

CSS Combinators

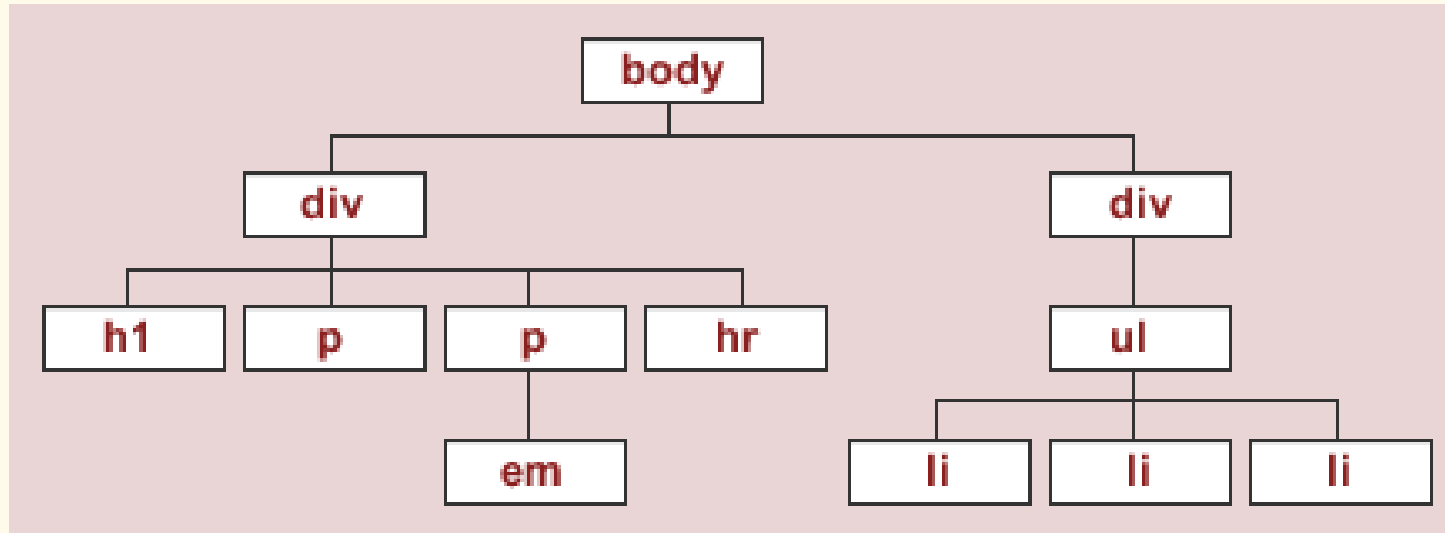
Website Development 1

CSS Combinators

- Combinators allow us to target elements based on their relationship to other elements (for example, we might want to target `` elements but *only* if they're inside a paragraph element)
- There are several options:
 - Descendant selector
 - Child selector
 - Sibling selector
 - Adjacent sibling selector

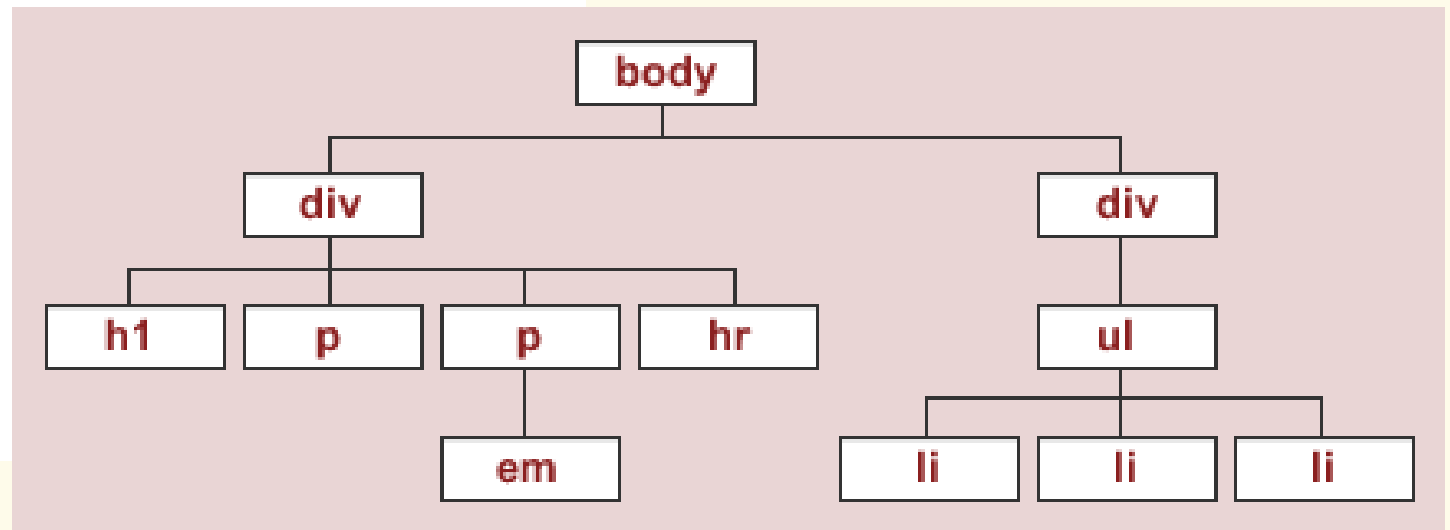
HTML DOM (Document Object Model)

- Before talking about combinators, it's important to understand the HTML DOM – Document Object Model
- Every HTML webpage can be viewed as a tree structure:



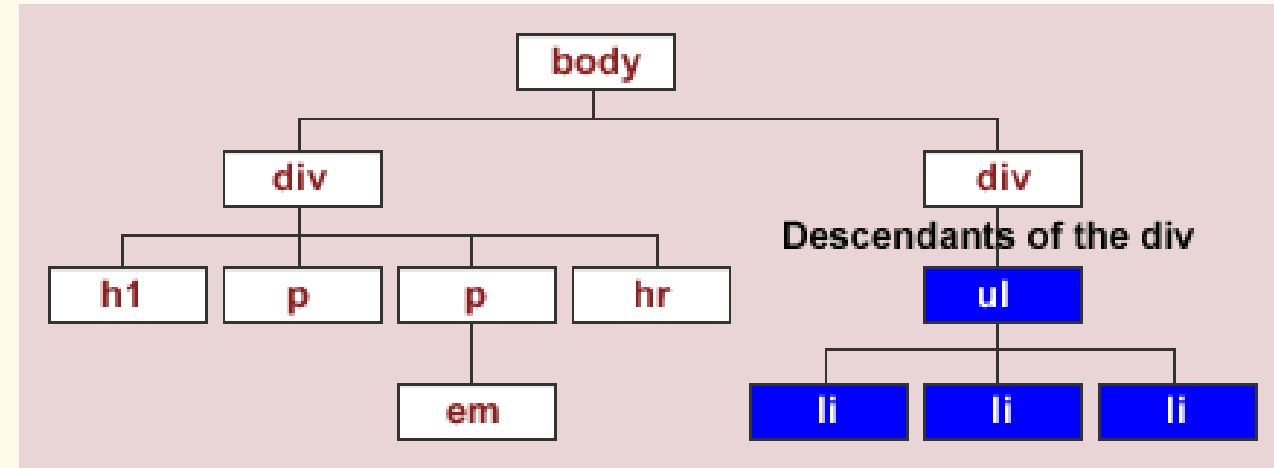
HTML DOM (Document Object Model)

```
<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>
```



Descendants and Ancestors

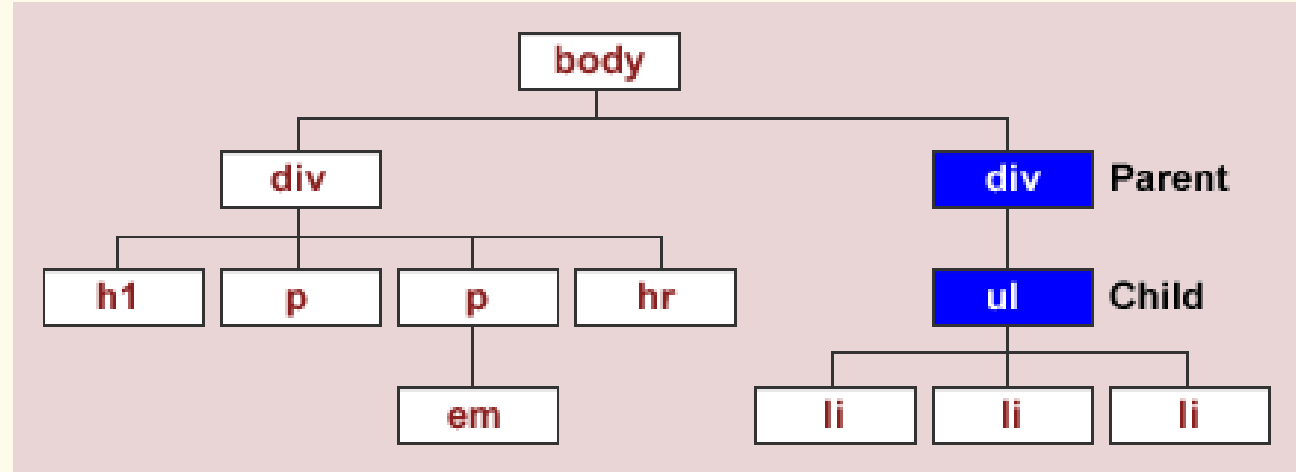
- A **descendant** refers to any element that is connected but lower down the document tree - no matter how many levels lower.



- An element that is connected further up the document tree is referred to as an **ancestor**

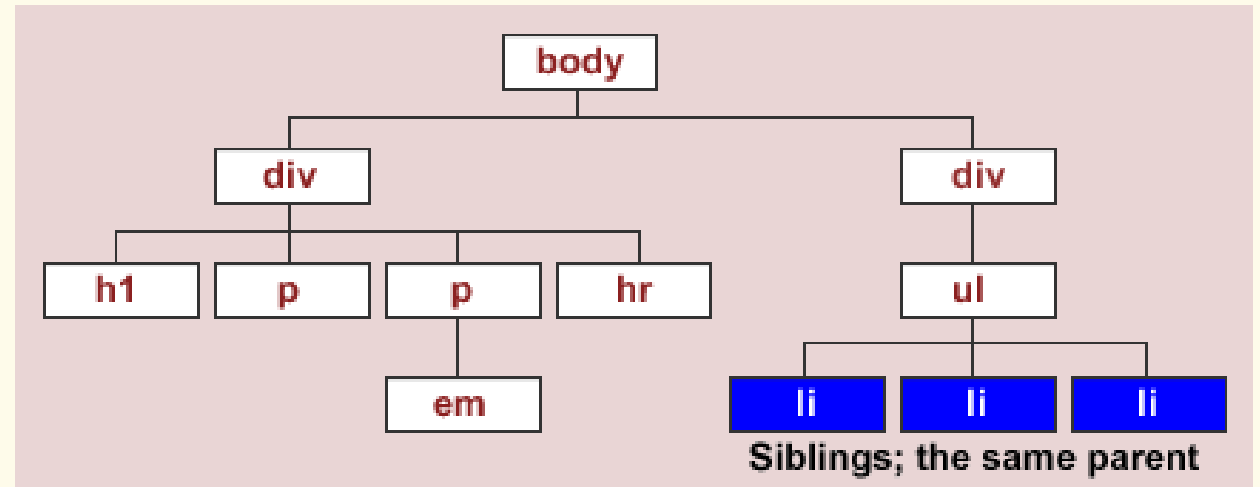
Parent and Child

- A parent is an element that is directly above and connected to an element in the document tree
- In this diagram, the `<div>` is a **parent** to the ``
- The `` is the **child** of the `<div>`



Siblings

- A sibling is an element that shares the same parent with another element.
- In this diagram, the `` elements are **siblings** as they all share the same parent - the ``.



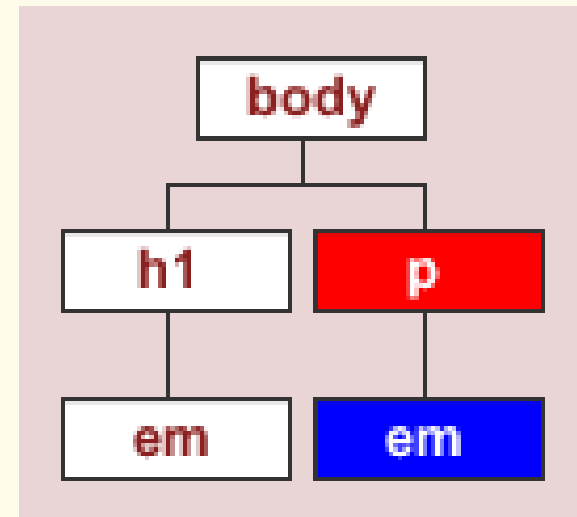
CSS Combinators

- Now that we understand how the relationships between HTML elements work, let's go back to our combinators:
 - Descendant selector
 - Child selector
 - Sibling selector
 - Adjacent sibling selector

Descendant Selector

- Descendant selectors are used to target descendants of an element
- In this diagram, the `` element is a descendant of the `<p>` element (meaning we have an em inside a paragraph)
- We specify this by adding a **space** between the elements in the CSS selector

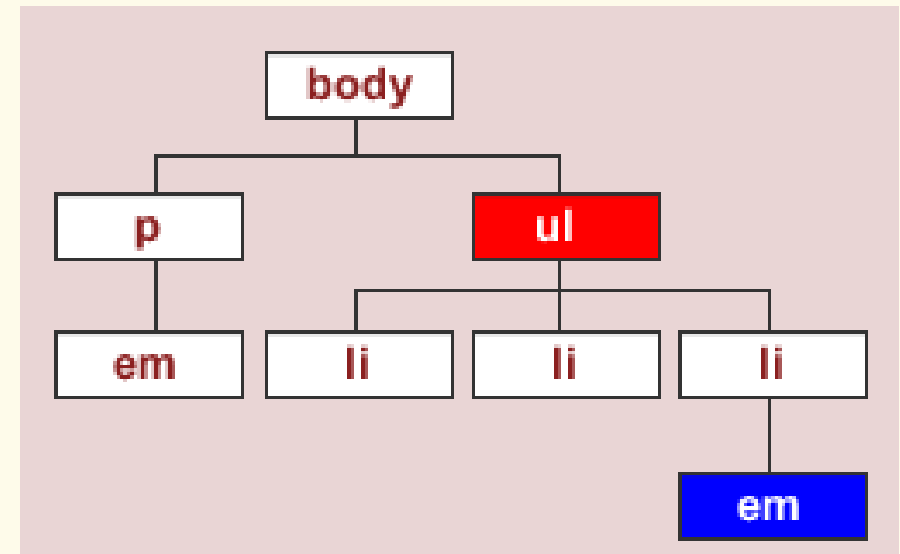
```
p em {  
    color: blue;  
}
```



Descendant Selector

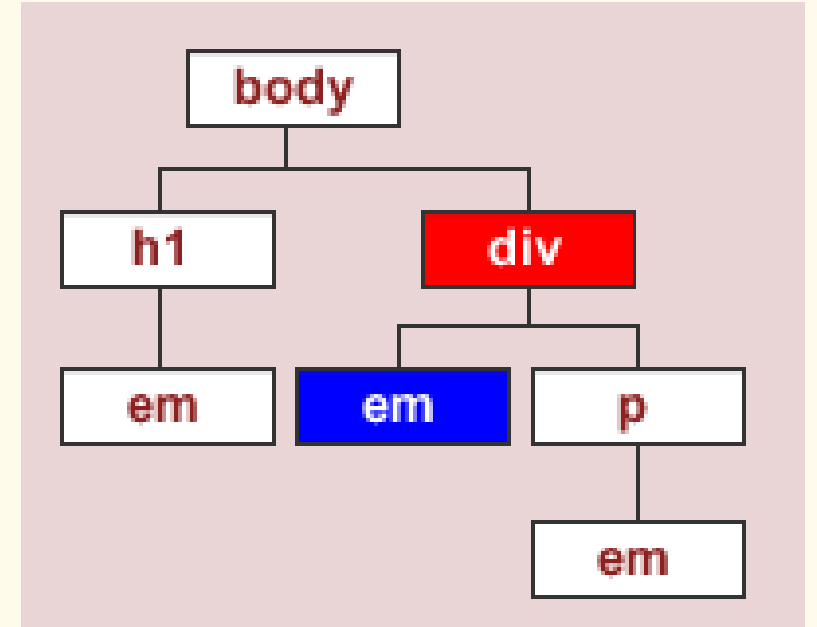
- Descendants don't have to be the direct child of an element; they can be many levels down the tree
- This example will target the `` inside the `` (even though there is an `` in between)

```
ul em {  
    color: blue;  
}
```



Child Selector

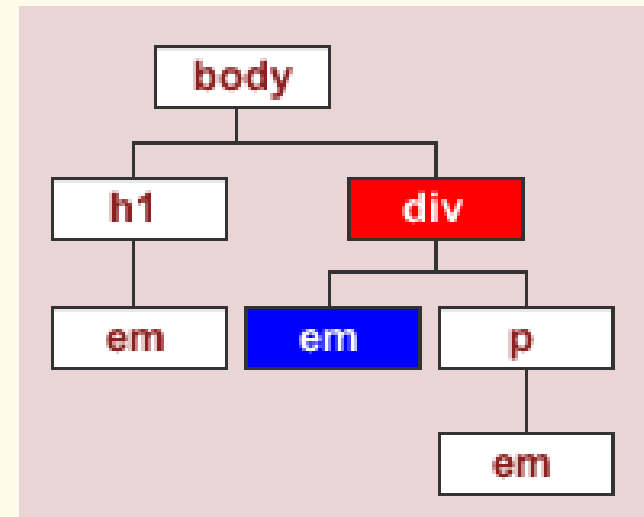
- Sometimes we do want to target the direct child only – not all the descendants
- For example, imagine that we only want to target an `` directly within a `<div>` (and *not* the `` further down the tree, inside the `<p>`)



Child Selector

- We can do this in CSS by putting a `>` between the elements in the selector, as shown here
- This will target direct children only (not all descendants)

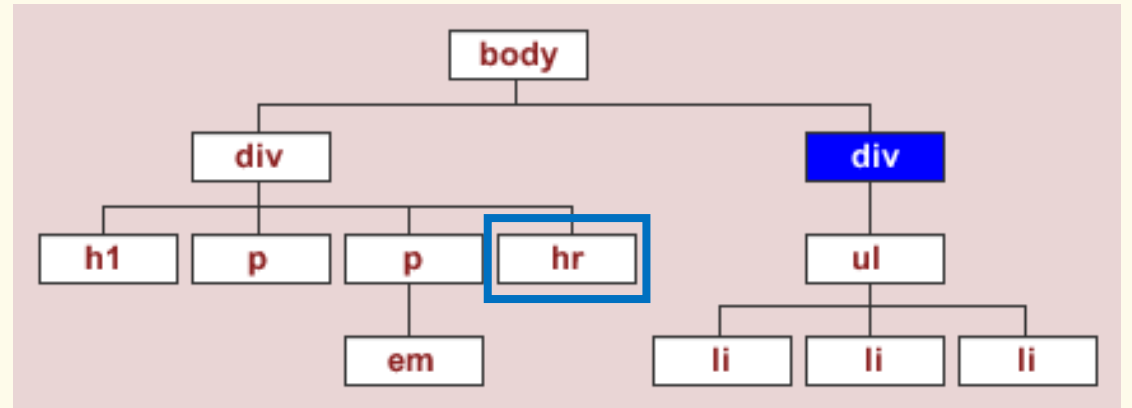
```
div > em {  
    color: blue;  
}
```



Sibling Selector

- The general sibling combinator (~) separates two selectors and matches all iterations of the **second** element, that are following the first element (though not necessarily immediately), and are children of the same parent element.
- The following example selects all `<hr>` elements that are siblings of `<h1>` elements

```
h1 ~ hr {  
    border: solid blue 2px;  
}
```



Adjacent Sibling Selector

- Lastly, we can also target **adjacent** siblings using the + character
- The adjacent sibling combinator (+) separates two selectors and matches the **second** element only if it immediately follows the first element, and both are children of the same parent element.
- For example, you may wish to target an <h3> element, but only <h3> elements that immediately follow an <h2> element.
- This has a real-world application; there is often too much space between <h2> and <h3> elements when they appear immediately after each other.

```
h2 + h3 {  
    margin: -1em;  
}
```

