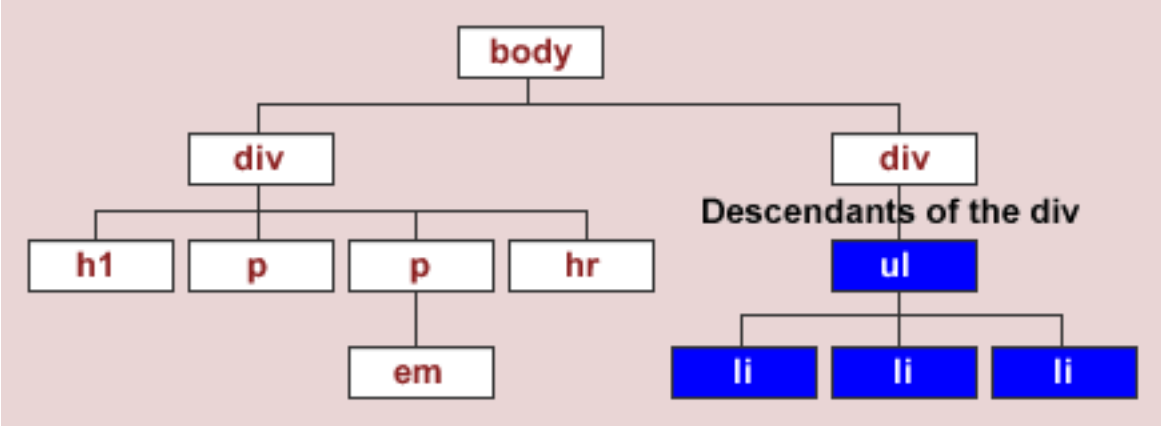
```
p { padding: 1em 2em 3em 4em; }
```

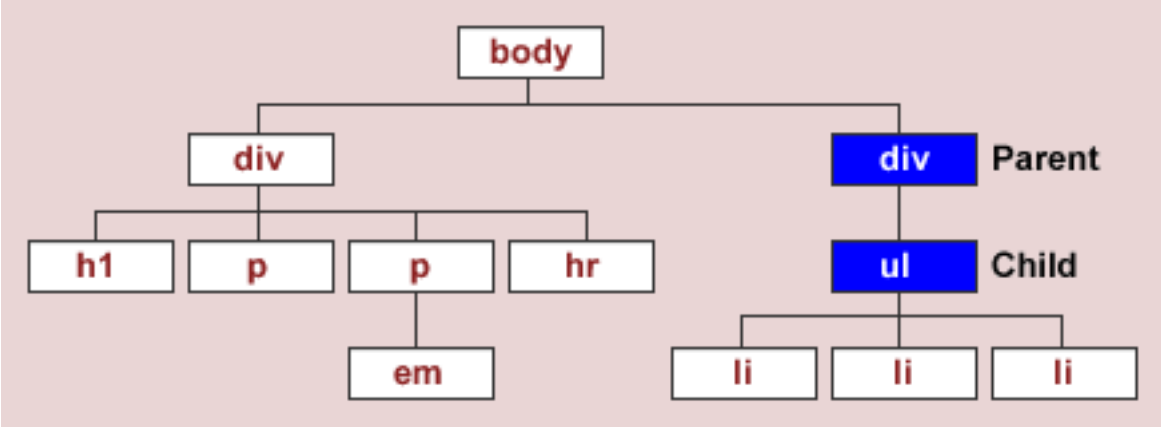
```
<body>
<div id="content">
    <h1>Heading here</h1>
    <p>Lorem ipsum dolor sit amet.</p>
    <p>Lorem ipsum dolor <em>sit</em> amet.</p>
    <hr>
</div>
<div id="nav">
    <ul>
        <li>item 1</li>
        <li>item 2</li>
        <li>item 3</li>
    </ul>
```

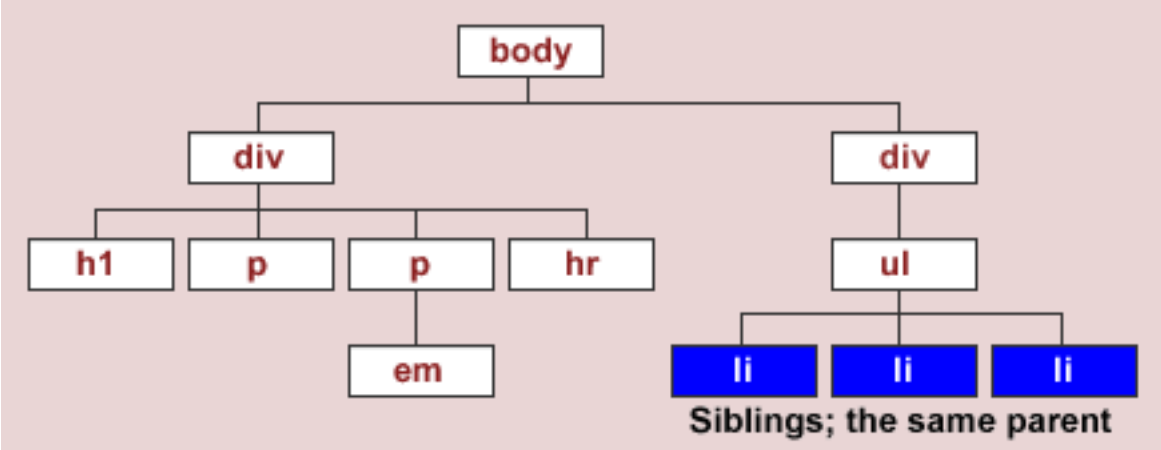
```
</div>
</body>
```

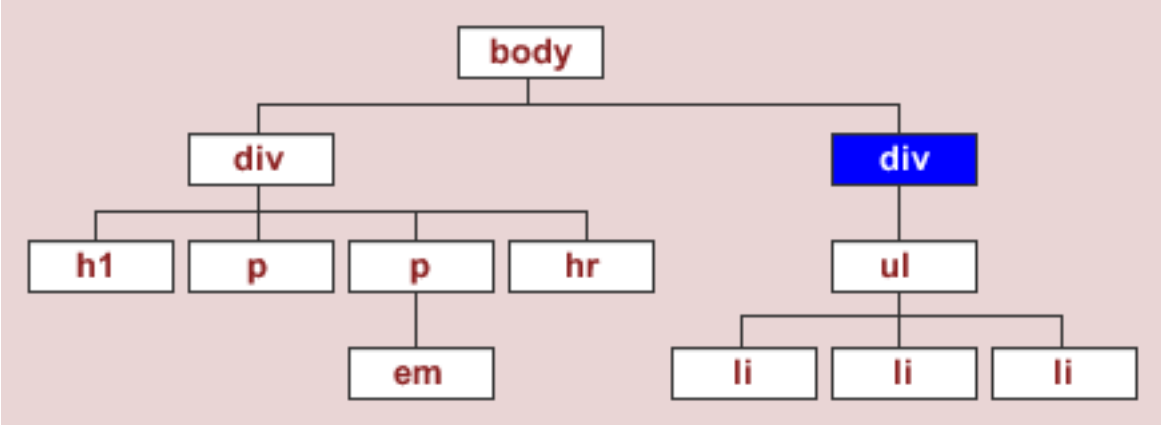

body
div
div
h1
p
p
hr
ul
em
li
li
li


body
Ancestor
div
div
h1
p
p
hr
ul
em
li
li
li


body
div
div
Descendants of the div
h1
p
p
hr
ul
em
li
li
li







- a child of the <body> element
- a parent of the <ul> element
- an ancestor of the <ul> and <li> elements
- a sibling of the other div element (who shares the same parent - the <body> element).

Exercise:
Diagram day2.html as a chart like those shown above. Choose 3 HTML tags within the diagram. For each of these HTML tags, identify any ancestors, descendants, parents, children, and siblings shared by that element.

Selectors

```
<body>
<div id="content">
    <h1>Heading here</h1>
    <p>Lorem ipsum dolor sit amet.</p>
    <p>Lorem ipsum dolor <em>sit</em> amet.</p>
    <hr>
</div>
<div id="nav">
    <ul>
        <li>item 1</li>
        <li>item 2</li>
        <li>item 3</li>
    </ul>
</div>
</body>
```

Name	Example	Browser support notes/CSS version
Type/Element/Tag	`p { color: red; }`	CSS1
Class	`.warning { color: orange; }`	CSS1
ID	`#danger { color: red; }`	CSS1
Descendant	`.warning p { font-weight: bold; }`	CSS1 http://www.sitepoint.com/web-foundations/descendant-selector-css-selector/
Child	`.warning > p { font-weight: bold; }`	CSS2.1. Not supported IE6 and less. http://www.sitepoint.com/web-foundations/child-selector-css-selector/
Universal	`* {`	CSS2.1

	`color: green;` `}`	http://www.sitepoint.com/web-foundations/universal-selector-css-selector/

Adjacent Sibling	`li + li {`
	`color: blue;`
	`}`
		CSS2.1. Not supported IE6 and less.
		http://www.sitepoint.com/web-foundations/adjacent-sibling-selector-css-selector/
		
		Must be immediately next to the sibling.

General Sibling	`li ~ li {`
	`color: blue;`
	`}`
		CSS3
		http://www.sitepoint.com/web-foundations/general-sibling-selector-css-selector/
		
		Must be siblings, but not next to each other necessarily.

```
<ul>
      <li><a href="" title="friend met">Peter</a></li>
      <li><a href="" title="friend">Patty</a></li>
      <li><a href="" title="met contact frilly">Priscilla</a></li>
</ul>
```

Attribute Selectors		Some CSS2.1, most CSS3 No support IE6 and less. Spotty support until IE9. After that, full support. IE7 and IE8 support only these CSS3 selectors: General siblings (`element1~element2`) and Attribute selectors `[attr^=val]`, `[attr$=val]`, and `[attr*=val]` http://caniuse.com/#feat=css-sel3 http://www.sitepoint.com/web-foundations/css3-attribute-selectors/

Simple Attribute	`a[lang] {`
	`margin-left: 10px;`
	`}`
		Affects all a tags in above example

Exact Attribute Value	`a[title='friend'] {`	affects 2nd link only – must be an exact

	color: red; }	match

Partial Attribute Value	a[title~='friend'] {
	color: green;
	}
		affects 1st and 2nd links since both contain friend

Beginning Substring Attribute Value	a[title^='met'] {
	color: green;
	}
		affects 3rd link only – starts with met

Ending Substring Attribute Value	a[title$='met'] {
	color: green;
	}
		affects 1st link only since it ends with met

Arbitrary Substring Attribute Value	a[title*="fri"] {
	color: green;
	}
		affects all 3 links since string "fri" is contained in each

Exercise:

```
<p><a href="http://example.com/folder1/file.pdf">Example1</a></p>
<p><a href="http://example.com/folder2/file.pdf">Example2</a></p>
```

Make the 2nd link bold and red, without changing any HTML.

```
a[href^='http://'][href*='/folder2/'][href$='.pdf'] {
     color: red;
     font-weight: bold;
}
```

Pseudo Classes

http://www.sitepoint.com/web-foundations/pseudo-classes/
http://www.sitepoint.com/web-foundations/css3-pseudo-classes/
http://www.sitepoint.com/web-foundations/understanding-nth-child-pseudo-class-expressions/

From CSS1:

- :link, :visited, :active

From CSS2:

- :hover, :focus
- :lang
- :first-child
- from CSS3, there are a zillion more

Lord Vader, Former Handle Anakin

This isn't the only useful order, nor is it in any way the "right" order. The order in which you specify your pseudo-classes will depend on the effects you want to show with different combinations of states. It's possible, for instance, that you might want to have different hover or focus effects on visited and unvisited links. In that case, you could combine pseudo-classes:`a:link:hover`.

http://www.sitepoint.com/web-foundations/pseudo-classes/

New CSS3 Pseudo Classes

In this Section

1. :nth-child(N)
   matches elements on the basis of their positions within a parent element's list of child elements
2. :nth-last-child(N)
   matches elements on the basis of their positions within a parent element's list of child elements
3. :nth-of-type(N)
   matches elements on the basis of their positions within a parent element's list of child elements of the same type
4. :nth-last-of-type(N)
   matches elements on the basis of their positions within a parent element's list of child elements of the same type
5. Understanding :nth-child Pseudo-class Expressions
6. :last-child
   matches an element that's the last child element of its parent element
7. :first-of-type
   matches the first child element of the specified element type
8. :last-of-type
   matches the last child element of the specified element type
9. :only-child
   matches an element if it's the only child element of its parent
10. :only-of-type
    matches an element that's the only child element of its type
11. :root
    matches the element that's the root element of the document
12. :empty
    matches elements that have no children

13. :target
    matches an element that's the target of a fragment identifier in the document's URI
14. :enabled
    matches user interface elements that are enabled
15. :disabled
    matches user interface elements that are disabled
16. :checked Pseudo-class
    matches elements like checkboxes or radio buttons that are checked
17. :not(S)
    matches elements that aren't matched by the specified selector

Pseudo Elements

https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-elements

Historically, pseudo elements and pseudo classes have had similar syntax. In CSS3, the pseudo elements are separated by a :: rather than :

- `::after`
- `::before`
- `::first-letter`
- `::first-line`

  You can use only one pseudo-element in a selector. It must appear after the simple selectors in the statement.

Browser	Lowest Version	Support of
Internet Explorer	8.0	`:pseudo-element`
	9.0	`:pseudo-element ::pseudo-element`
Firefox (Gecko)	1.0 (1.0)	`:pseudo-element`
	1.0 (1.5)	`:pseudo-element ::pseudo-element`
Opera	4.0	`:pseudo-element`
	7.0	`:pseudo-element ::pseudo-element`
Safari (WebKit)	1.0 (85)	`:pseudo-element ::pseudo-element`

Understanding :nth-child and :nth-of-type

:nth-child(N)
:nth-last-child(N)
:nth-of-type(N)
:nth-last-of-type(N)

N = keyword, number, or number expression

Keywords: odd, even

li:nth-child(odd) {}

N is a number, that's the thing selected (i.e. one thing).

li:nth-child(5) {}

N can be an expression in the form **an + b**.

b = the position where the stuff starts
a = every a-th item will have this effect.

1n+3 = start at the 3rd item and every item after that

2n+3 = start at the 3rd item and make every other item after that

b is not required, so li:nth-child(2n) is legal (and does the same thing as even)

Odd would be 2n-1

If you want to count backwards, this is also possible:
li:nth-child(-n+3) – colors the 3rd li red, and counts backwards

nth-last-child does all of its counting from the last item
li:nth-last-child(3n) – items 2, 5, and 8 are red

nth-of-type is less prone to breaking… nth-child may be more common.

http://css-tricks.com/the-difference-between-nth-child-and-nth-of-type/