# Multicolumn Page Layout

Web Development

# Agenda

- The DOM, HTML Parsing & Rendering
- Formatting concepts in CSS
- A simple, worked example of multicolumn layout
- A worked example of a “tabbed” multicolumn layout

# Parsing & Rendering

- A web browser typically reads and renders HTML documents in two phases:
    - the parsing phase
    - the rendering phase.

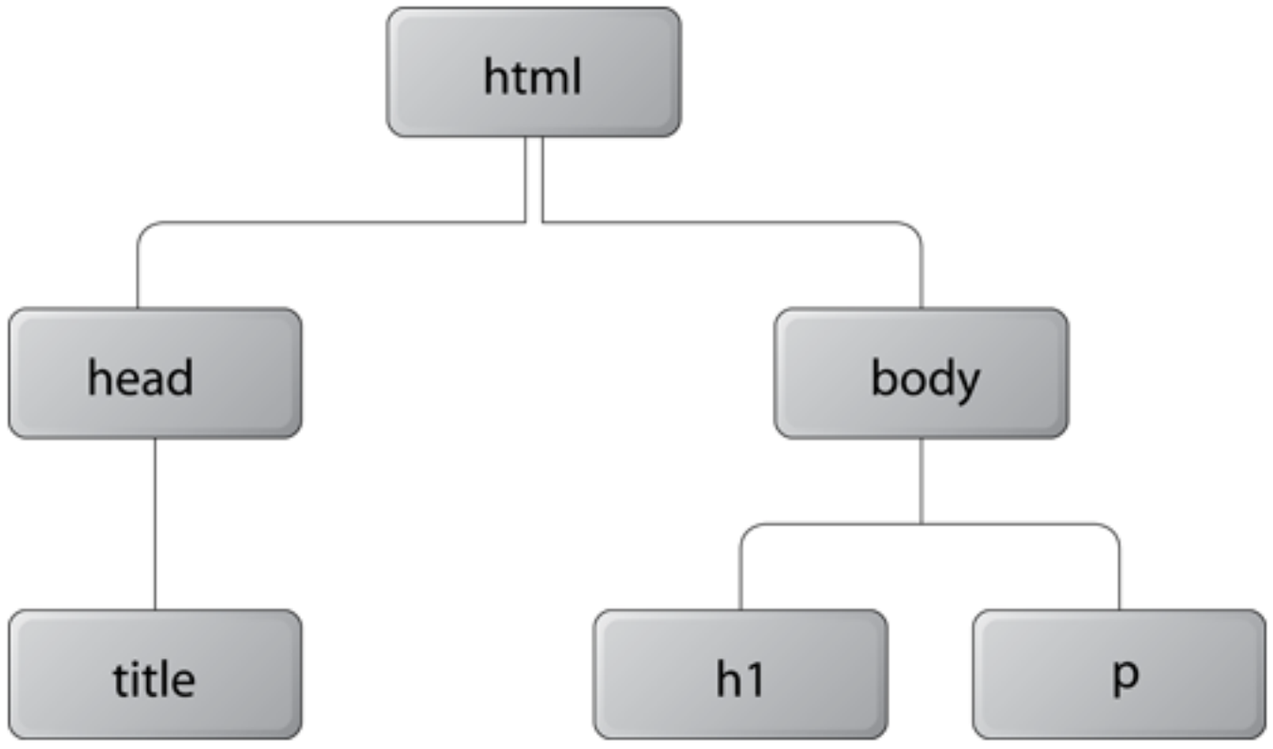
- During the parsing phase, the browser reads the markup in the document, breaks it down into components, and builds a Document Object Model (DOM) tree.

- The DOM is a in-memory data structure, typically traversed by Java Script code

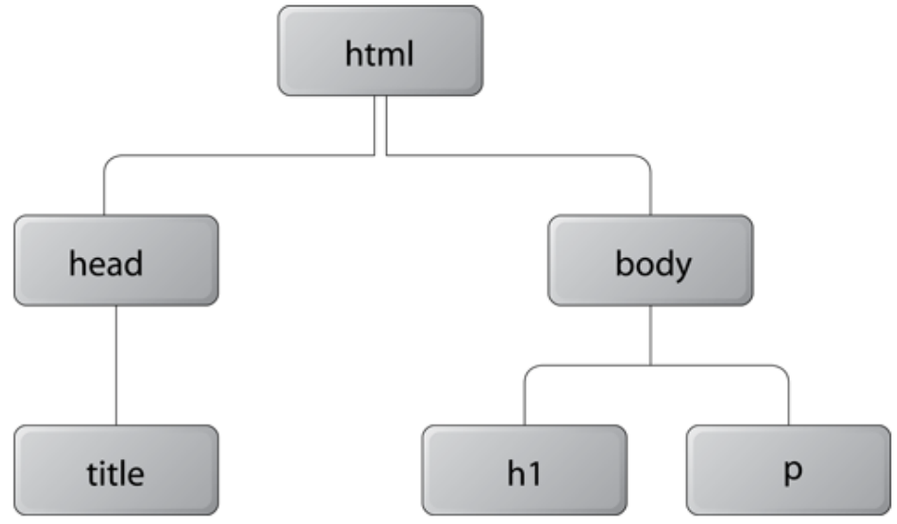
# DOM Tree

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <title>Widgets</title>
  </head>
  <body>
    <h1>Widgets</h1>
    <p>Welcome to Widgets, the number one company
    in the world for selling widgets!</p>
  </body>
</html>
```

- Each object in the DOM tree is called a node.
- There are several types of nodes, including element nodes and text nodes.
- At the top of the tree is a document node, which contains an element node called the root node; this is always the html element in HTML documents.
- It branches into two child element nodes—head and body—which then branch into other children.



# Parents, Grandparents, Children, Grandchildren




## Siblings

- A child node is structurally subordinate to its parent node - the child's tags are nested inside the tags of the parent.

- A node can be called a descendant node if it's a child, grandchild, and so on, of another node.

- A node can be called an ancestor node if it's a parent, grandparent, and so on, of another node.

- Nodes that have the same parent are called siblings. .

# Block & Inline

- When the DOM tree has been constructed, and any CSS style sheets have been loaded and parsed, the browser starts the rendering phase.
- Each node in the DOM tree will be rendered as zero or more boxes.
- A  box is always rectangular, it has four sides with a 90° angle between each side.

- HTML Elements are either block-level or inline-level.
  - block-level HTML elements generate block boxes,
  - while inline-level HTML elements generate inline boxes.
- The separation into block-level and inline elements in HTML is specified in the HTML document type definition

# CSS Box Model

- The box model is the central mechanism for determining how an element is positioned on a web page.

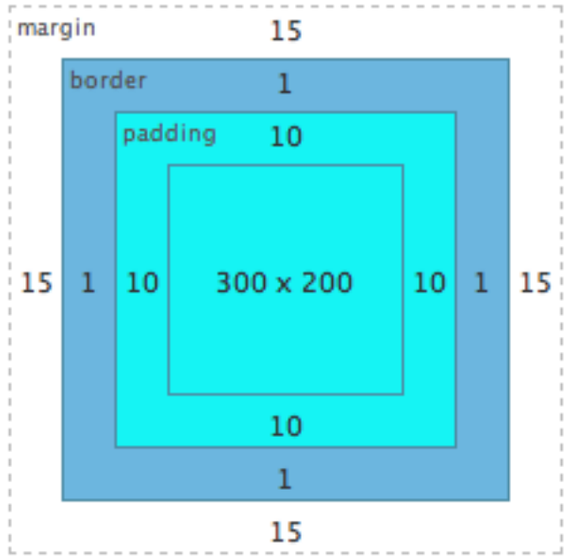
- Some key concepts:
  - Box Structure
  - Calculating Box Dimensions
  - The Containing Block
  - Collapsing Margins

# Box Structure

- The content area in the center, the padding around the content area, the border area, and the margin area.

- The outer edge of the content area is called the content edge or inner edge; the outer edge of the padding area is called the padding edge; the outer edge of the border area is called the border edge; and the outer edge of the margin area is called the margin edge or outer edge

```
.box
{
  width: 300px;
  height: 200px;
  padding: 10px;
  border: 1px solid #000;
  margin: 15px;
}
```



# Calculating Box Dimensions

- Block-level elements can only be rectangular.

- Calculate the overall dimensions of a block-level element by taking into account the height and width of the content area, as well as any margins, padding, and borders that are applied to the element.

- Define the content width of an element by declaring its width and height properties. If no declarations are applied, the default value for the width and height properties is auto.

# The Containing Block

- CSS rendering comprises the tasks of laying out and rendering numerous boxes. Element boxes are positioned within a formatting context, which, by default, is provided by the box generated by a parent element.

- When we specify the positions or dimensions of element boxes, we're doing so relative to what's known as the containing block, which is a very important concept in CSS layout.

- The containing block for the root element is called the initial containing block, and has the same dimensions as the viewport - generally the browser window

- The containing block for other elements is generally the immediate parent

# Collapsing Margins

- When the vertical margins of two elements are touching, only the margin of the element with the largest margin value will be honored, while the margin of the element with the smaller margin value will be collapsed to zero

- In this example, the gap between the elements is only 25px, and the smaller margin has collapsed to zero

- Exceptions to this rule:
  - floated elements
  - absolutely positioned elements
  - 4 other special cases....

```
h1
{
  margin: 0 0 25px 0;
  background: #cfc;
}
p
{
  margin: 20px 0 0 0;
  background: #cf9;
}
```

# Formatting Concepts

- Layout Rules
  - Block Formatting
  - Inline Formatting
  - List Formatting
  - Table Formatting
  - Replaced Elements

- Positioning Rules
- Relative Positioning
- Absolute Positioning
- Fixed Positioning
- Stacking Contexts
- Floating & Clearing

http://reference.sitepoint.com/css/formattingcontext

http://reference.sitepoint.com/css/positioning

# References (1) https://alistapart.com/article/holygrail

Issue № 211

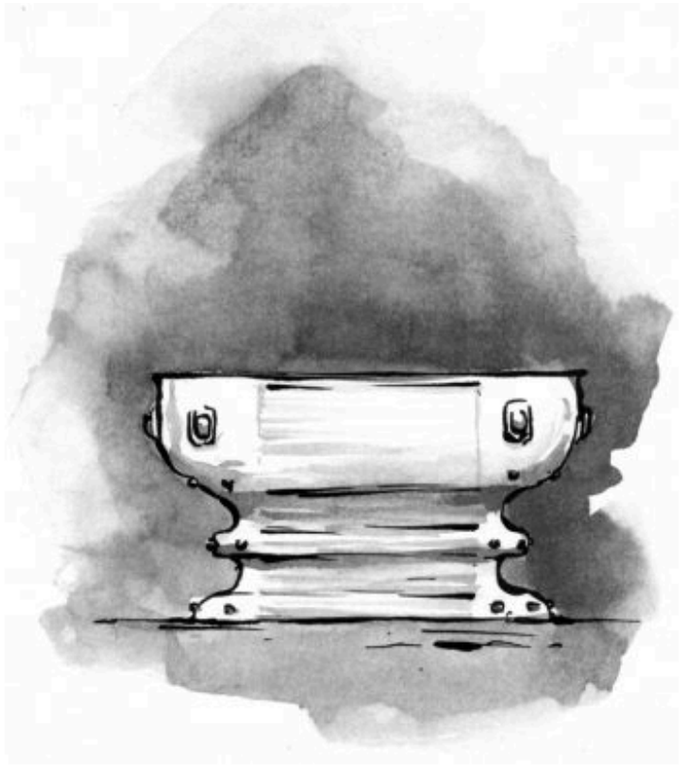


Illustration by Kevin Cornell

## In Search of the Holy Grail

by **Matthew Levine** · January 30, 2006

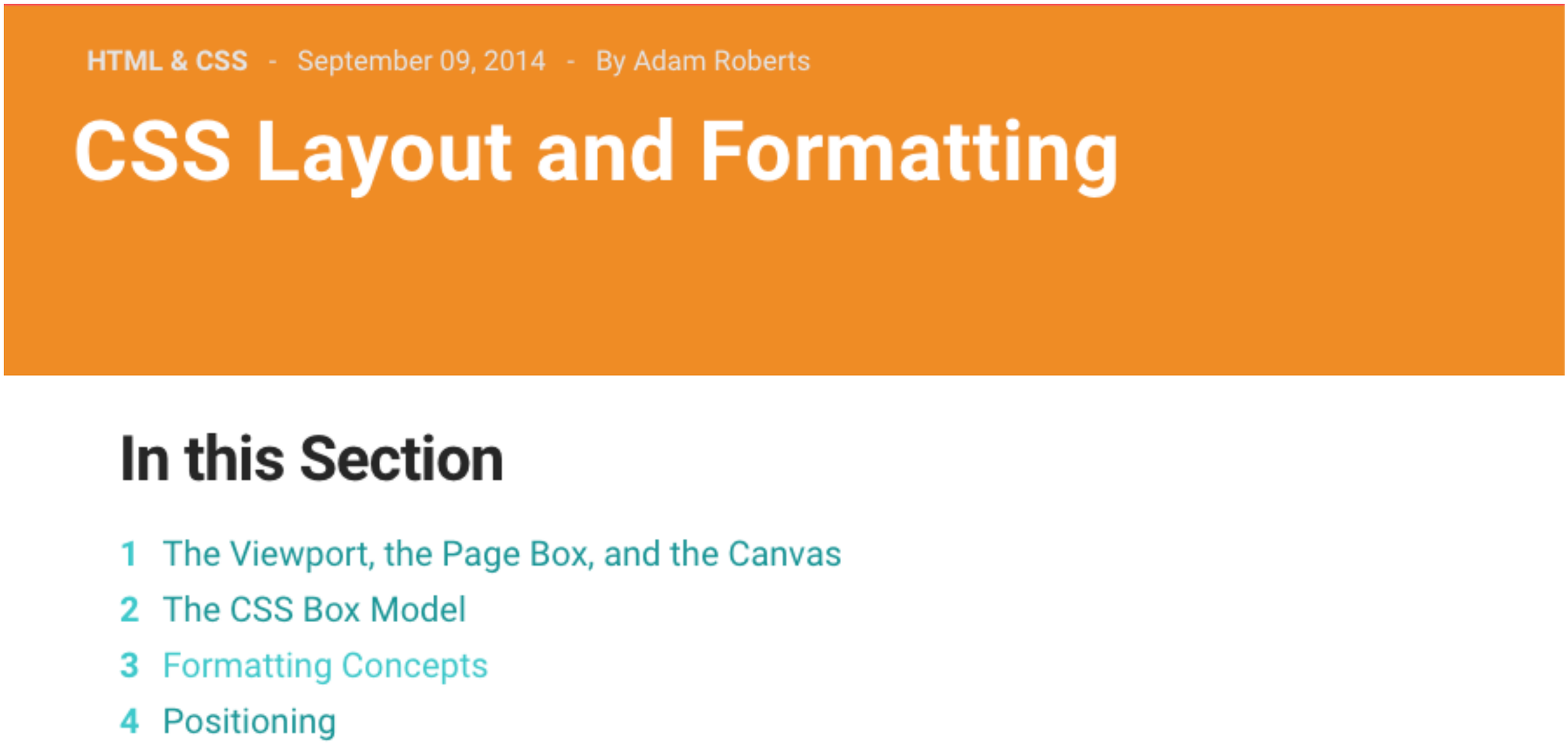
Published in CSS, HTML, Layout & Grids, Interaction Design

I'm sorry. Really. I didn't name it. I don't mean to overstate its importance or trivialize the other and rather weightier Holy Grails.

> But the name's out there, and we all know what it means.
>
> Three columns. One fixed-width sidebar for your navigation, another for, say, your Google Ads or your Flickr photos—and, as in a fancy truffle, a liquid center for the real substance. Its wide applicability in this golden age of blogging, along with its considerable difficulty, is what has earned the layout the title of Holy Grail.

References (2) https://www.sitepoint.com/css-layout-formatting/

---



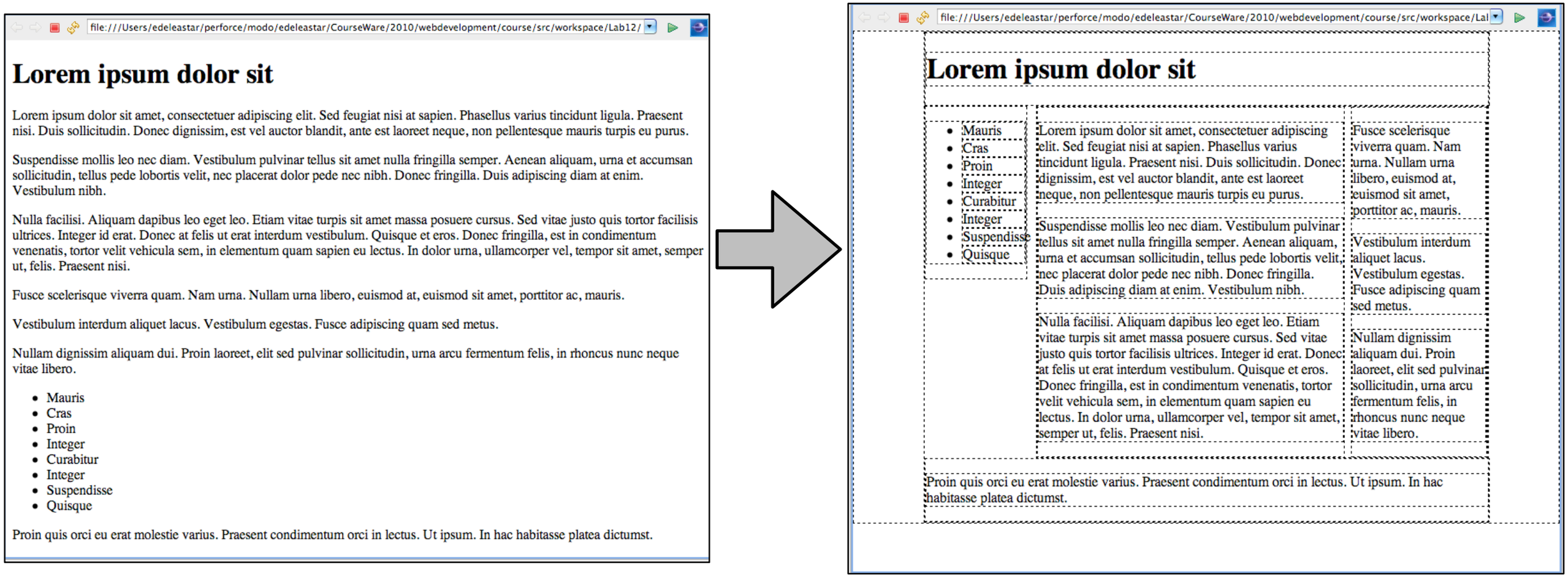
HTML & CSS - September 09, 2014 - By Adam Roberts

# CSS Layout and Formatting

## In this Section

1 The Viewport, the Page Box, and the Canvas
2 The CSS Box Model
3 Formatting Concepts
4 Positioning

# Multi-Column Layout - A Worked Example



# Content

- 1 <h1>
- 6 <p>’s
- 1 <ul>
- 1 <p>

```
...
  <body>
    <h1>
      Lorem ipsum dolor sit
    </h1>
    <p>
      Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Sed feugiat nisi
      at sapien. Phasellus varius tincidunt ligula. Praesent nisi. Duis
      sollicitudin. Donec dignissim, est vel auctor blandit, ante est laoreet
      neque, non pellentesque mauris turpis eu purus.
    </p>
    <p>
      Suspendisse mollis leo nec diam. Vestibulum pulvinar tellus sit amet nulla
      fringilla semper. Aenean aliquam, urna et accumsan sollicitudin, tellus
      pede lobortis velit, nec placerat dolor pede nec nibh. Donec fringilla. Duis
      adipiscing diam at enim. Vestibulum nibh.
    </p>
    <p>
      Nulla facilisi. Aliquam dapibus leo eget leo. Etiam vitae turpis sit amet
      massa posuere cursus. Sed vitae justo quis tortor facilisis ultrices.
      Integer id erat. Donec at felis ut erat interdum vestibulum. Quisque et eros.
      Donec fringilla, est in condimentum venenatis, tortor velit vehicula sem, in
      elementum quam sapien eu lectus. In dolor urna, ullamcorper vel, tempor sit
      amet, semper ut, felis. Praesent nisi.
    </p>
    <p>
      Fusce scelerisque viverra quam. Nam urna. Nullam urna libero, euismod at,
      euismod sit amet, porttitor ac, mauris.
    </p>
    <p>
      Vestibulum interdum aliquet lacus. Vestibulum egestas. Fusce adipiscing
      quam sed metus.
    </p>
    <p>
      Nullam dignissim aliquam dui. Proin laoreet, elit sed pulvinar
      sollicitudin, urna arcu fermentum felis, in rhoncus nunc neque vitae
      libero.
    </p>
    <ul>
      <li>Mauris</li>
      <li>Cras</li>
      <li>Proin</li>
      <li>Integer</li>
      <li>Curabitur</li>
      <li>Integer</li>
      <li>Suspendisse</li>
      <li>Quisque</li>
    </ul>
    <p>
      Proin quis orci eu erat molestie varius. Praesent condimentum
      orci in lectus. Ut ipsum. In hac habitasse platea dictumst.
    </p>
  </body>
...
```

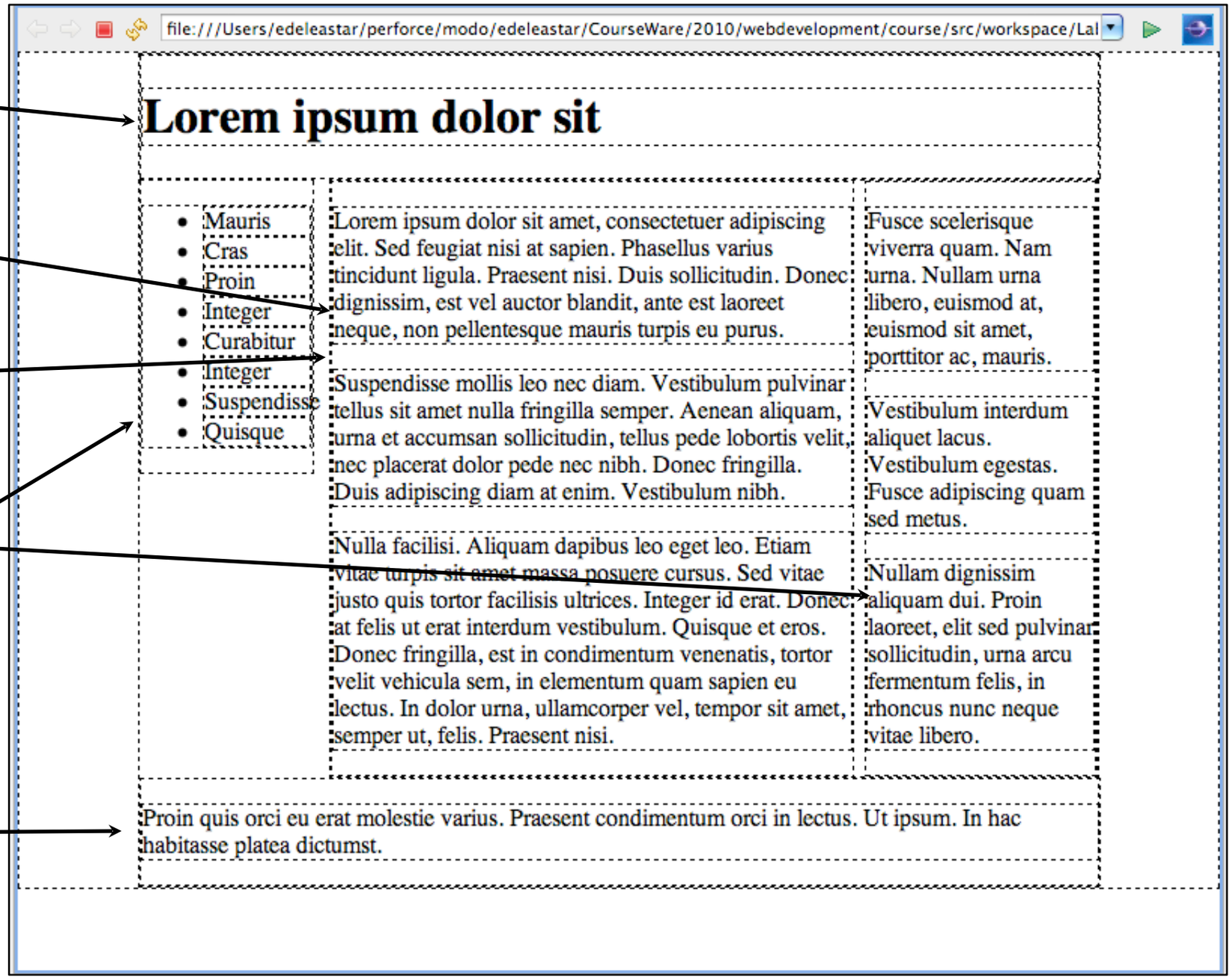
# Content

- header
- maincontent
  - primary
  - second
- navigation
- footer

```
...
  <body>
    <h1>
      Lorem ipsum dolor sit
    </h1>
    <p>
      Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Sed feugiat nisi
      at sapien. Phasellus varius tincidunt ligula. Praesent nisi. Duis
      sollicitudin. Donec dignissim, est vel auctor blandit, ante est laoreet
      neque, non pellentesque mauris turpis eu purus.
    </p>
    <p>
      Suspendisse mollis leo nec diam. Vestibulum pulvinar tellus sit amet nulla
      fringilla semper. Aenean aliquam, urna et accumsan sollicitudin, tellus
      pede lobortis velit, nec placerat dolor pede nec nibh. Donec fringilla. Duis
      adipiscing diam at enim. Vestibulum nibh.
    </p>
    <p>
      Nulla facilisi. Aliquam dapibus leo eget leo. Etiam vitae turpis sit amet
      massa posuere cursus. Sed vitae justo quis tortor facilisis ultrices.
      Integer id erat. Donec at felis ut erat interdum vestibulum. Quisque et eros.
      Donec fringilla, est in condimentum venenatis, tortor velit vehicula sem, in
      elementum quam sapien eu lectus. In dolor urna, ullamcorper vel, tempor sit
      amet, semper ut, felis. Praesent nisi.
    </p>
    <p>
      Fusce scelerisque viverra quam. Nam urna. Nullam urna libero, euismod at,
      euismod sit amet, porttitor ac, mauris.
    </p>
    <p>
      Vestibulum interdum aliquet lacus. Vestibulum egestas. Fusce adipiscing
      quam sed metus.
    </p>
    <p>
      Nullam dignissim aliquam dui. Proin laoreet, elit sed pulvinar
      sollicitudin, urna arcu fermentum felis, in rhoncus nunc neque vitae
      libero.
    </p>
    <ul>
      <li>Mauris</li>
      <li>Cras</li>
      <li>Proin</li>
      <li>Integer</li>
      <li>Curabitur</li>
      <li>Integer</li>
      <li>Suspendisse</li>
      <li>Quisque</li>
    </ul>
    <p>
      Proin quis orci eu erat molestie varius. Praesent condimentum
      orci in lectus. Ut ipsum. In hac habitasse platea dictumst.
    </p>
  </body>
...
```

# Target

- header
- maincontent
  - primary
  - secondary
- navigation
- footer



# DIVs (1)

```
<div id="header">
  <h1>
    Lorem ipsum dolor sit
  </h1>
</div>
```

- header
- maincontent
  - primary
  - second

```
<div id="maincontent">
  <div id="primary">
    <p>
      Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Sed feugiat nisi
      at sapien. Phasellus varius tincidunt ligula. Praesent nisi. Duis
      sollicitudin. Donec dignissim, est vel auctor blandit, ante est laoreet
      neque, non pellentesque mauris turpis eu purus.
    </p>
    <p>
      Suspendisse mollis leo nec diam. Vestibulum pulvinar tellus sit amet nulla
      fringilla semper. Aenean aliquam, urna et accumsan sollicitudin, tellus
      pede lobortis velit, nec placerat dolor pede nec nibh. Donec fringilla. Duis
      adipiscing diam at enim. Vestibulum nibh.
    </p>
    <p>
      Nulla facilisi. Aliquam dapibus leo eget leo. Etiam vitae turpis sit amet
      massa posuere cursus. Sed vitae justo quis tortor facilisis ultrices.
      Integer id erat. Donec at felis ut erat interdum vestibulum. Quisque et eros.
      Donec fringilla, est in condimentum venenatis, tortor velit vehicula sem, in
      elementum quam sapien eu lectus. In dolor urna, ullamcorper vel, tempor sit
      amet, semper ut, felis. Praesent nisi.
    </p>
  </div>
  <div id="secondary">
    <p>
      Fusce scelerisque viverra quam. Nam urna. Nullam urna libero, euismod at,
      euismod sit amet, porttitor ac, mauris.
    </p>
    <p>
      Vestibulum interdum aliquet lacus. Vestibulum egestas. Fusce adipiscing
      quam sed metus.
    </p>
    <p>
      Nullam dignissim aliquam dui. Proin laoreet, elit sed pulvinar
      sollicitudin, urna arcu fermentum felis, in rhoncus nunc neque vitae
      libero.
    </p>
  </div>
</div>
```

# Divs (2)

- navigation

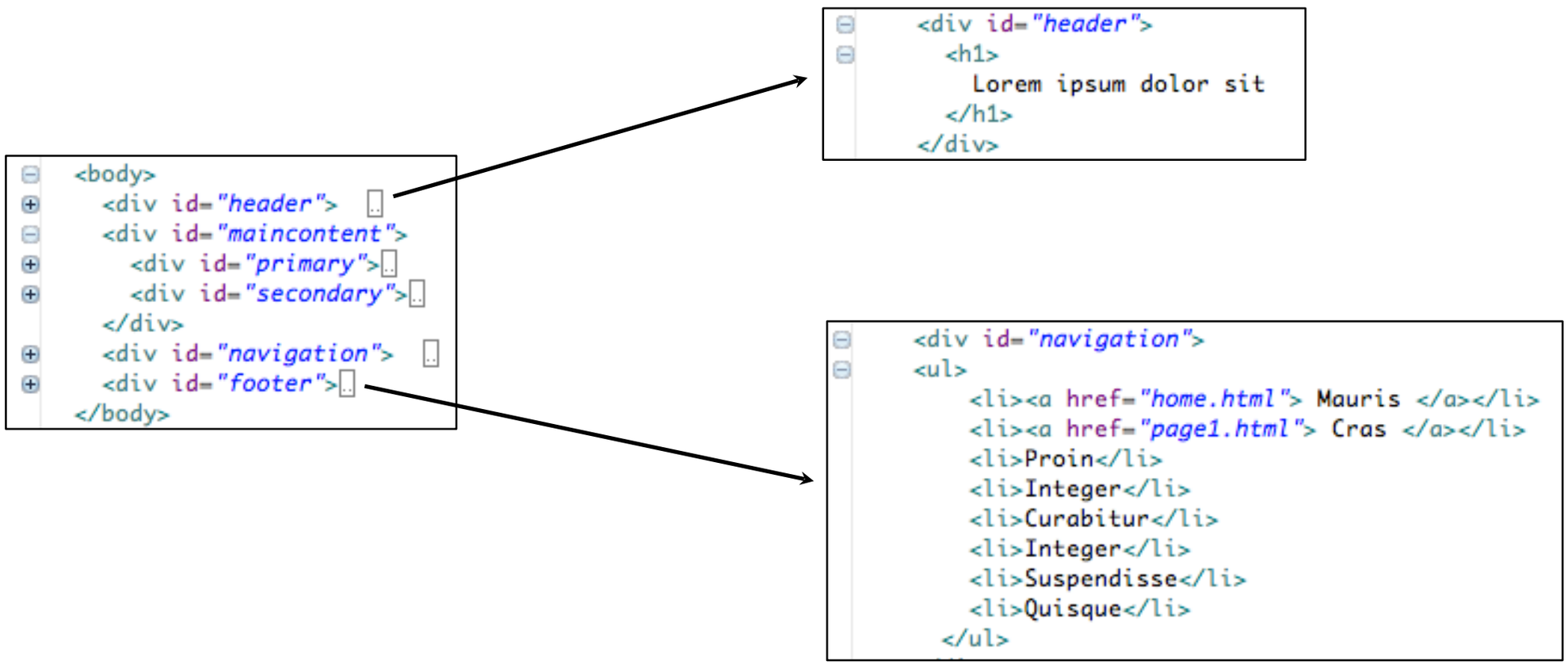
```
<div id="navigation">
  <ul>
    <li><a href="home.html"> Mauris </a></li>
    <li><a href="page1.html"> Cras </a></li>
    <li>Proin</li>
    <li>Integer</li>
    <li>Curabitur</li>
    <li>Integer</li>
    <li>Suspendisse</li>
    <li>Quisque</li>
  </ul>
</div>
```

- footer

```
<div id="footer">
  <p>
    Proin quis orci eu erat molestie varius. Praesent condimentum
    orci in lectus. Ut ipsum. In hac habitasse platea dictumst.
  </p>
</div>
```

# Sublime Folding View

- Eclipse Folding editor can “Expand all” or “Collapse all” of the elements, facilitating a precise focus on the elements under consideration.



# Stylesheet

```
<link type="text/css" rel="stylesheet" href="home.css" media="screen" />
```

- Width of body to be 80% of “containing element” (the browser window)

```
body
{
  width: 80%;
  margin: 0 auto;
}
```

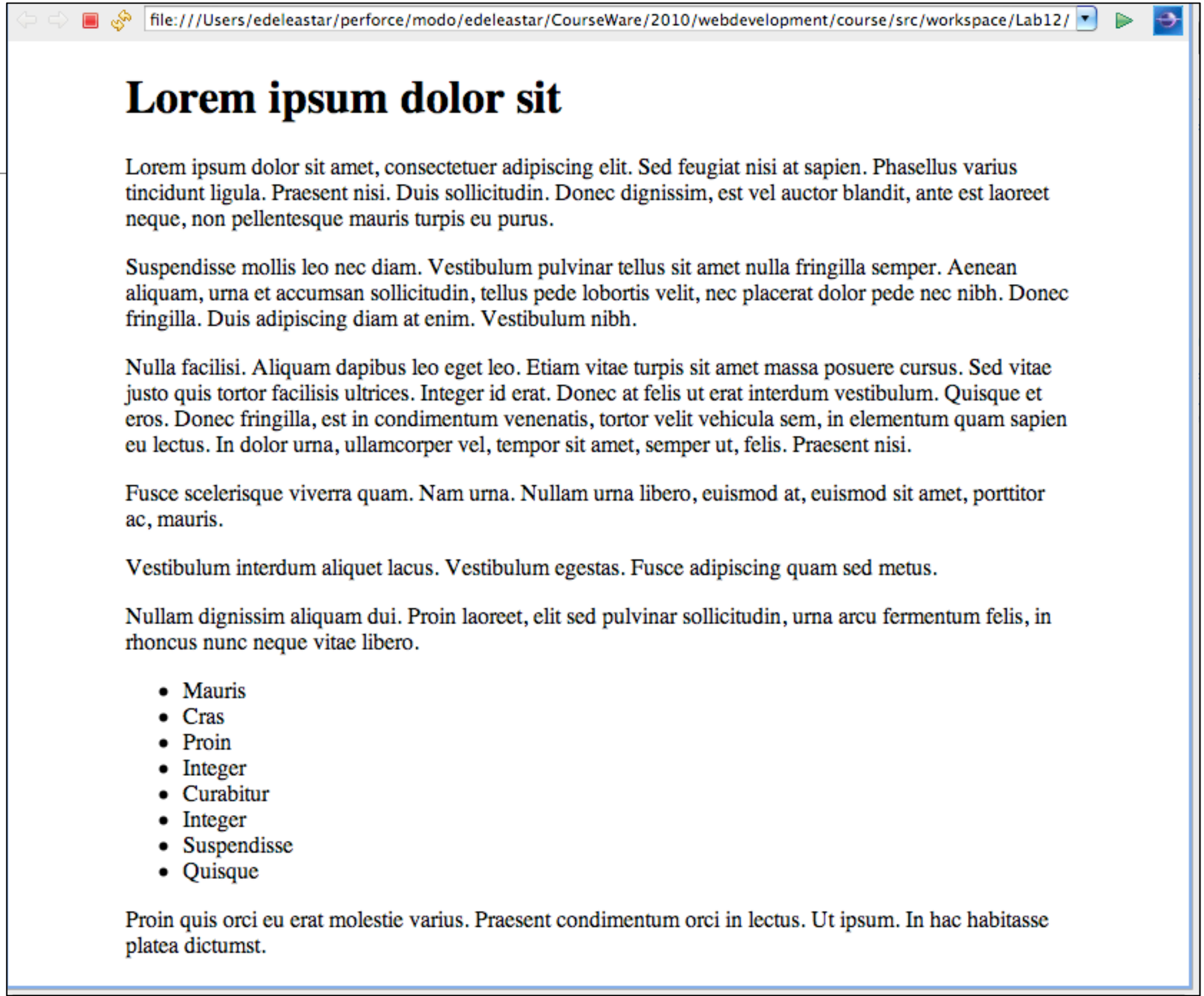
- No top/bottom margin for the page
- “auto” for left and right
  - i.e. the margins to be proportional to the container width and equal in size.

shortcut for...

```
body
{
  width: 80%;
  margin-top: 0;
  margin-bottom: 0;
  margin-left: auto;
  margin-right: auto;
}
```

```css
body
{
  width: 80%;
  margin: 0 auto;
}
```

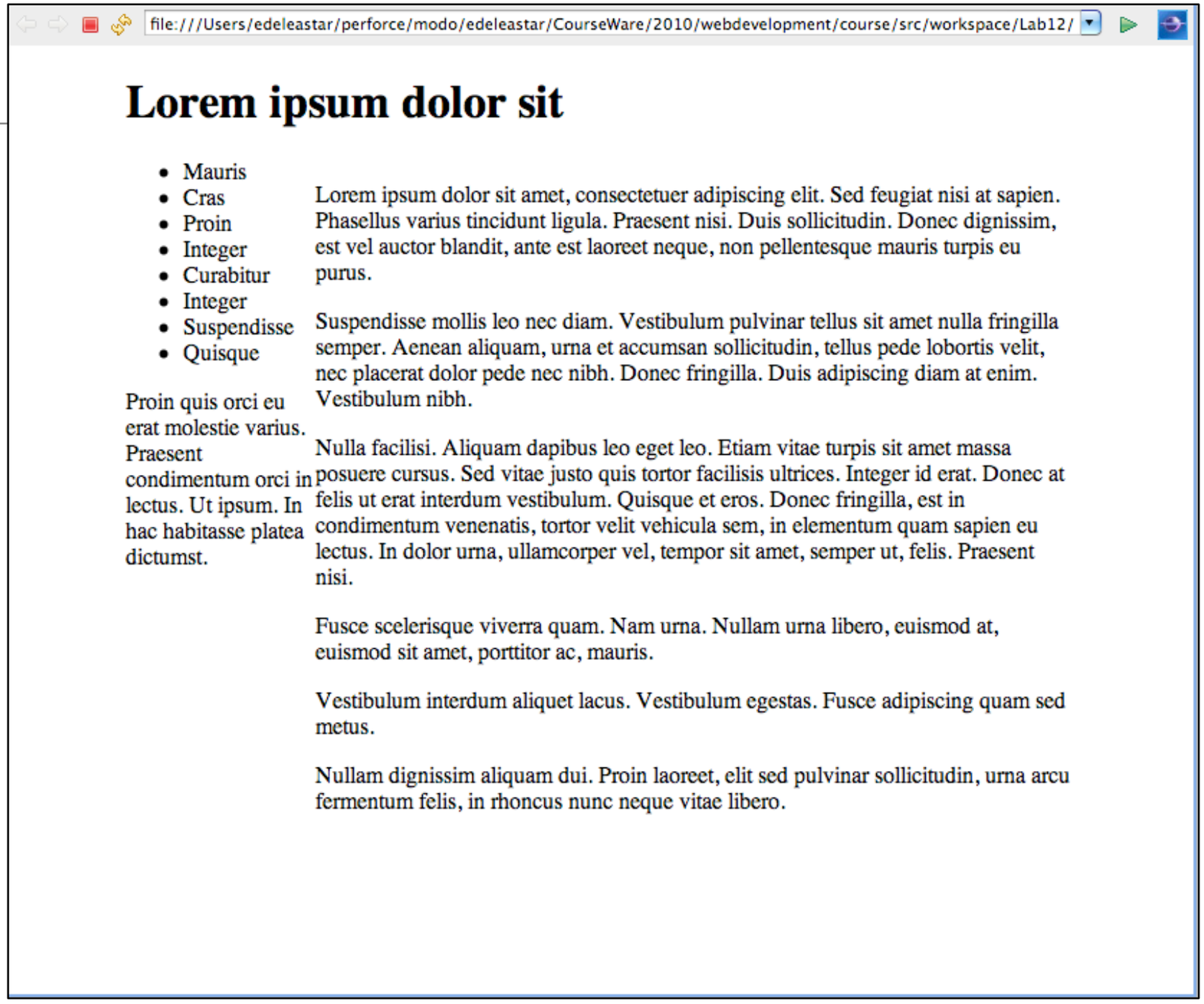
- Entire page is positioned centered within the browser



# Float

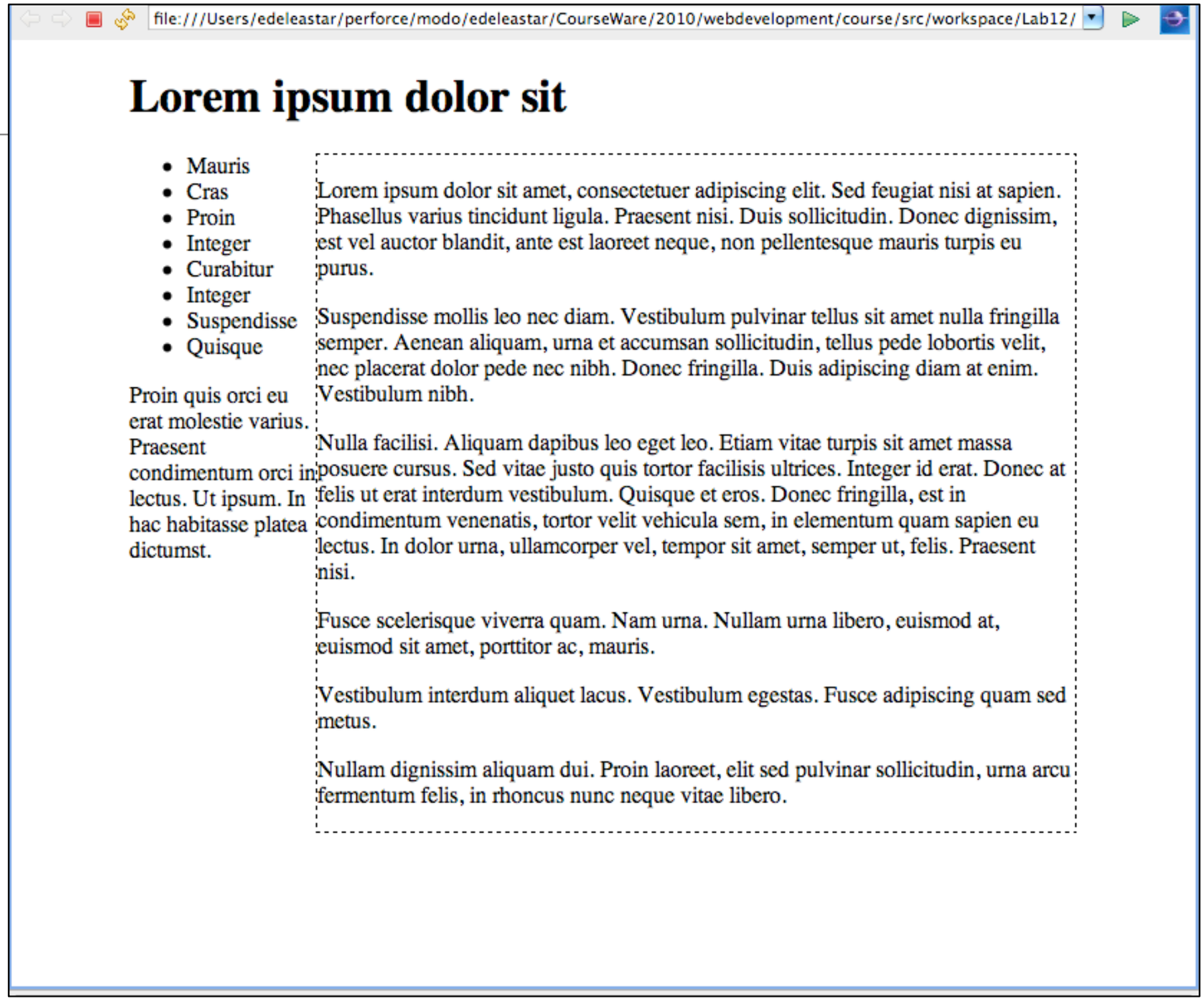
```
#maincontent
{
  width: 80%;
  float: right;
}
```

- Width of maincontent is to be 80% of it's container , which in turn is 80% of its container.
- Maincontent is a "floated" box shifted to the right until its margin edge touches the padding edge of the containing block (or the margin edge of another floated element).



# Show Borders (temporarily)

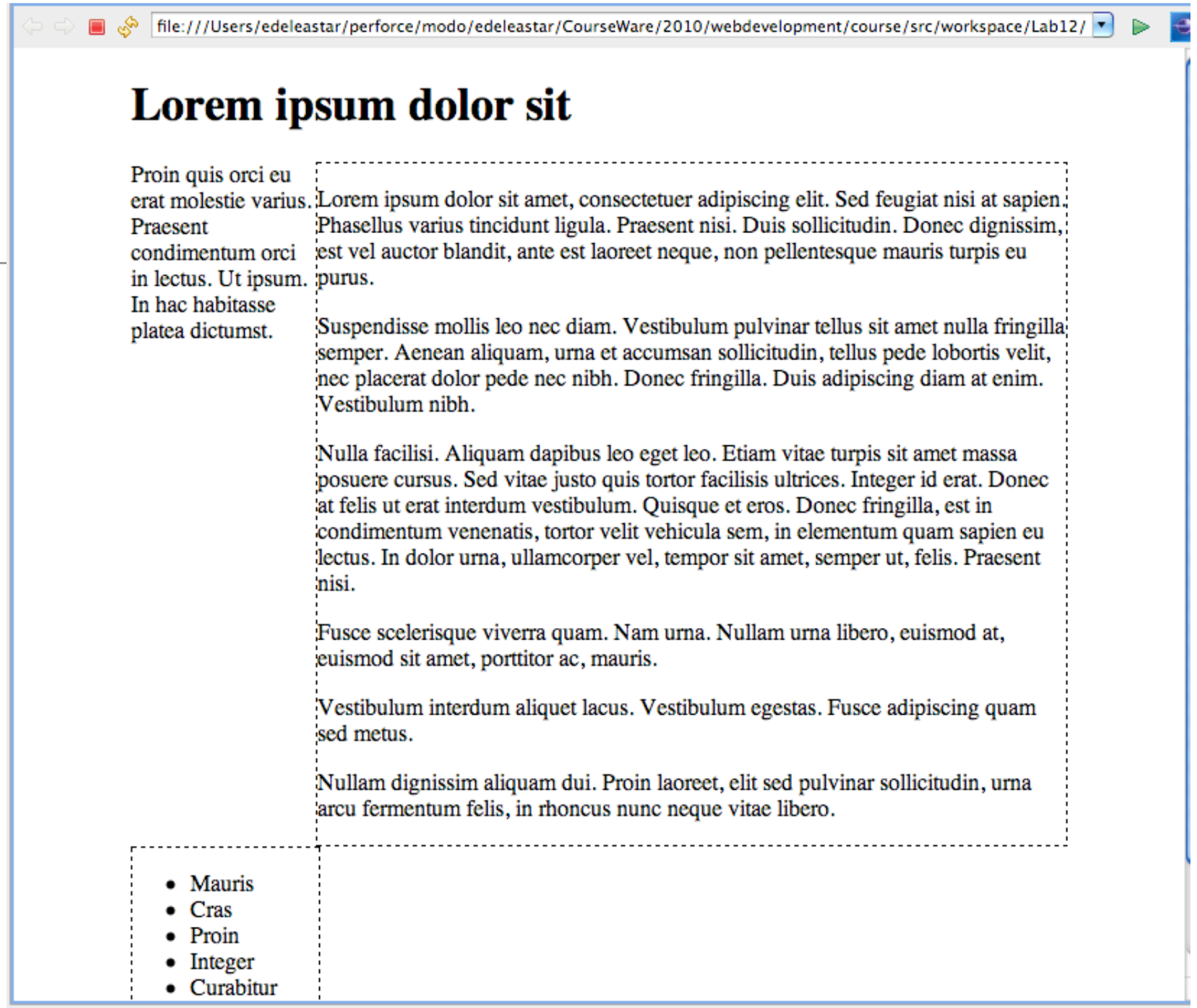
```
#maincontent
{
  width: 80%;
  float: right;
  border: dashed thin;
}
```




- Show a dashed border around the maincontent to better understand the layout behaviour

# Navigation - take 1

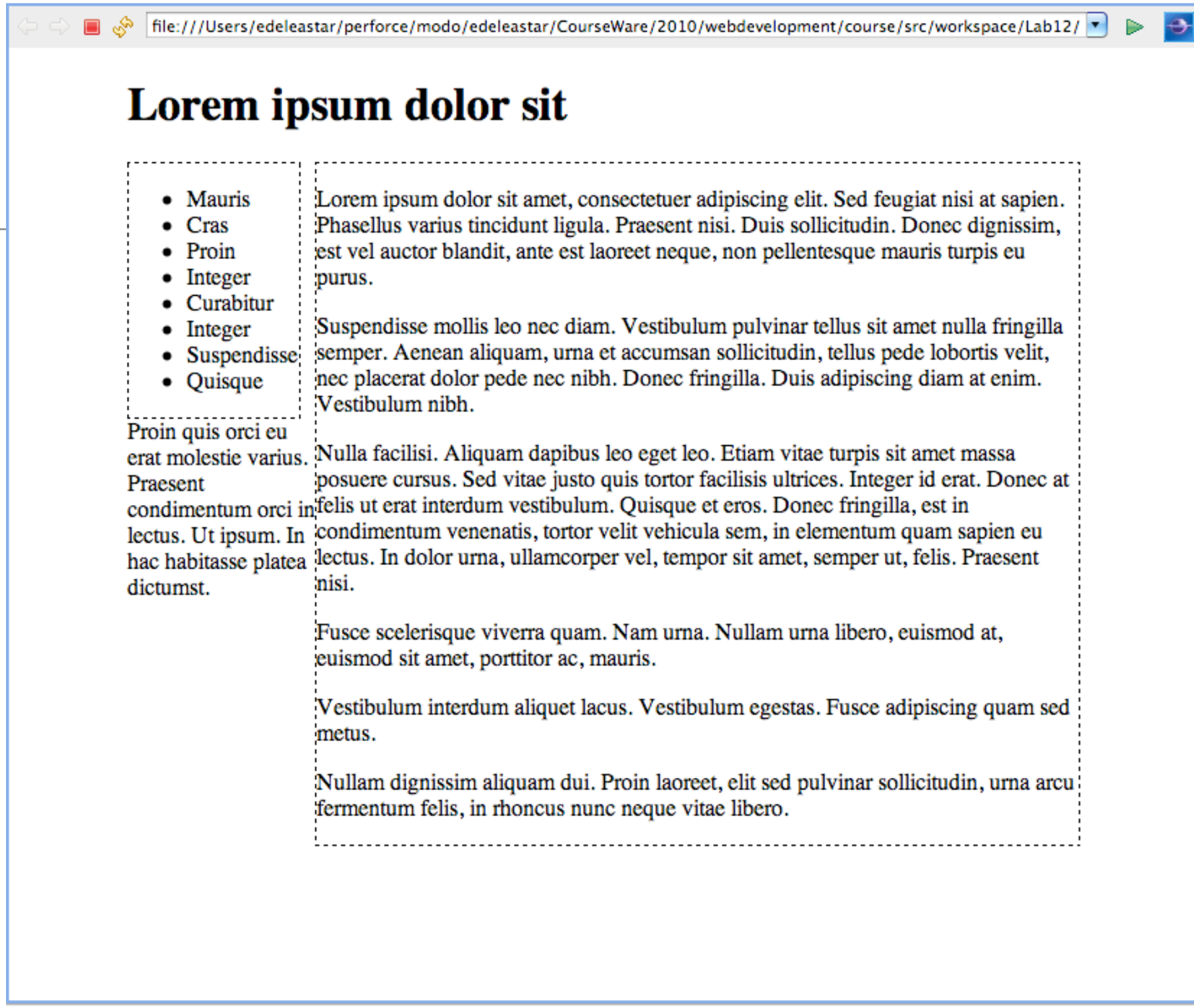
```
#navigation
{
  width: 20%;
  float: left;
  border: dashed thin;
}
```




- Resize the navigation to be 20% of container
- Float left
- However, as width does not include padding & margins, this is too wide to fit on “row” with maincontent, so “flows” below it

# Navigation - take 2

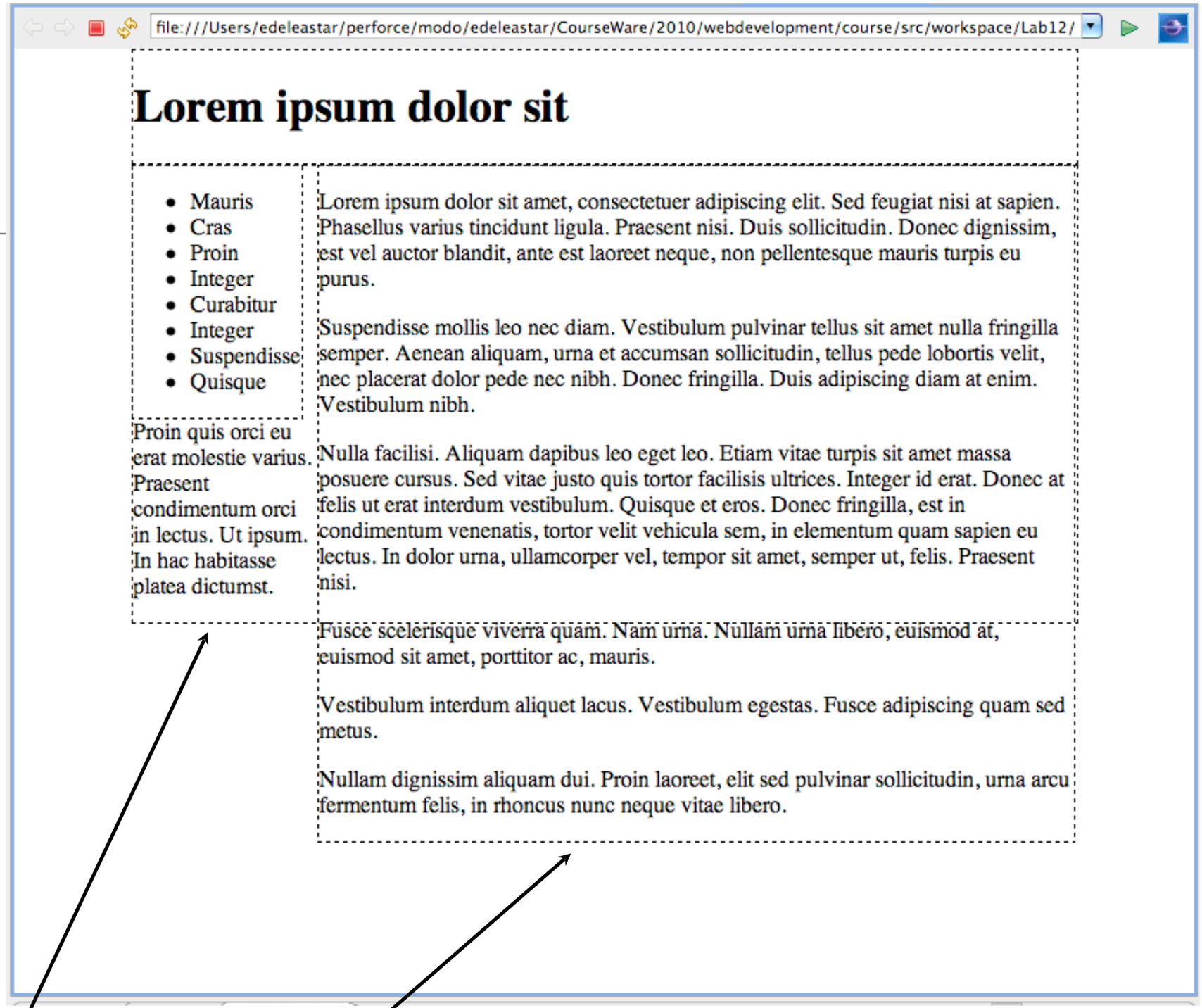
```
#navigation
{
  width: 18%;
  float: left;
  border: dashed thin;
}
```




- Reset width to 18% - now enough room for navigation & maincontent

# Header & Footer

```
#header
{
  width: 100%;
  border: dashed thin;
}

#footer
{
  width: 100%;
  border: dashed thin;
}
```




- Header & Footer to be 100% of the container
- Anomalies arise - footer (the text on the left) seem misplaced.
- These borders seem confused

# Clear

- “clear” forces an element to start below any floated elements

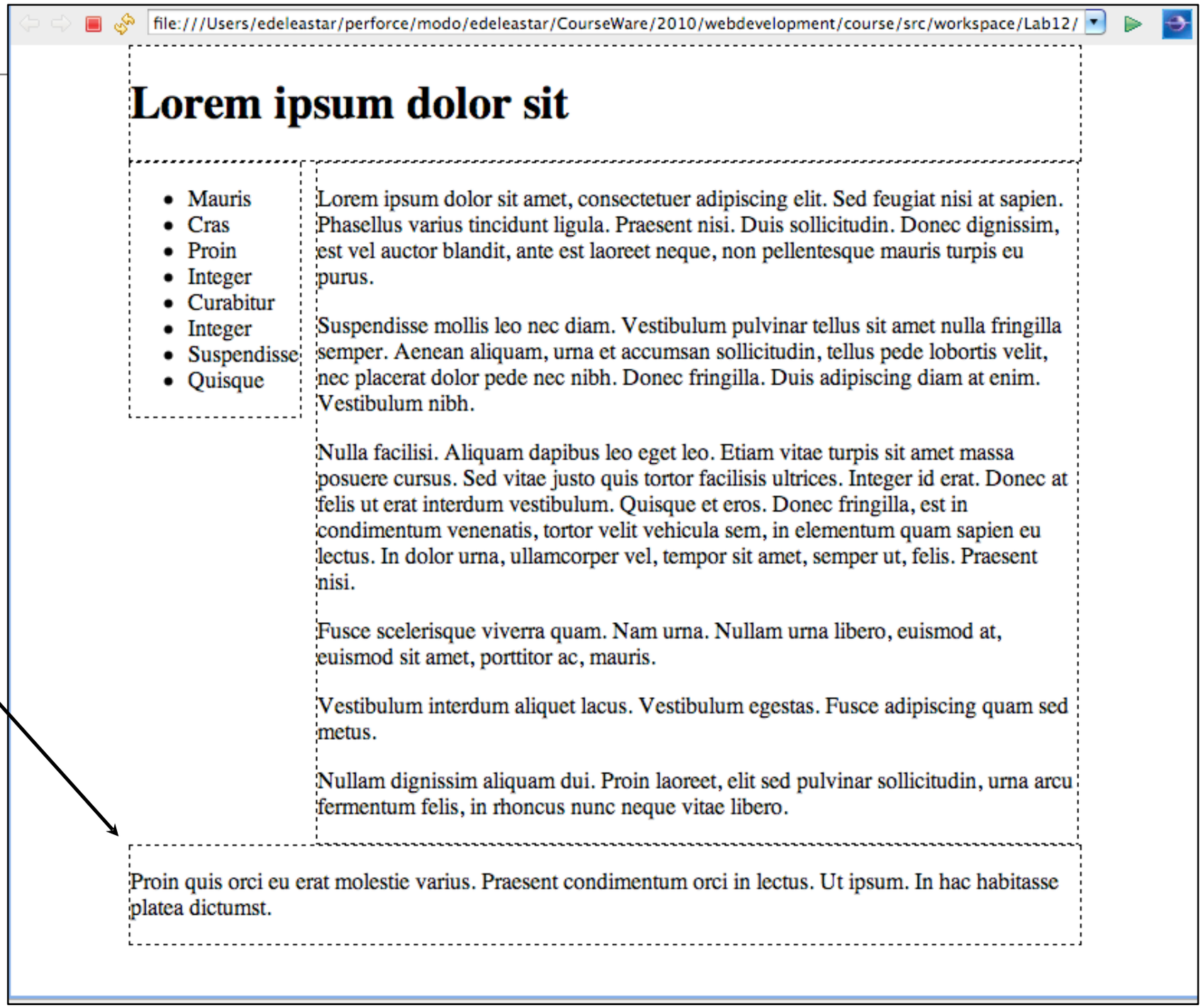
- An element whose clear property is set to “left” will start below all left-floated boxes in the same block formatting context

- A clear value of “right” will clear all right-floated boxes.

- If clear is set to “both”, the element will start below any floated box in that context

# Footer:clear

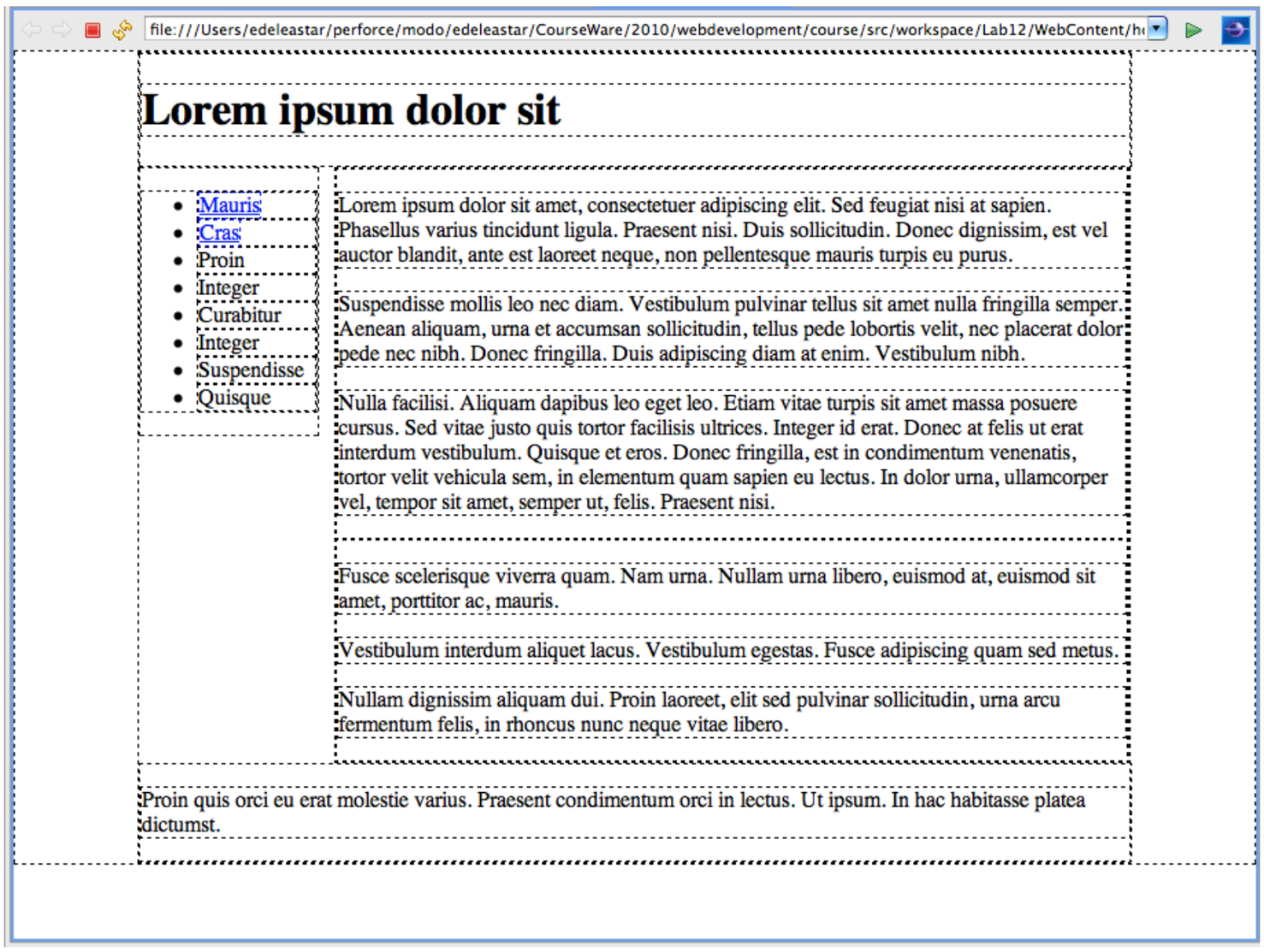
```
#footer
{
  width: 100%;
  clear: both;
  border: dashed thin;
}
```



# Pattern Matching Selectors

```css
body
{
  width: 80%;
  margin: 0 auto;
}

#maincontent
{
  width: 80%;
  float: right;
  border: dashed thin;
}

#navigation
{
  width: 18%;
  float: left;
  border: dashed thin;
}

#header
{
  width: 100%;
  border: dashed thin;
}

#footer
{
  width: 100%;
  clear: both;
  border: dashed thin;
}
```

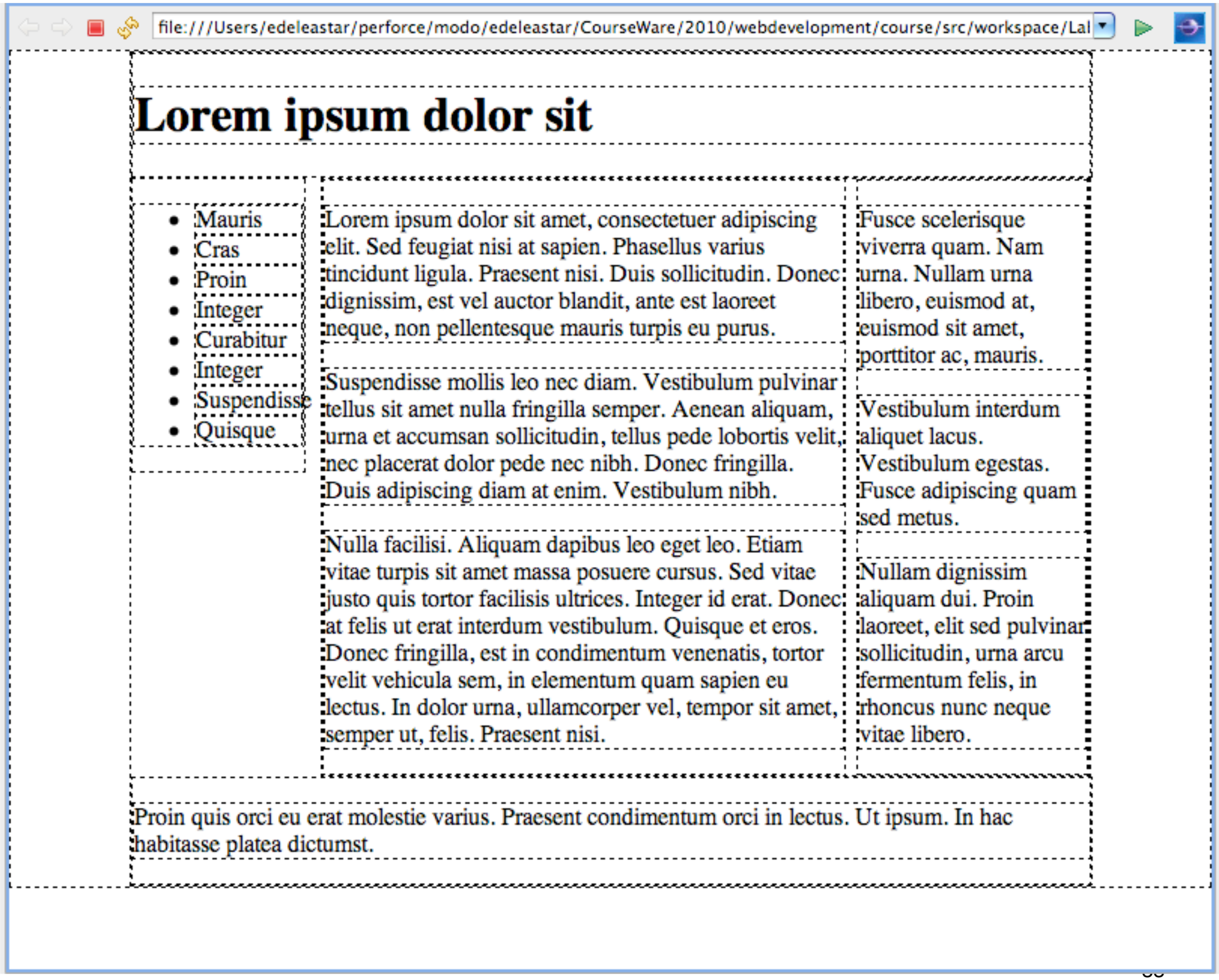
- "*" matches all elements.
- Simplifies CSS, and can be removed when layout under control




```css
*
{
  border: dashed thin;
}

body
{
  width: 80%;
  margin: 0 auto;
}

#maincontent
{
  width: 80%;
  float: right;
}

#navigation
{
  width: 18%;
  float: left;
}
#header
{
  width: 100%;
}

#footer
{
  width: 100%;
  clear: both;
}
```

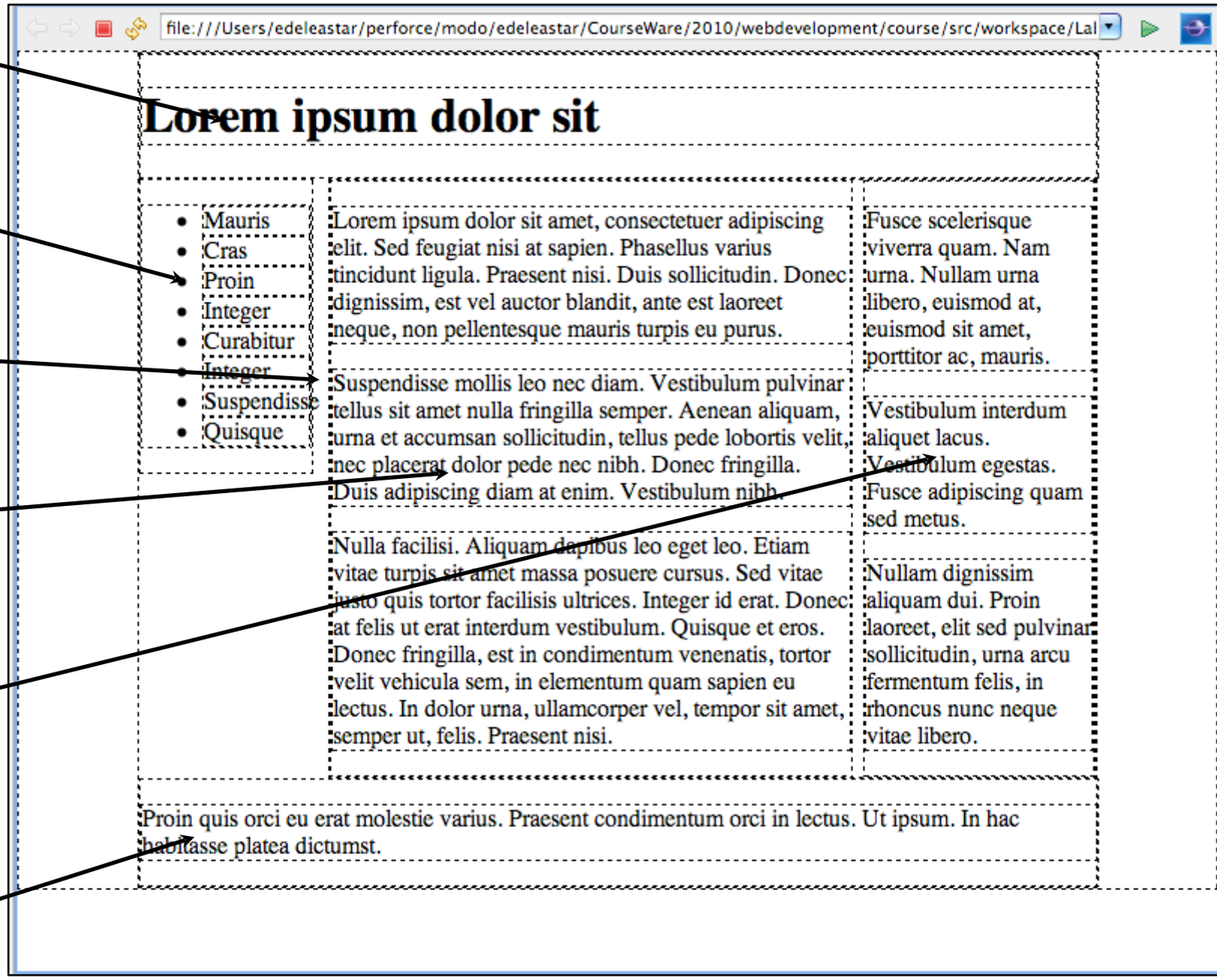
# Primary & Secondary

```
#primary
{
  width: 68%;
  float: left;
}

#secondary
{
  width: 30%;
  float: right;
}
```



# “CSS Craftsman”

```
#header
{
  width: 100%;
}

#navigation
{
  width: 18%;
  float: left;
}

#maincontent
{
  width: 80%;
  float: right;
}

#primary
{
  width: 68%;
  float: left;
}

#secondary
{
  width: 30%;
  float: right;
}
#footer
{
  width: 100%;
  clear: both;
}
```



file:///Users/edeleastar/perforce/modo/edeleastar/CourseWare/2010/webdevelopment/course/src/workspace/Lab12/WebContent/h
Lorem ipsum dolor sit
Mauris
Cras
Proin
Integer
Curabitur
Integer
Suspendisse
Quisque
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Sed feugiat nisi at sapien. Phasellus varius tincidunt ligula. Praesent nisi. Duis sollicitudin. Donec dignissim, est vel auctor blandit, ante est laoreet neque, non pellentesque mauris turpis eu purus.
Suspendisse mollis leo nec diam. Vestibulum pulvinar tellus sit amet nulla fringilla semper. Aenean aliquam, urna et accumsan sollicitudin, tellus pede lobortis velit, nec placerat dolor pede nec nibh. Donec fringilla. Duis adipiscing diam at enim. Vestibulum nibh.
Nulla facilisi. Aliquam dapibus leo eget leo. Etiam vitae turpis sit amet massa posuere cursus. Sed vitae justo quis tortor facilisis ultrices. Integer id erat. Donec at felis ut erat interdum vestibulum. Quisque et eros. Donec fringilla, est in condimentum venenatis, tortor velit vehicula sem, in elementum quam sapien eu lectus. In dolor urna, ullamcorper vel, tempor sit amet, semper ut, felis. Praesent nisi.
Fusce scelerisque viverra quam. Nam urna. Nullam urna libero, euismod at, euismod sit amet, porttitor ac, mauris.
Vestibulum interdum aliquet lacus. Vestibulum egestas. Fusce adipiscing quam sed metus.
Nullam dignissim aliquam dui. Proin laoreet, elit sed pulvinar sollicitudin, urna arcu fermentum felis, in rhoncus nunc neque vitae libero.
Proin quis orci eu erat molestie varius. Praesent condimentum orci in lectus. Ut ipsum. In hac habitasse platea dictumst.