Internet Applications Design and Implementation

2016 - 2017 - 2nd edition

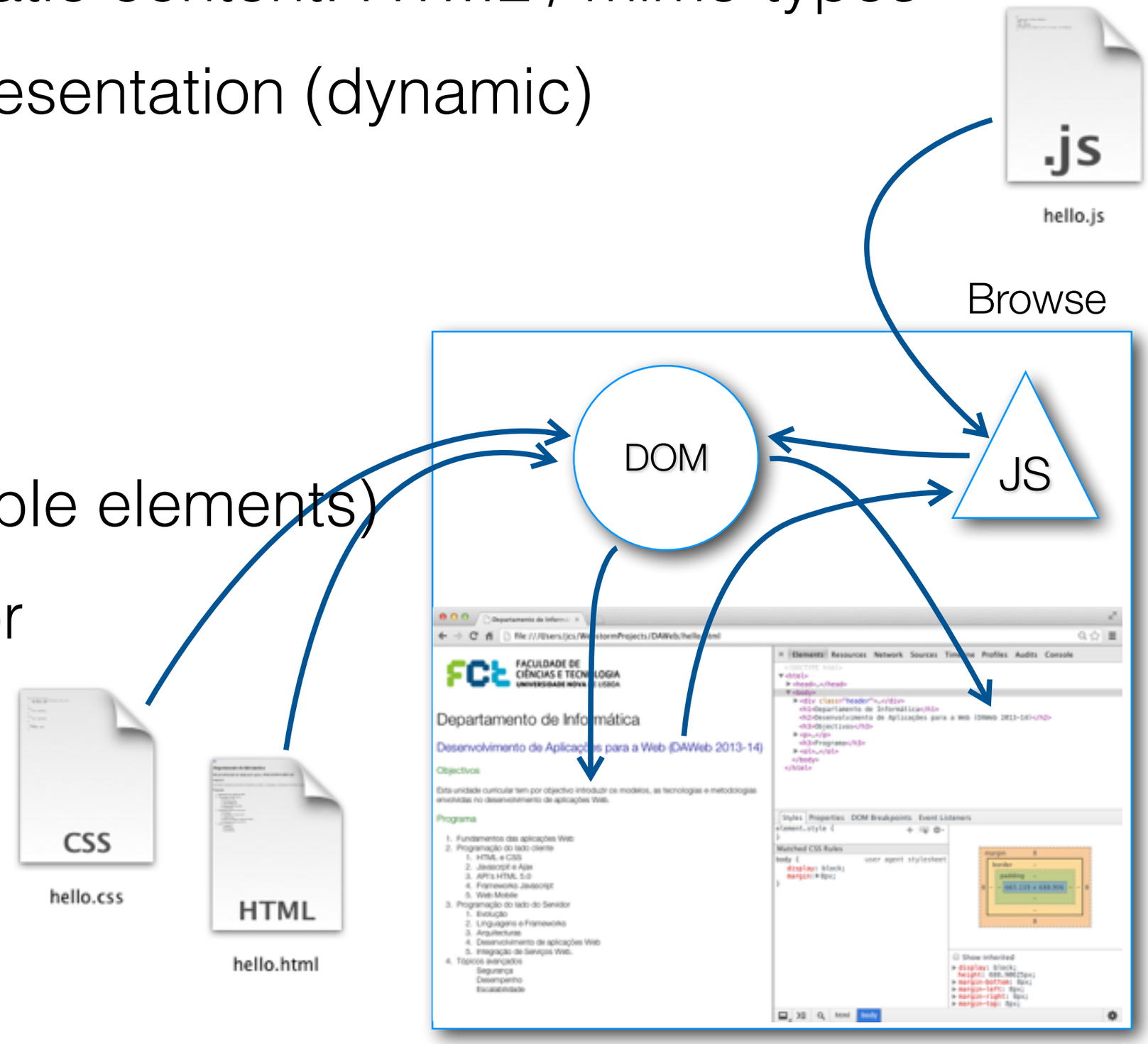
(2 - Client applications HTML/CSS/JS)

MIEI - Integrated Master in Computer Science and Informatics
Specialization block

João Costa Seco (joao.seco@fct.unl.pt)
Jácome Cunha (jacome@fct.unl.pt)

Browser logic architecture

- Source files with static content: HTML / mime-types
- DOM abstract representation (dynamic)
  - Elements
  - Style annotations
  - Event handlers
- User-interface (visible elements)
- JavaScript behavior

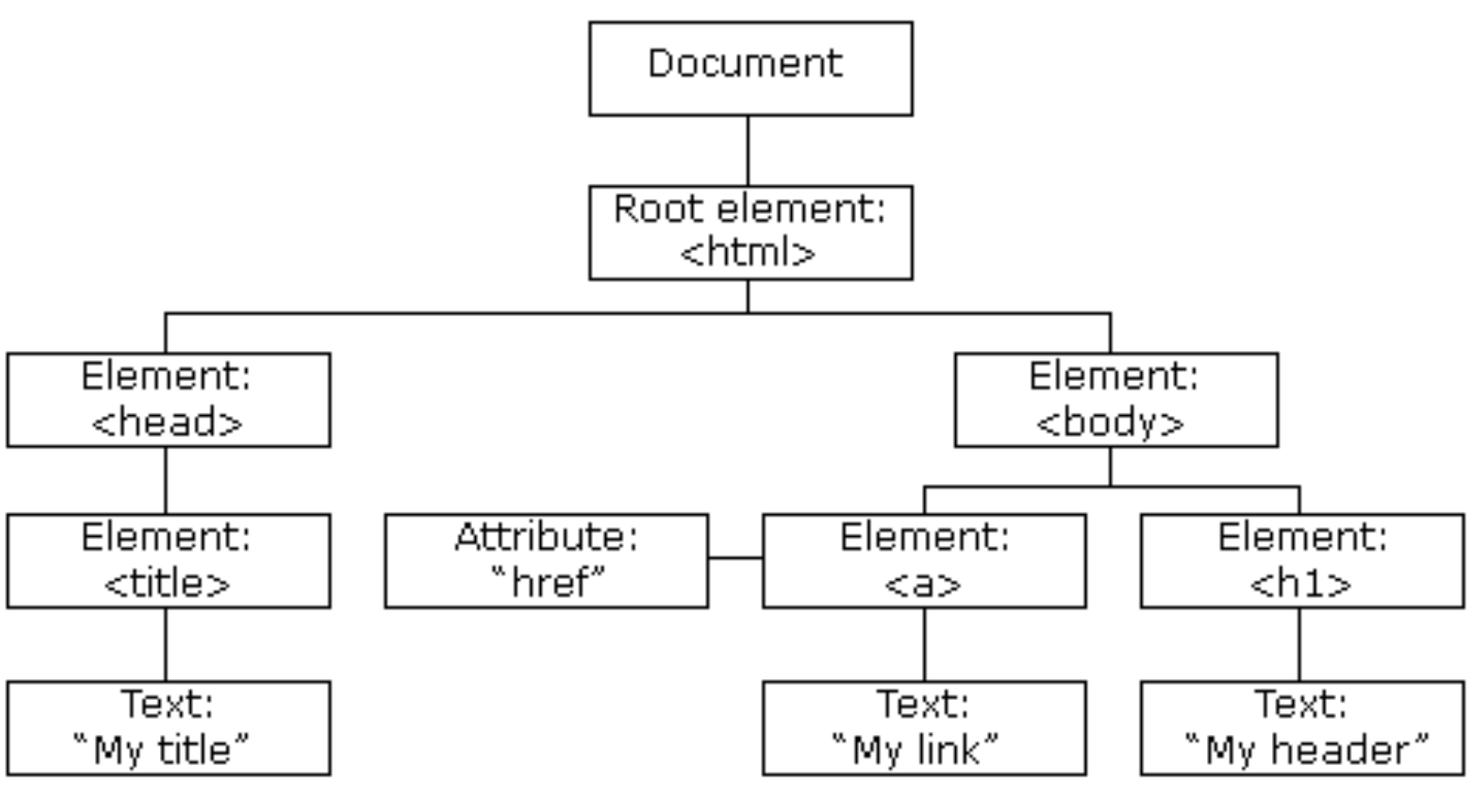


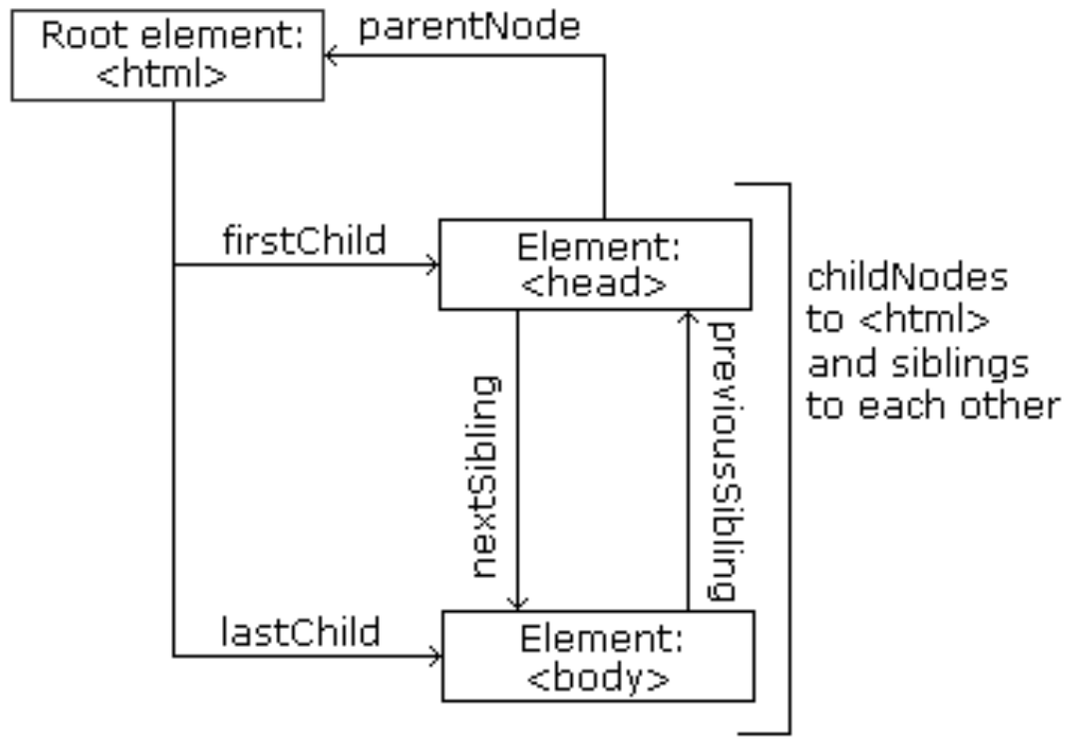
HTML5 comes as a package

- HTML5 apps are divided into:
  - HTML code to define structure and basic content to web pages
    - new tags like `<video>` `<audio>` `<svg>` `<header>`
  - CSS (Cascading Style Sheets) to define the appearance, layout, and some behavior of DOM elements
  - JavaScript defines behavior associated to the webpage elements (including dynamic construction of pages)
- The supporting structure for all these elements is the DOM

DOM - Document Object Model

- Standard (W3C) specification of the API of the data structure representing a structured document (HTML, XML, etc.).
- Defines objects and properties for all elements of a page and methods to access and modify them.
- Allows the dynamic retrieval, constructions, and modification of HTML elements in a web page.





http://www.w3schools.com/htmldom

DOM - Document Object Model

- DOM is a convention to represent HTML (XML) documents in a tree of objects.
- It's the **dynamic** supporting structure of a web page.
- Each node of the DOM has a particular structure (attributes) and associated behavior (methods).
- DOM objects can be created, accessed, and modified using:
  - HTML: the static structure of DOM objects
  - CSS: using (rich) object selectors
  - JavaScript: by traversing the DOM tree using the DOM API.

Let’s look at the trailer

https://bitbucket.org/jacome/ciai1617/src/5707acbb5ab6f8375691f5e8e74bcd47c77bc4be/docs/html-css-js/?at=master


HTML

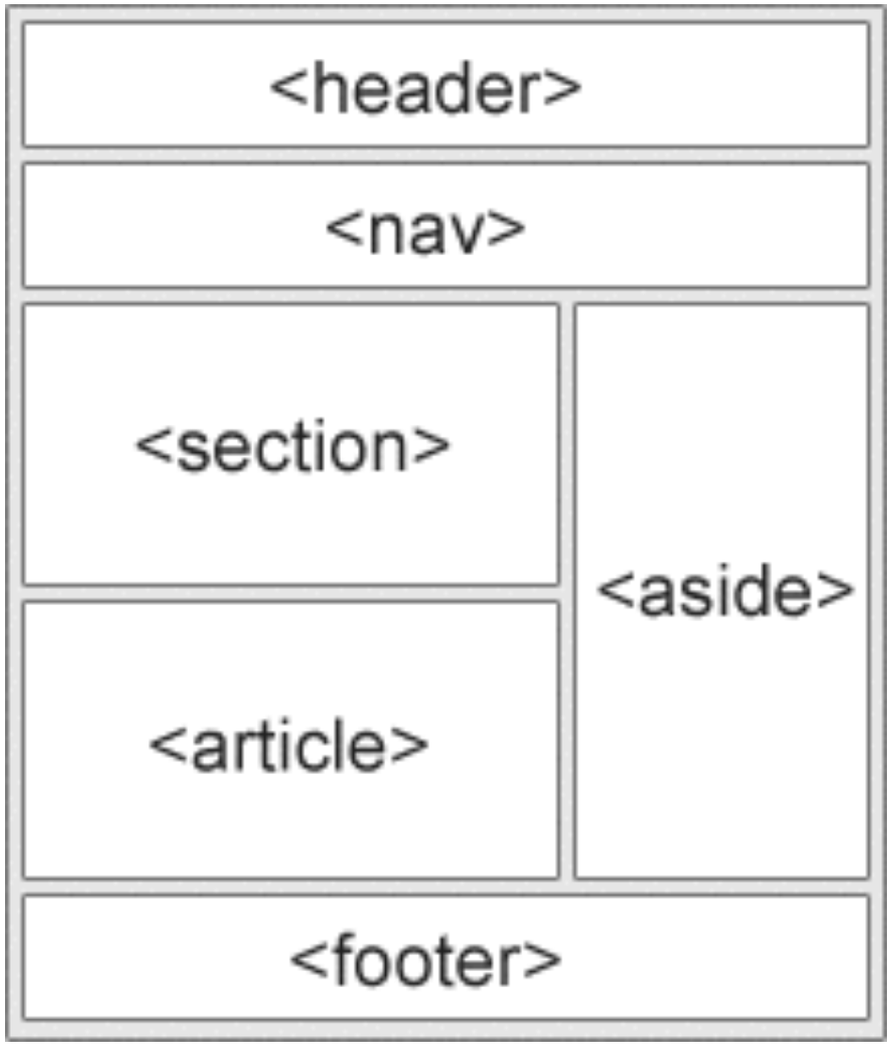
HTML

- Each HTML element has a tag
  - **<body>   ** …
- Content
  - **This is a paragraph**
- and a set of attributes:
  - Generic: set on any HTML element. (ex: **id**, **class**, **hidden**, …)
  - Particular: img/**src**, option/**value**, etc.
  - Events: dependent on the kind of element
    body / **onload**, button / **onclick**, etc.

HTML essentials in one slide

- <!DOCTYPE html>
- <html></html> root element
- <head></head> section to declare meta-information, stylesheets and scripts
- <body></body> section to define content
- <p> <h1>… block elements
-   <i> inline elements
- < > é escaped characters

HTML **5** semantic elements



HTML **5** semantic elements (part of)

Tag	Description
<article>	Defines an article in the document
<figcaption>	Defines a caption for a <figure> element
<figure>	Defines self-contained content, like illustrations, diagrams, photos, code listings, etc.
<footer>	Defines a footer for the document or a section
<header>	Defines a header for the document or a section
<main>	Defines the main content of a document
<menuitem>	Defines a command/menu item that the user can invoke from a popup menu
<nav>	Defines navigation links in the document
<section>	Defines a section in the document

Trailer again :)

Actually, let’s construct it!

https://bitbucket.org/jacome/ciai1617/src/5707acbb5ab6f8375691f5e8e74bcd47c77bc4be/docs/html-css-js/?at=master

- What's structural?
  - …
- What's style?
  - …
- What's behavior?
  - …

General structure - <html>


```
<!DOCTYPE html>
<html>
  <head>
      ...
  </head>

  <body>
      ...
  </body>
</html>
```

Invisible info - <head>


```html
<head>
    <!-- head defines invisible information about the page -->
    <!-- Meta information (machine parseable) can also be defined -->
    <meta charset="UTF-8">
    <meta name="description" content="My Home Library Web Site">
    <meta name="keywords" content="HTML,CSS,XML,JavaScript">
    <meta name="author" content="João Costa Seco">
    <meta http-equiv="refresh" content="30">

    <!-- The title of the browser window (mandatory) -->
    <title>My home library</title>

    <!-- To define the base URL for all relative links -->
    <base href="ctp.di.fct.unl.pt/~jcs/daweb14-15">

    <!-- To define the style of the elements of the page -->
    <style>
        * { font-family: sans-serif;}
    </style>
    <!-- Or import an external file with the stylesheet -->
    <link href="screen.css" type="text/css" rel="stylesheet" media="screen"/>

    <!-- Custom behaviour can be defined using Javascript -->
    <!-- Directly in the HTML code -->
    <script>
        function Hello() { alert("Hello World."); }
    </script>
    <!-- Or by importing a script file -->
    <script src="books.js"></script>
</head>
```

Visible info - <body> - **old style**

```
<div id="header">
  <h1>Monday Times</h1>
</div>

<div id="menu">
  <ul>
    <li>News</li>
    <li>Sports</li>
    <li>Weather</li>
  </ul>
</div>

<div id="content">
  <h2>News Section</h2>
  <div class="article">
    <h2>News Article</h2>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque in
porta lorem. Morbi condimentum est nibh, et consectetur tortor feugiat at.</p>
  </div>
  <div class="article">
    <h2>News Article</h2>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque in
porta lorem. Morbi condimentum est nibh, et consectetur tortor feugiat at.</p>
  </div>
</div>

<div id="footer">
  <p>&amp;copy; 2016 Monday Times. All rights reserved.</p>
</div>
```

Visible info - <body> - **new style**

```
<header>
<h1>Monday Times</h1>
</header>

<nav>
<ul>
<li>News</li>
<li>Sports</li>
<li>Weather</li>
</ul>
</nav>

<section>
<h2>News Section</h2>
<article>
<h2>News Article</h2>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque in
porta lorem. Morbi condimentum est nibh, et consectetur tortor feugiat at.</p>
</article>
<article>
<h2>News Article</h2>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque in
porta lorem. Morbi condimentum est nibh, et consectetur tortor feugiat at.</p>
</article>
</section>

<footer>
<p>&copy; 2014 Monday Times. All rights reserved.</p>
</footer>
```

Textual block elements


```
<html>
    <body>
        <h1>This is a heading</h1>
        <p>This is a paragraph.</p>
        <p>This is another paragraph.</p>
    </body>
</html>
```

Textual inline elements


```
<html>
    <body>
        <h1>This is a heading</h1>

        <p>This is a paragraph. <span>This is a span</span> <b>This is a strong tag</b>
        <a>This is an anchor</a> </p>

        <p>This is another paragraph.</p>

    </body>
</html>
```

Structural block elements


```
<html>
  <body>
    <nav>This is a navigation section</nav>
    <article>This is a document section.
      <section>This is another section. </section>
    </article>
    <article>This is a document section.
      <section>This is another section. </section>
      <section>This is another section. </section>
    </article>
  </body>
```

Block elements

This is a block element.

This is a block element.

This is a block element.

Inline elements


```
<span>This is </span> <span> an inline </span> <span> element.
</span> <span>This is </span> <span> an inline </span> <span>
element. </span> <span>This is </span> <span> an inline </span>
<span> element. </span>
```

Attributes

HTML elements can be modified through attributes

```
<a href="http://www.w3schools.com">This is a link</a>

<img src="w3schools.jpg" alt="W3Schools.com" width="104" height="142">

<div style="border:1px">This is a section.</div>
```

HTML <button> Tag

« Previous Complete HTML Reference Next »

Example

A clickable button is marked up as follows:

```
<button type="button">Click Me!</button>
```

Try it Yourself »

Attributes

= New in HTML5.

Attribute	Value	Description
autofocus	autofocus	Specifies that a button should automatically get focus when the page loads
disabled	disabled	Specifies that a button should be disabled
form	*form_id*	Specifies one or more forms the button belongs to

HTML - Basic Interaction

- Links
  - Produce a GET request to the given URL and replace the entire content of the DOM
- Forms
  - Pre-defined controls (buttons)
  - Produce a request and expects a response document (HTML)
  - replaces the entire content of the DOM (depends on _target)
  - method attribute defines the request type (GET/POST)
  - called URL format depends on the used method (body/query string)

HTML - FORMS


```
<form action="/my-handling-form-page" method="post">
    <div>
        <label for="name">Name:</label>
        <input type="text" id="name" />
    </div>
    <div>
        <label for="mail">E-mail:</label>
        <input type="email" id="mail" />
    </div>
    <div>
        <label for="msg">Message:</label>
        <textarea id="msg"></textarea>
    </div>

    <div class="button">
        <button type="submit">Send your message</button>
    </div>
</form>
```

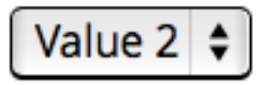
HTML

- Inputs

```
<input type="text" name="input" value="Type here">
```



```
<button name="button">Click me</button>
```




```
<select name="select">
  <option value="value1">Value 1</option>
  <option value="value2" selected>Value 2</option>
  <option value="value3">Value 3</option>
</select>
```

- Other interface elements (output)

```
<p>Heat the oven to <meter min="200" max="500" value="350">350 degrees</meter>.
```

```
<progress value="70" max="100">70 %</progress>
```

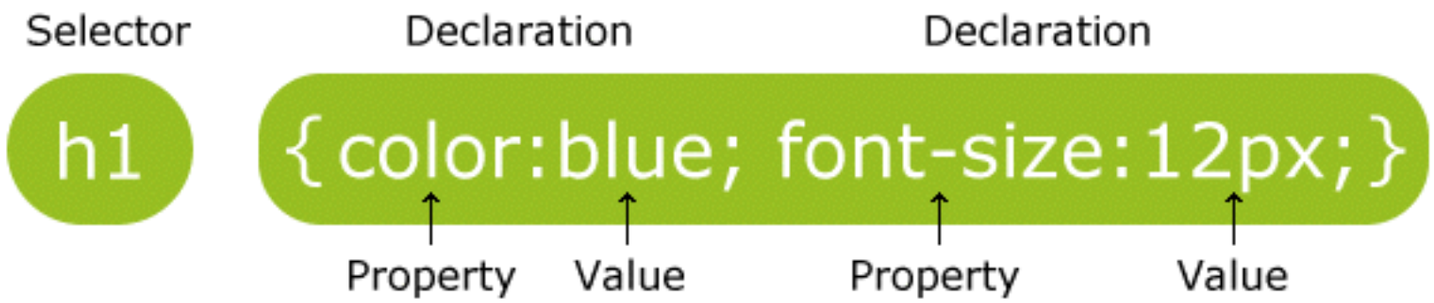
Let’s do an example!

CSS - Cascading Style Sheets

- Describes the presentation of a document defined using a markup language (HTML, XML)
- Decouples presentation from the document structure and behavior of a web application
- Based on declarative object descriptors and object attributes (and values) - Rules
- Based on general application priorities on conflicting cases
- Includes basic presentation behavior (animations)

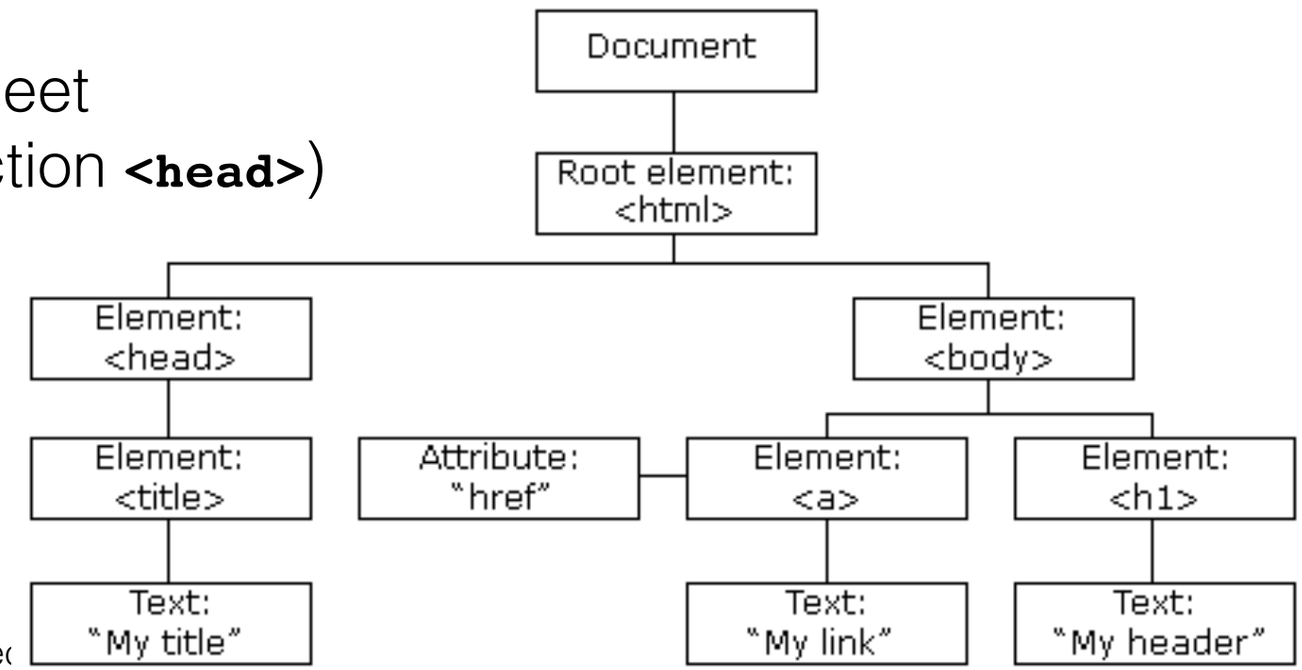
Decorating the DOM

- A set of rules is applied to the DOM



- rule = selector + style definitions with the form **`property:value`**

- Rules are applied by the following order
  - Browser default
  - External and Internal Style Sheet (following the order in the section **`<head>`**)
  - Inline Style (HTML element)
- Propagate hierarchically through the DOM



CSS Selectors

from http://www.w3schools.com/cssref/css_selectors.asp

Selector	Example	Example description	CSS
.class	.intro	Selects all elements with class="intro"	1
#id	#firstname	Selects the element with id="firstname"	1
*	*	Selects all elements	2
element	p	Selects all <p> elements	1
element,element	div, p	Selects all elements and all <p> elements	1
element element	div p	Selects all <p> elements inside elements	1
element>element	div > p	Selects all <p> elements where the parent is a element	2
element+element	div + p	Selects all <p> elements that are placed immediately after elements	2
element1~element2	p ~ ul	Selects every element that are preceded by a <p> element	3
[*attribute*]	[target]	Selects all elements with a target attribute	2
[*attribute=value*]	[target=_blank]	Selects all elements with target="_blank"	2
[*attribute~=value*]	[title~=flower]	Selects all elements with a title attribute containing the word "flower"	2
[*attribute|=value*]	[lang|=en]	Selects all elements with a lang attribute value starting with "en"	2
[*attribute^=value*]	a[href^="https"]	Selects every <a> element whose href attribute value begins with "https"	3

CSS object selectors

- all elements:

```
* { font-family: sans-serif; }
```

- by HTML tag

```
h1, h2 { color: #f0f0f0; }
```

- by id

```
#form1 { border: solid 1px; }
```

- by class

```
.listitem { list-style: none; }
```

- composed selector

```
#booklist li { background-color: "rbg(100,80,10)"; }
```

- composed selector

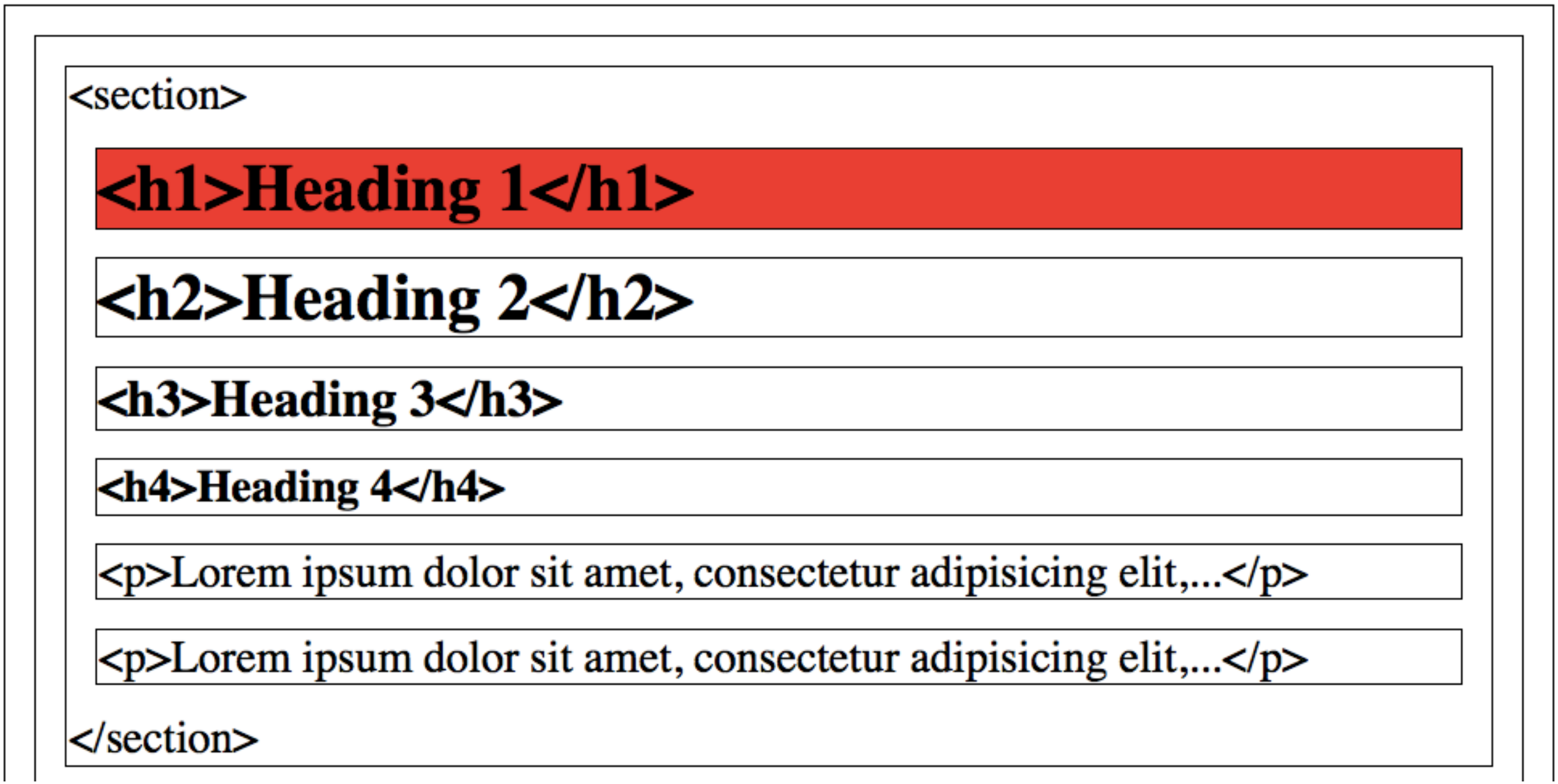
```
ul.menu > li { margin: 10px; padding: 5px;}
```

- pseudo-classes

```
.menuitem:hover { background-color: blue; }
```

CSS Selectors

```
h1 {background-color: red;}
```



CSS Selectors

```
h1, h2, h4 {background-color: red;}
```

<section>

<h1>Heading 1</h1>

<h2>Heading 2</h2>

<h3>Heading 3</h3>

<h4>Heading 4</h4>

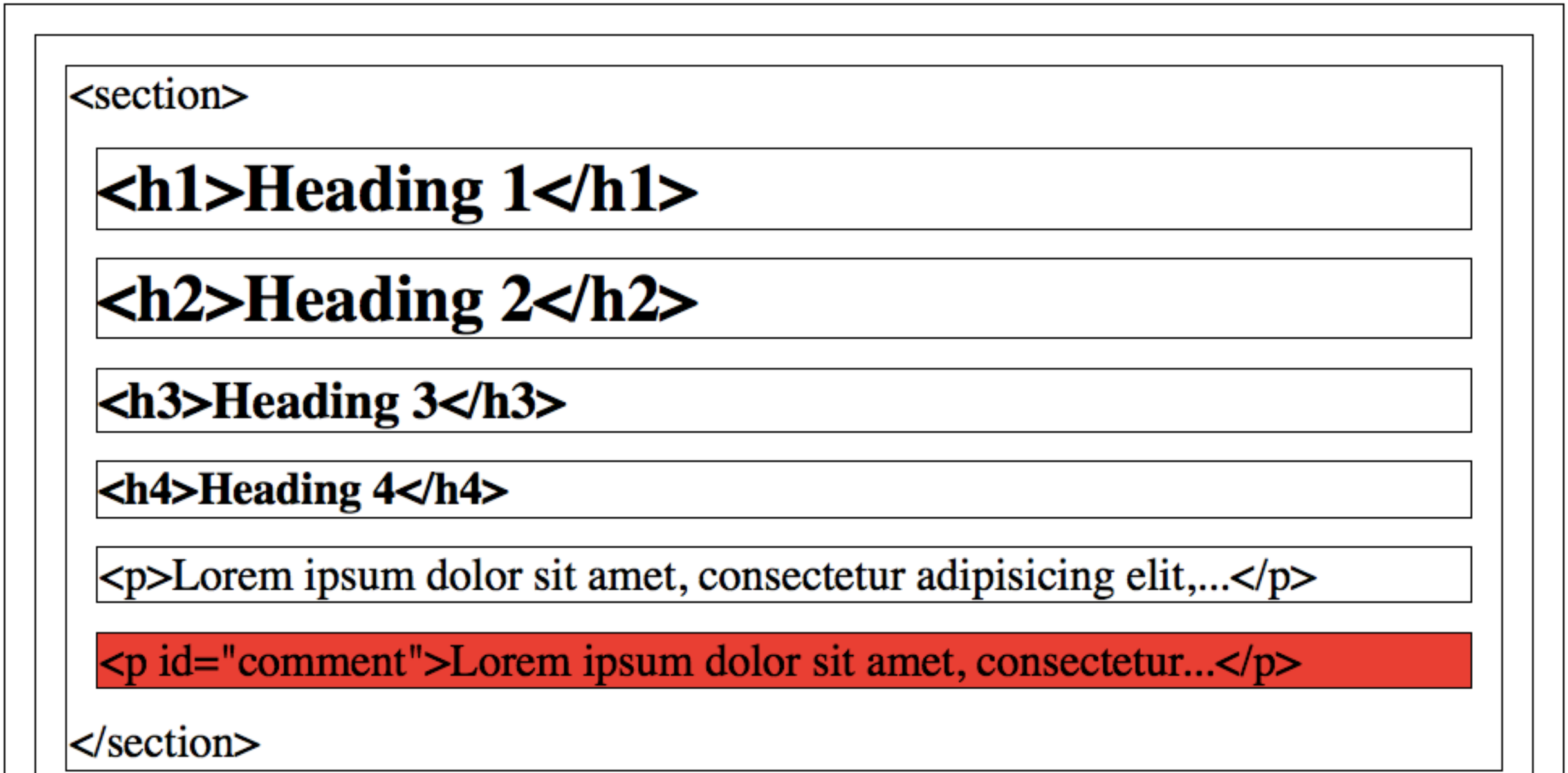
Lorem ipsum dolor sit amet, consectetur adipisicing elit,...

Lorem ipsum dolor sit amet, consectetur...

</section>

CSS Selectors

```
#comment {background-color: red;}
```



CSS Selectors

```
.text {background-color: red;}
```

<section>

<h1>Heading 1</h1>

<h2>Heading 2</h2>

<h3>Heading 3</h3>

<h4>Heading 4</h4>

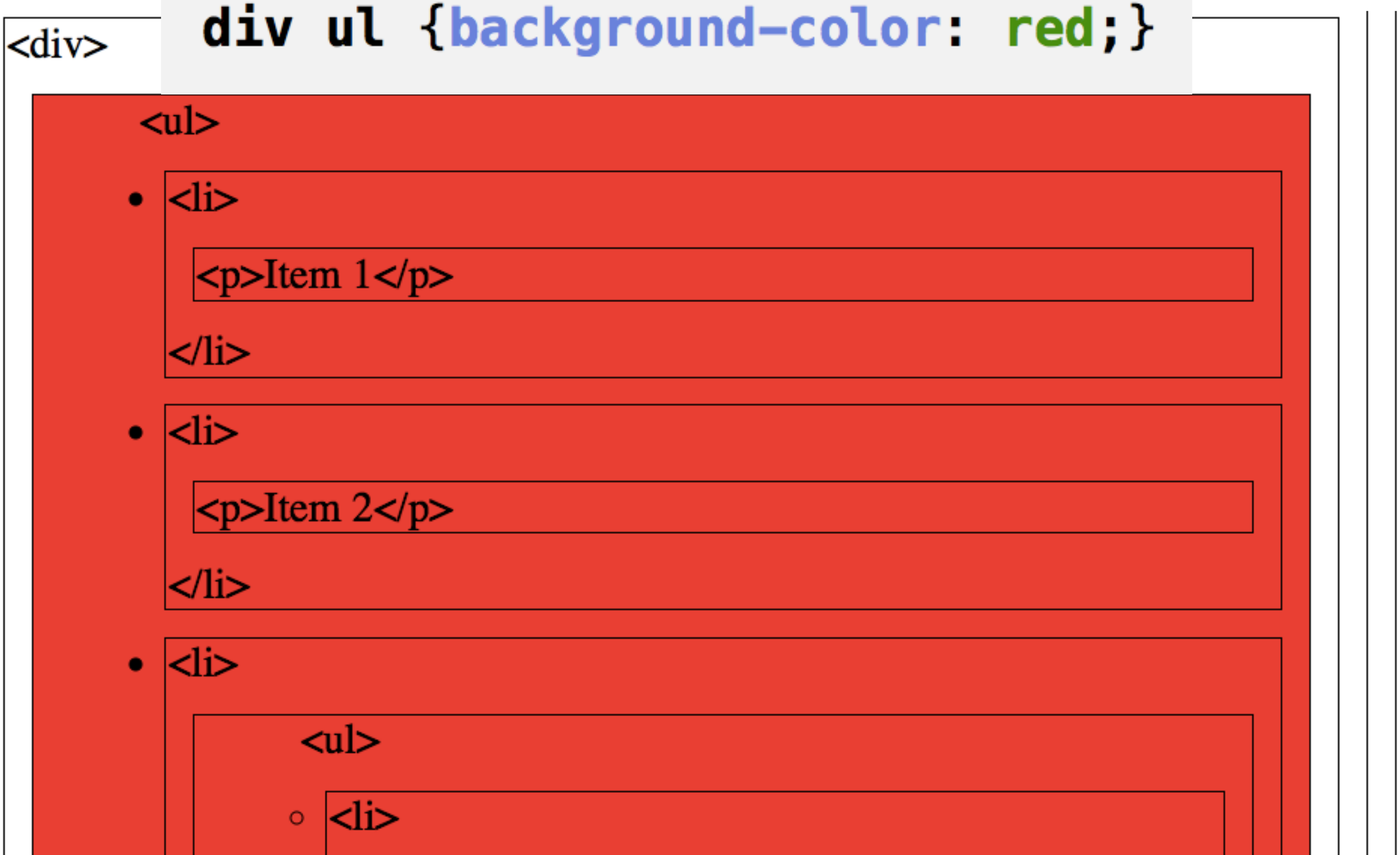
Lorem ipsum dolor sit amet, consectetur...

Lorem ipsum dolor sit amet, consectetur...

</section>

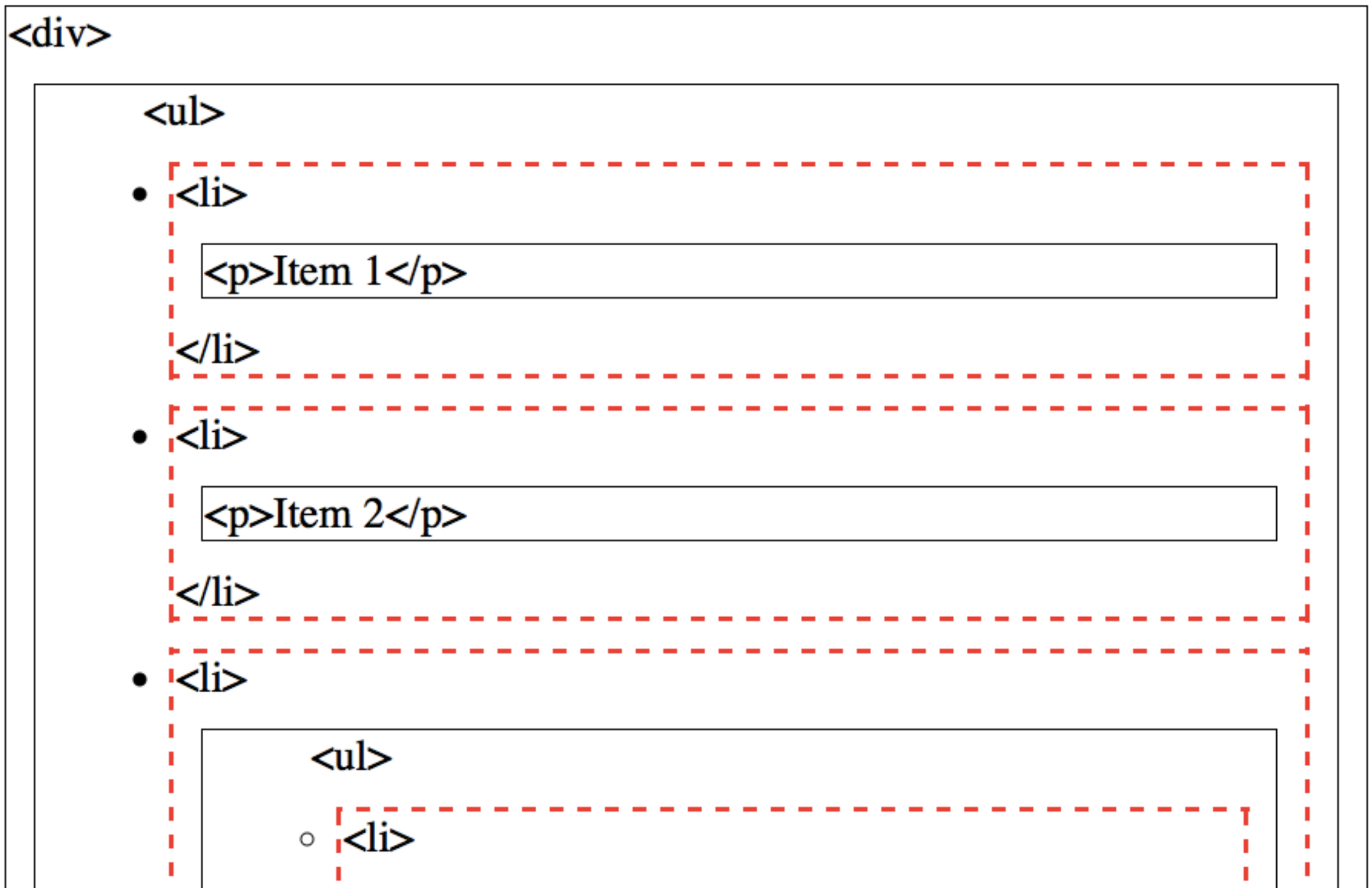
CSS Selectors

```
div ul {background-color: red;}
```



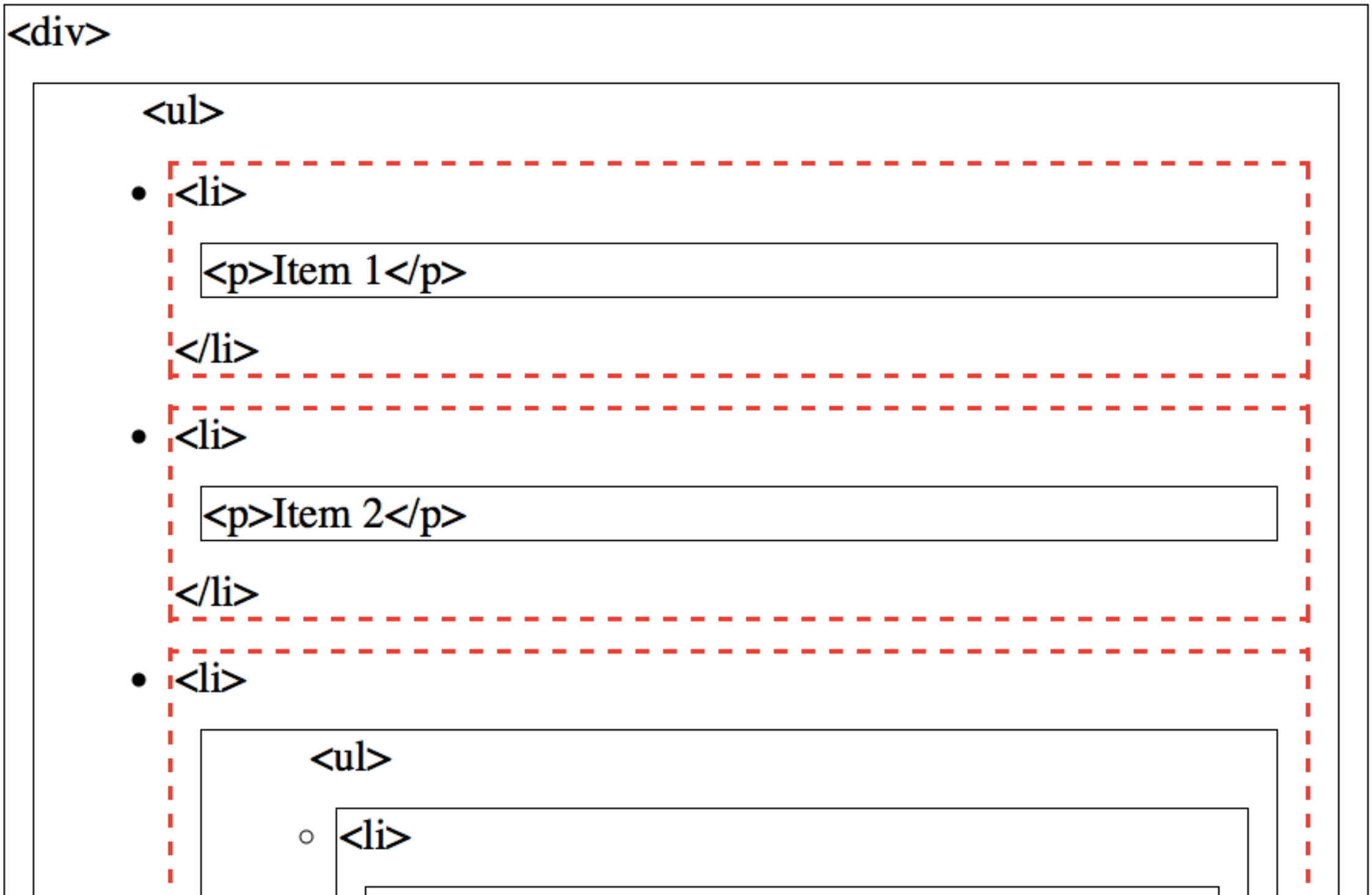
CSS Selectors

```
div li {border: dashed red 2px;}
```



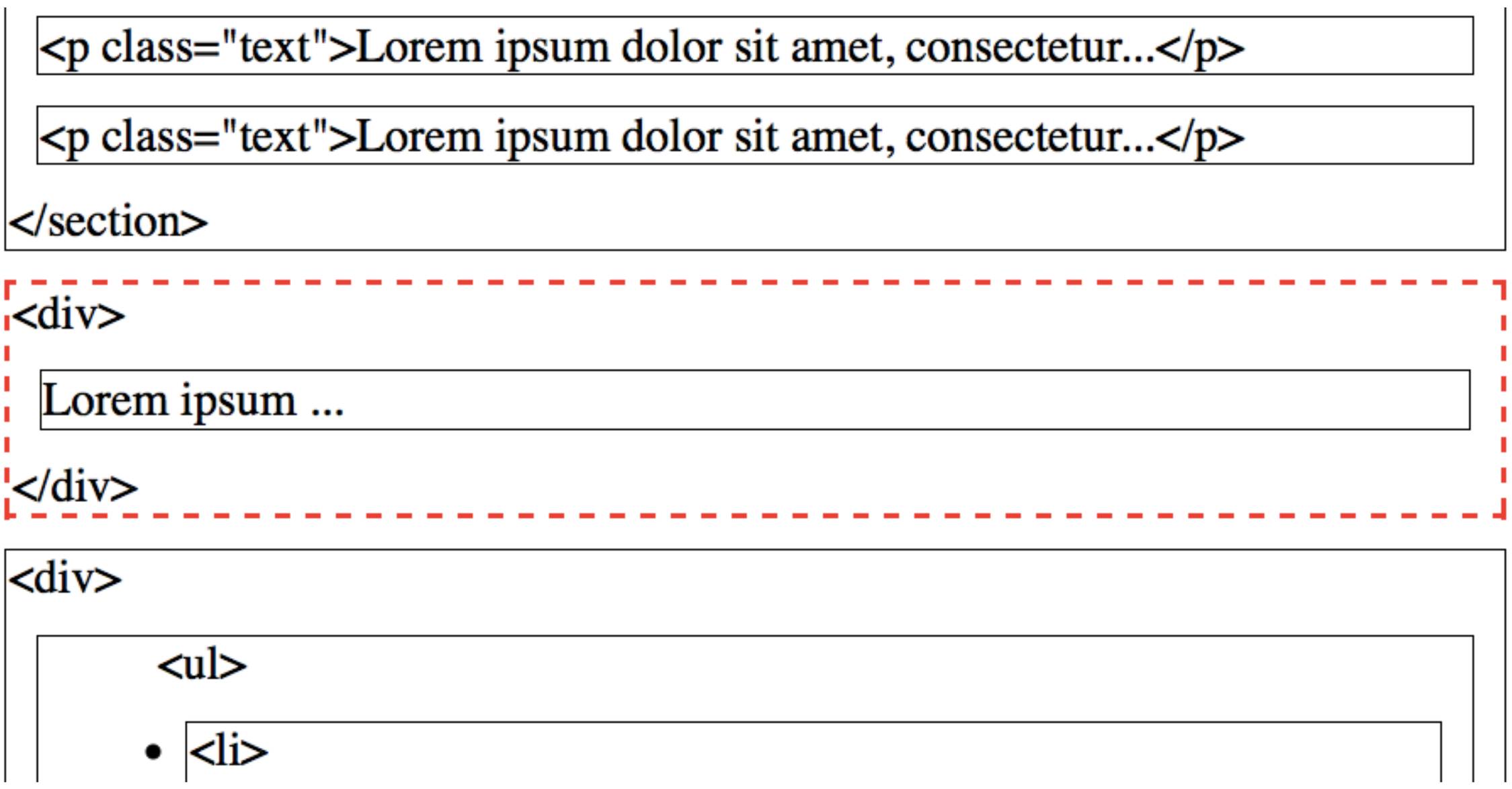
CSS Selectors

```
div > ul > li {border: dashed red 2px;}
```



CSS Selectors

```
section+div {border: dashed red 2px;}
```



CSS Selectors

```
div~div {border: dashed red 2px;}
```

Lorem ipsum dolor sit amet, consectetur...

Lorem ipsum dolor sit amet, consectetur...

</section>

Lorem ipsum ...

-

Item 1

CSS Properties

CSS Properties

CSS Property Groups

- Color
- Background and Borders
- Basic Box
- Flexible Box
- Text
- Text Decoration
- Fonts
- Writing Modes
- Table
- Lists and Counters
- Animation
- Transform
- Transition
- Basic User Interface
- Multi-column
- Paged Media
- Generated Content
- Filter Effects
- Image/Replaced Content
- Masking
- Speech
- Marquee

CSS Properties

Color Properties

Property	Description	CSS
color	Sets the color of text	1
opacity	Sets the opacity level for an element	3

Background and Border Properties

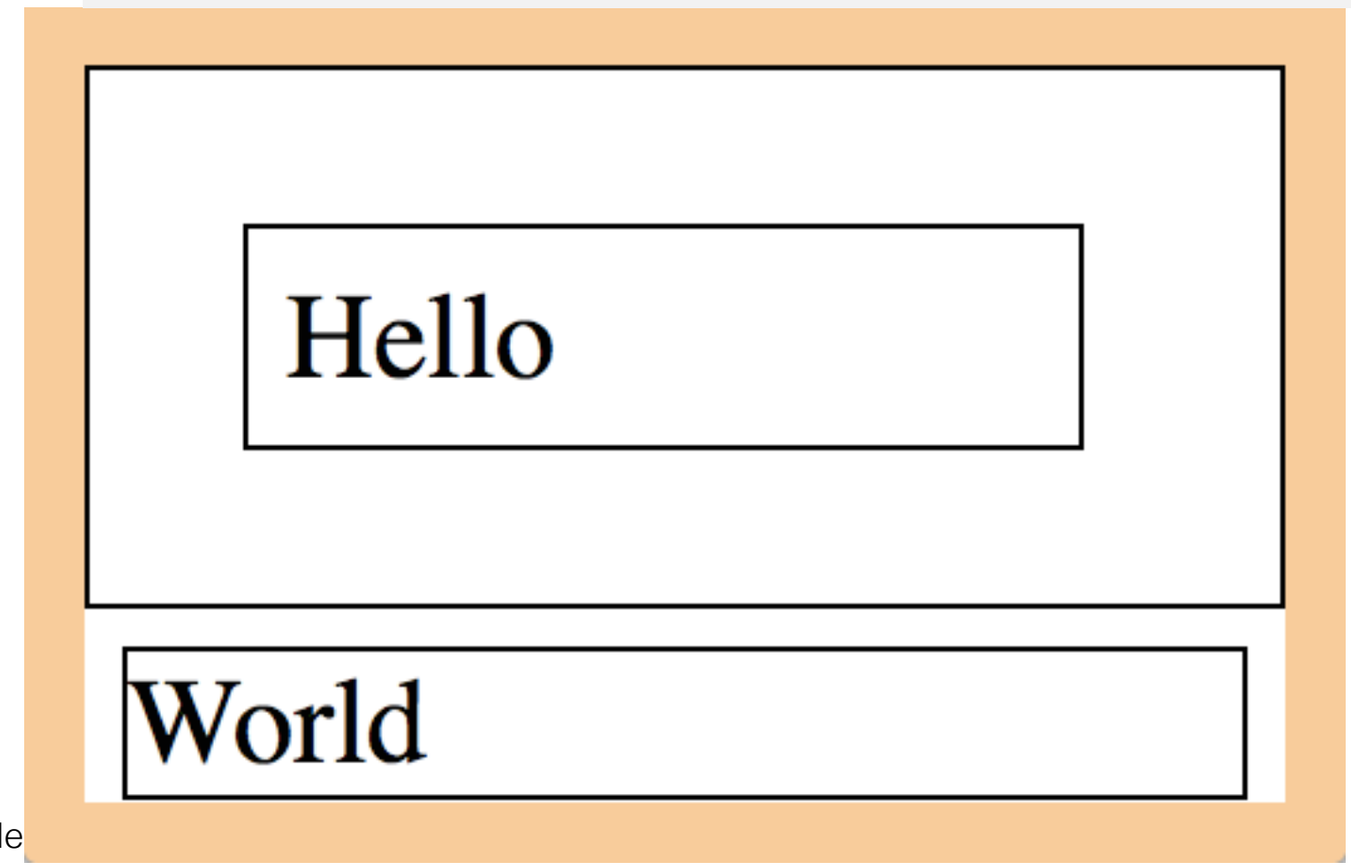
Property	Description	CSS
background	Sets all the background properties in one declaration	1
background-attachment	Sets whether a background image is fixed or scrolls with the rest of the page	1
background-color	Sets the background color of an element	1
background-image	Sets the background image for an element	1
background-position	Sets the starting position of a background image	1
background-repeat	Sets how a background image will be repeated	1
background-clip	Specifies the painting area of the background	3
background-origin	Specifies the positioning area of the background images	3
background-size	Specifies the size of the background images	3
border	Sets all the border properties in one declaration	1

Example, yet again :)

The Box Model

- The layout of web pages is based on the setting of box dimensions, using 4 different measures
  - Width and height; padding; border; margin

```
#a1 { width:100px; margin:20px; padding: 5px;}
#a2 { margin: 5px; }
```



The Box Model


```
div {
    width: 300px;
    padding: 25px;
    border: 25px solid navy;
    margin: 25px;
}
```

```
div {
    box-sizing: border-box;
    width: 50%;
    float: left;
}
```

http://css-tricks.com/box-sizing/

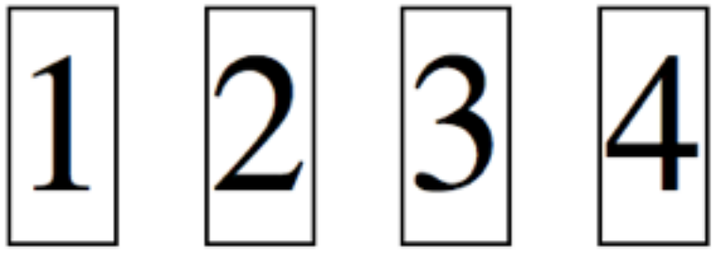
Display, Visible, Float, and Position

- There are three ways (properties) that control the positioning of an HTML element in a rendered page.
- **`display`**: inline, block, inline-block, none, etc.
- **`visibility`**: visible, hidden, collapse (tables)
- **`float`**: left, right…
- **`position`**: static, absolute, fixed, relative
  - associated properties: left, right, top, bottom

Display

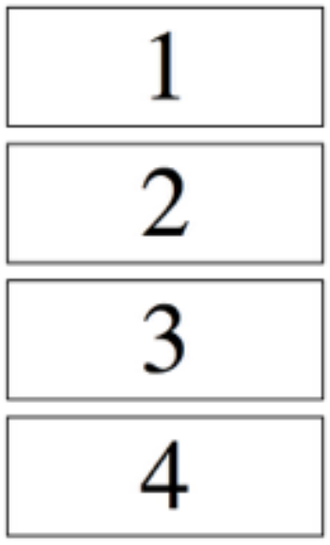
```
<div id="a1">1</div>
<div id="a2">2</div>
<div id="a3">3</div>
<div id="a4">4</div>
```

- inline



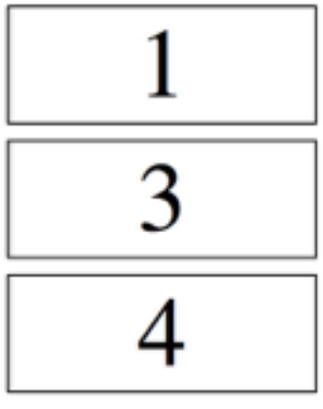

```
div { display: inline ; }
```

- block




```
div { display: block ; }
```

- none

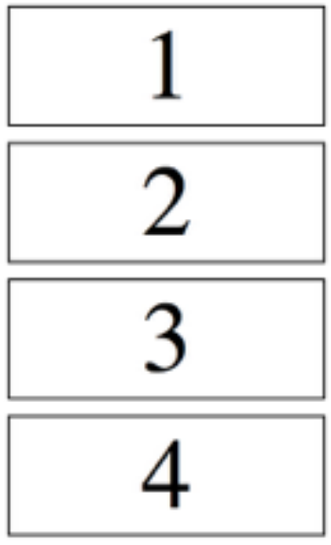



```
#a2 { display:none; }
```

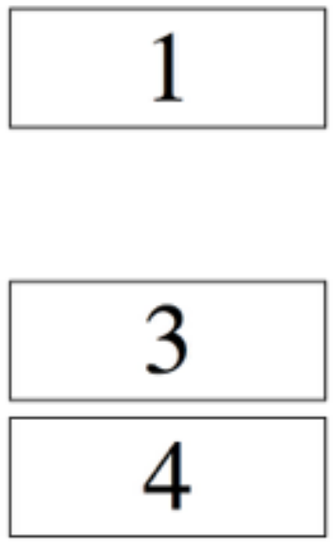
Visibility

```
<div id="a1">1</div>
<div id="a2">2</div>
<div id="a3">3</div>
<div id="a4">4</div>
```

- visible



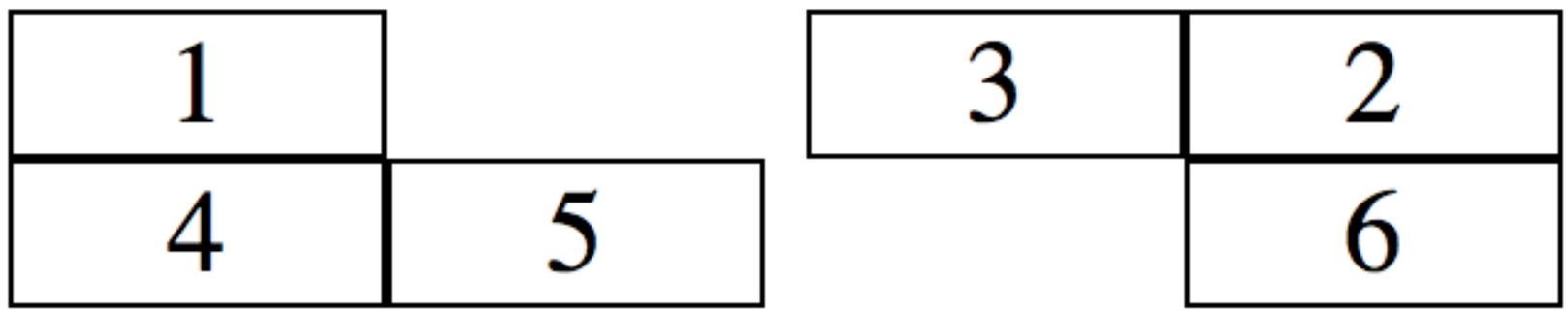
- hidden




```
#a2 { visibility:hidden;    }
```

Float & Clear

- Property **`float`** lets the elements to be arranged freely and other elements wrap around them (e.g. text)
- Possible values: left, right



- clear resets wrapping of elements
- overflow may make the container extend to the borders of the floating elements.

```
<div class="left">1</div>
<div class="right">2</div>
<div class="right">3</div>
<div class="left clear">4</d
<div class="left">5</div>
<div class="right">6</div>
```

The Box Model

Basic Box Properties

Property	Description	CSS
bottom	Specifies the bottom position of a positioned element	2
clear	Specifies which sides of an element where other floating elements are not allowed	1
clip	Clips an absolutely positioned element	2
display	Specifies how a certain HTML element should be displayed	1
float	Specifies whether or not a box should float	1
height	Sets the height of an element	1
left	Specifies the left position of a positioned element	2
overflow	Specifies what happens if content overflows an element's box	2
overflow-x	Specifies whether or not to clip the left/right edges of the content, if it overflows the element's content area	3
overflow-y	Specifies whether or not to clip the top/bottom edges of the content, if it overflows the element's content area	3
padding	Sets all the padding properties in one declaration	1
padding-bottom	Sets the bottom padding of an element	1
padding-left	Sets the left padding of an element	1
padding-right	Sets the right padding of an element	1

Responsive design

- Sets of methods and techniques to optimize reading and navigation of a certain web design to a myriad of devices and displays.
- Usually achieved by gracious resizing, hiding, and rearrangement of page sections.
  - Media queries (CSS @media rule)
  - Flexible (flow and grid) layouts (e.g. bootstrap)
  - JavaScript reactive interfaces

Media queries

- The @media rule is used to define different style rules for different media types/devices.

```
<!-- CSS media query on a link element -->
<link rel="stylesheet" media="(max-width: 800px)" href="example.css" />


<!-- CSS media query within a stylesheet -->
<style>
@media (max-width: 600px) {
  .facet_sidebar {
    display: none;
  }
}
</style>
```

https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Media_queries

Media queries

- @media <media type> and <media features>

```
@media (min-width: 700px) { ... }
```

```
@media (min-width: 700px) and (orientation: landscape) { ... }
```

```
@media tv and (min-width: 700px) and (orientation: landscape) { ... }
```

```
@media (min-width: 700px), handheld and (orientation: landscape) { ... }
```

https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Media_queries

Bootstrap

Preprocessors

Bootstrap ships with vanilla CSS, but its source code utilizes the two most popular CSS preprocessors, Less and Sass. Quickly get started with precompiled CSS or build on the source.

One framework, every device.

Bootstrap easily and efficiently scales your websites and applications with a single code base, from phones to tablets to desktops with CSS media queries.

Full of features

With Bootstrap, you get extensive and beautiful documentation for common HTML elements, dozens of custom HTML and CSS components, and awesome jQuery plugins.

Bootstrap Grid System

- Bootstrap is a CSS/Javascript client framework that provides simpler and effective layout tools.
- Bootstrap divides a page into 12 columns and provides classes for rows and columns:
  - `container`, `row`, `col-X-Y`, etc.
    - X: **xs** (phones), **sm** (tablets), **md** (desktops), and **lg** (larger desktops)
    - Y: 1 .. 12
    - Explore bootstrap grids at: http://getbootstrap.com/examples/grid/

Bootstrap Grid System

span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1
span 4				span 4				span 4			
span 4				span 8							
span 6						span 6					
span 12											

ST

JavaScript in the browser

```
<h1>JavaScript Can Validate Input</h1>

<p>Please input a number between 1 and 10:</p>

<input id="numb" type="number">

<button type="button" onclick="myFunction()">Submit</button>

<p id="demo"></p>

<script>
function myFunction() {
    var x, text;

    // Get the value of the input field with id="numb"
    x = document.getElementById("numb").value;

    // If x is Not a Number or less than one or greater than 10
    if (isNaN(x) || x < 1 || x > 10) {
        text = "Input not valid";
    } else {
        text = "Input OK";
    }
    document.getElementById("demo").innerHTML = text;
}
</script>
```

JavaScript - events

```
<some-HTML-element some-event='some JavaScript'>
```

```
<button
onclick='document.getElementById("demo").innerHTML=Date()'>
The time is?
</button>
```

```
<button onclick="displayDate()">The time is?</button>
```

JavaScript + AJAX

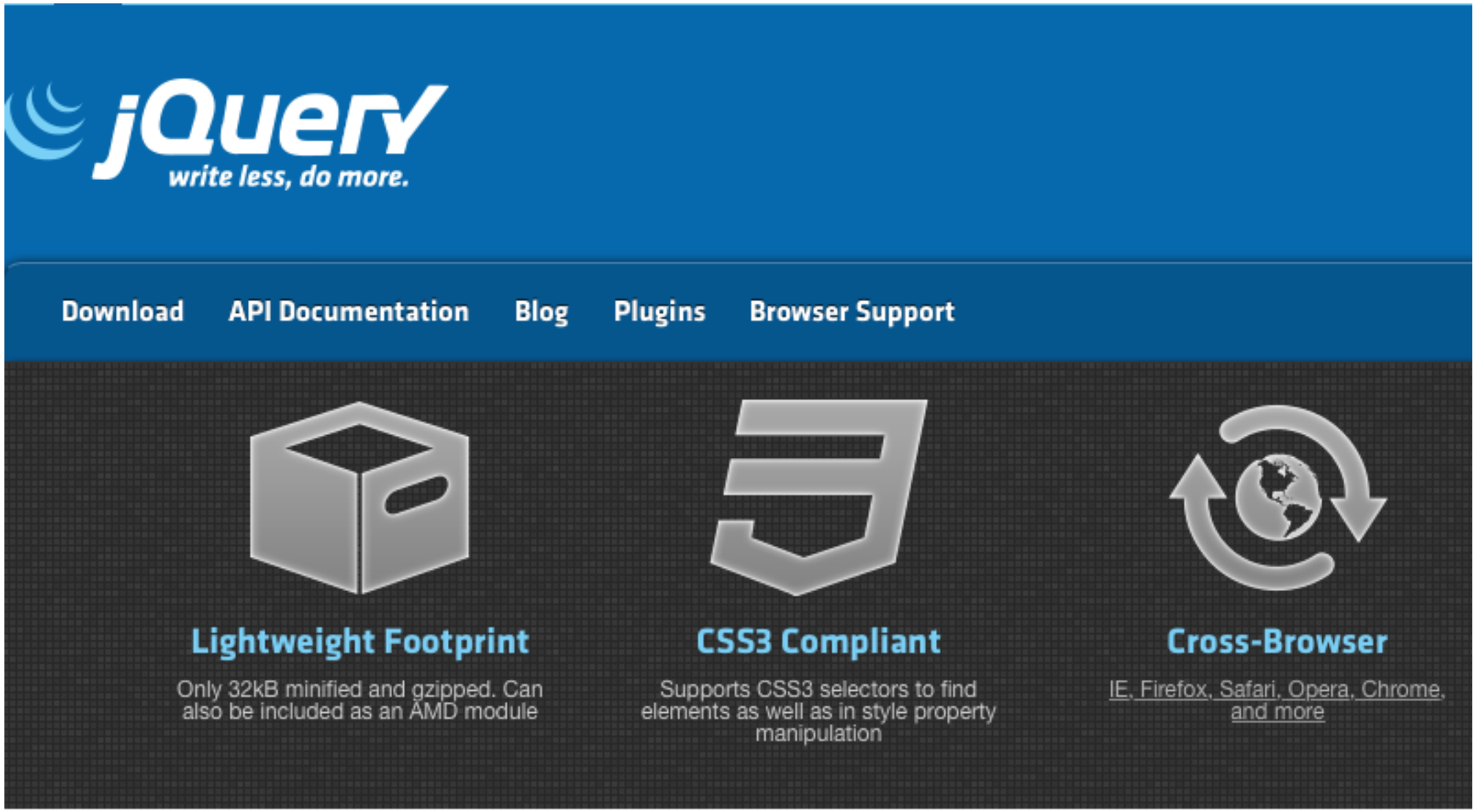
```
<div id="demo"><h2>Let AJAX change this text</h2></div>

<button type="button" onclick="loadDoc()">Change Content</button>

<script>
function loadDoc() {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (xhttp.readyState == 4 && xhttp.status == 200) {
      document.getElementById("demo").innerHTML = xhttp.responseText;
    }
  }
  xhttp.open("GET", "ajax_info.txt", true);
  xhttp.send();
}
</script>
```

JavaScript Frameworks


JavaScript Frameworks

- Abstraction of browser related details
  (No, they are not the same as the standard!).
- Higher abstraction level in browser related operations.
  - Searching the DOM and iterating elements (jQuery)
  - Updating values of elements in the DOM (React)
- Inversion of control
  - Basic event model already provide it
  - Frameworks extend it further

jQuery



What is jQuery?

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

jQuery

$(*selector*).*action*()

- A $ sign to define/access jQuery
- A (*selector*) to "query (or find)" HTML elements
- A jQuery *action*() to be performed on the element(s)

jQuery - examples

$(this).hide() - hides the current element.

$("p").hide() - hides all <p> elements.

$(".test").hide() - hides all elements with class=“test".

$("#test").hide() - hides the element with id="test".

```
$(document).ready(function(){
    // jQuery methods go here...
});
```

jQuery - selectors

Syntax	Description
$("*")	Selects all elements
$(this)	Selects the current HTML element
$("p.intro")	Selects all <p> elements with class="intro"
$("p:first")	Selects the first <p> element
$("ul li:first")	Selects the first element of the first
$("ul li:first-child")	Selects the first element of every
$("[href]")	Selects all elements with an href attribute
$("a[target='_blank']")	Selects all <a> elements with a target attribute value equal to "_blank"
$("a[target!='_blank']")	Selects all <a> elements with a target attribute value NOT equal to "_blank"
$(":button")	Selects all <button> elements and <input> elements of type="button"
$("tr:even")	Selects all even <tr> elements
$("tr:odd")	Selects all odd <tr> elements

jQuery - events

```
$("p").click(function(){
    $(this).hide();
});
```

```
$("#p1").hover(function(){
    alert("You entered p1!");
},
function(){
    alert("Bye! You now leave p1!");
});
```

jQuery - AJAX

```
$.ajax({
  method: "POST",
  url: "some.php",
  data: { name: "John", location: "Boston" }
})
  .done(function( msg ) {
    alert( "Data Saved: " + msg );
  });
```

```
$.ajax({
  url: "test.html",
  cache: false
})
  .done(function( html ) {
    $( "#results" ).append( html );
  });
```