INSTITUT
TEKNOLOGI
HARAPAN
BANGSA
School of Telematics

Desain Web IF 1P03

CSS 3

Ken Ratri. MT

2016

Konten

- **Display**
- **float**
- **Pseudo-classes**
- **Psedo-elemen**

Display

- The display property specifies if/how an element is displayed.
- Every HTML element has a default display value depending on what type of element it is. The default display value for most elements is block or inline.

Display

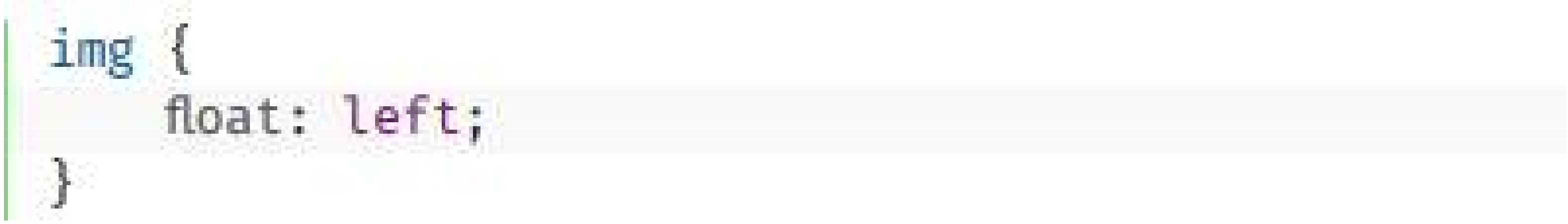
- An inline element does not start on a new line and only takes up as much width as necessary.
- display: none; is commonly used with JavaScript to hide and show elements without deleting and recreating them. Take a look at our last example on this page if you want to know how this can be achieved.
- A block-level element always starts on a new line and takes up the full width available

CSS Float

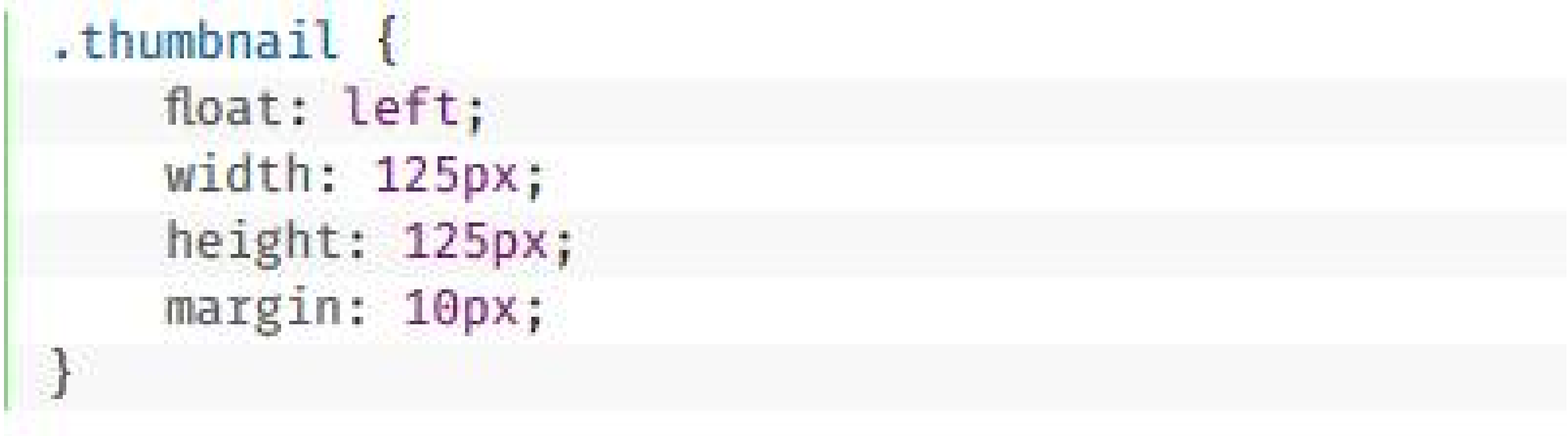
- You can float elements to the left or right, but only applies to the elements that generate boxes that are not absolutely positioned. Any element that follows the floated element will flow around the floated element on the other side.
- The float property may have one of the three values:

Value	Description
left	The element floats on the left side of its containing block.
right	The element floats on the right side of its containing block.
none	Removes the float property from an element.

example



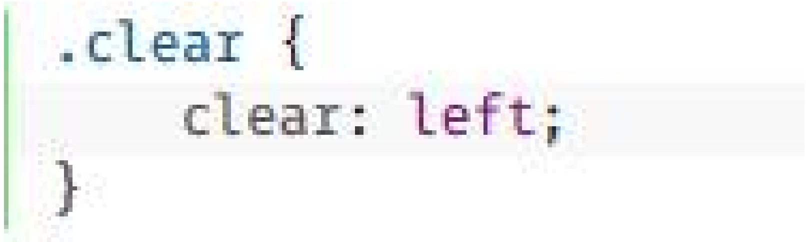
```css
img {
    float: left;
}
```



```css
.thumbnail {
    float: left;
    width: 125px;
    height: 125px;
    margin: 10px;
}
```

Turning off Float

- Elements that comes after the floating element will flow around it.The clear property specifies which sides of an element's box other floating elements are not allowed.

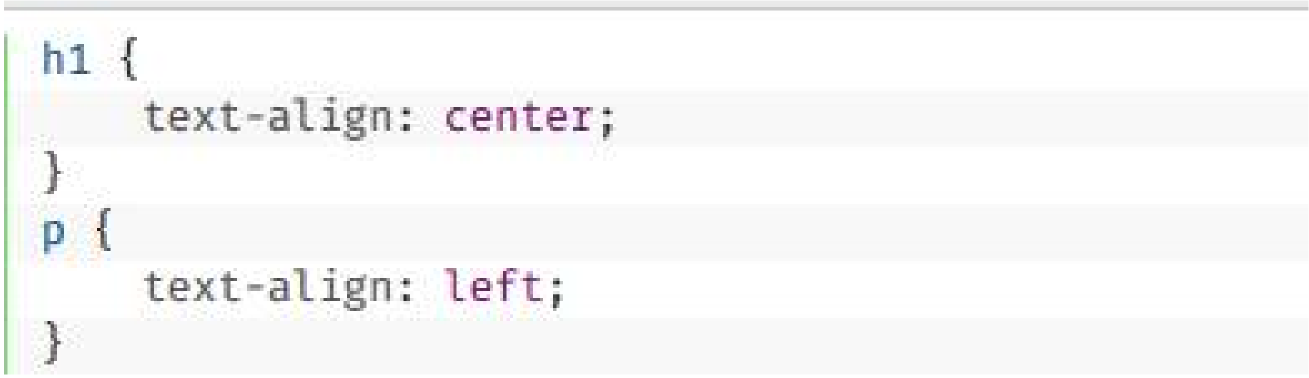


```
.clear {
    clear: left;
}
```

CSS Alignment

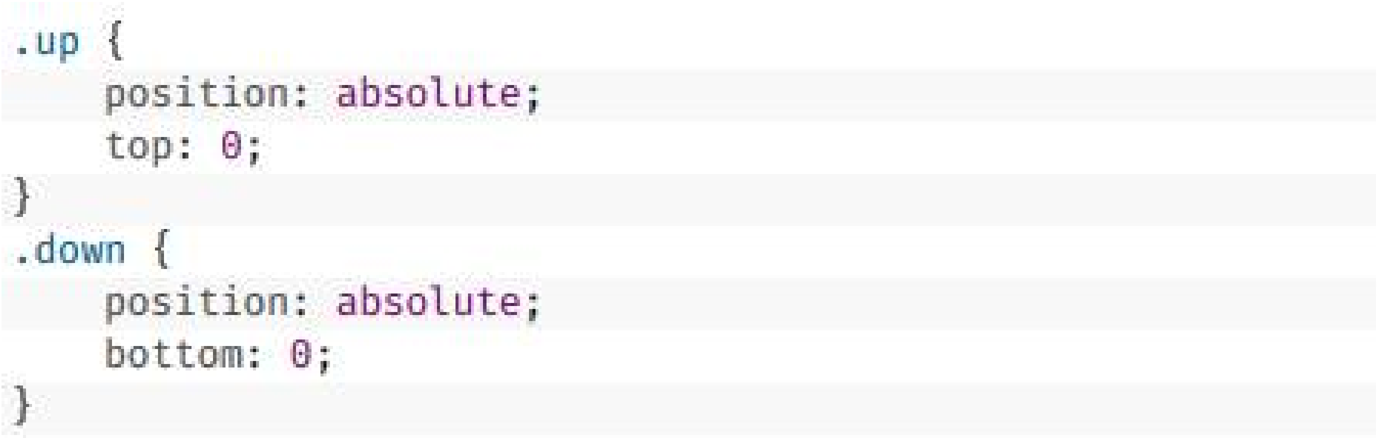
- CSS has several properties that can be used to align elements on the web pages.
- Text Alignment

Text inside the block-level elements can aligned by setting the text-align properly.

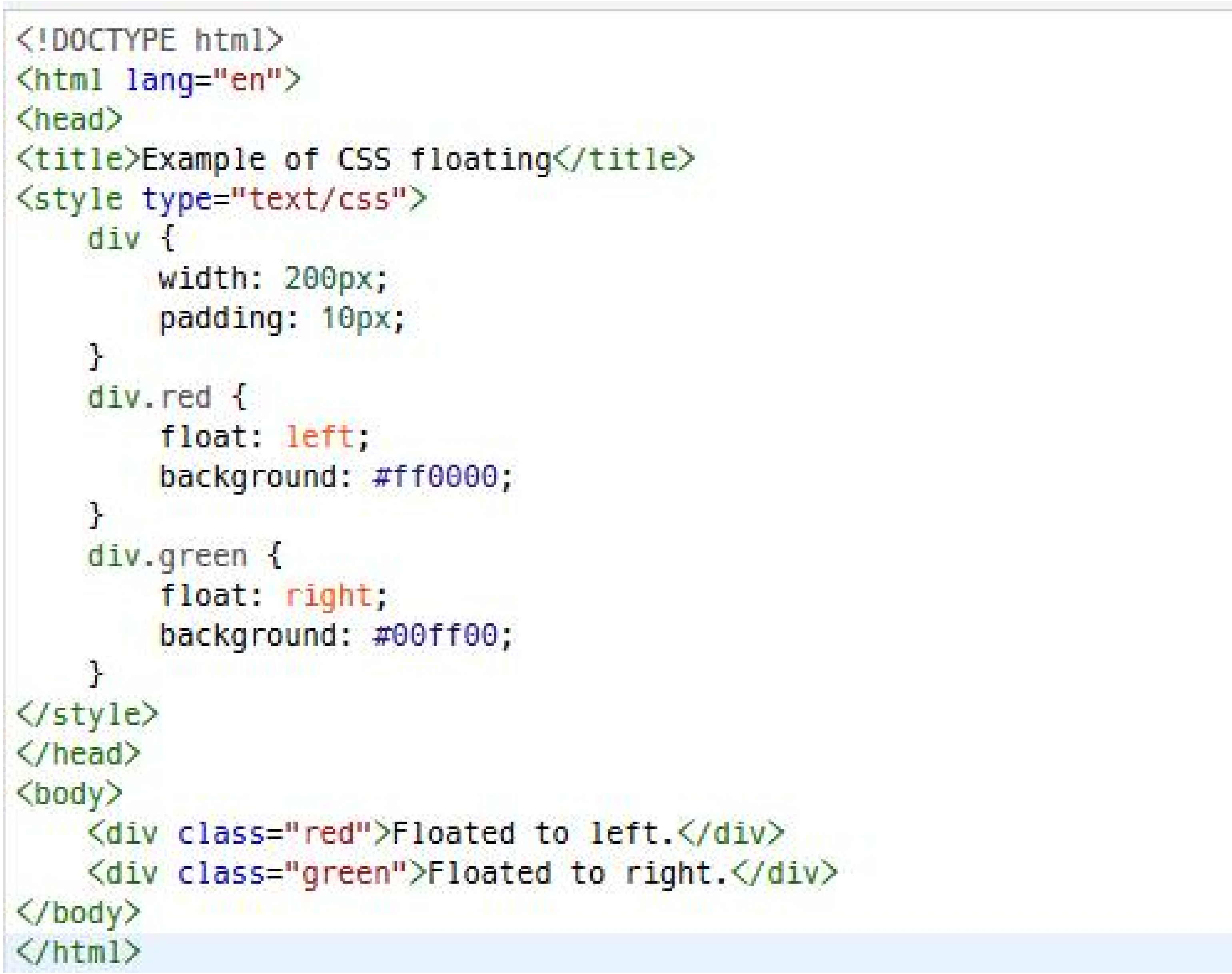


```
h1 {
    text-align: center;
}
p {
    text-align: left;
}
```

Aligning Elements Using the position Property



```
.up {
    position: absolute;
    top: 0;
}
.down {
    position: absolute;
    bottom: 0;
}
```

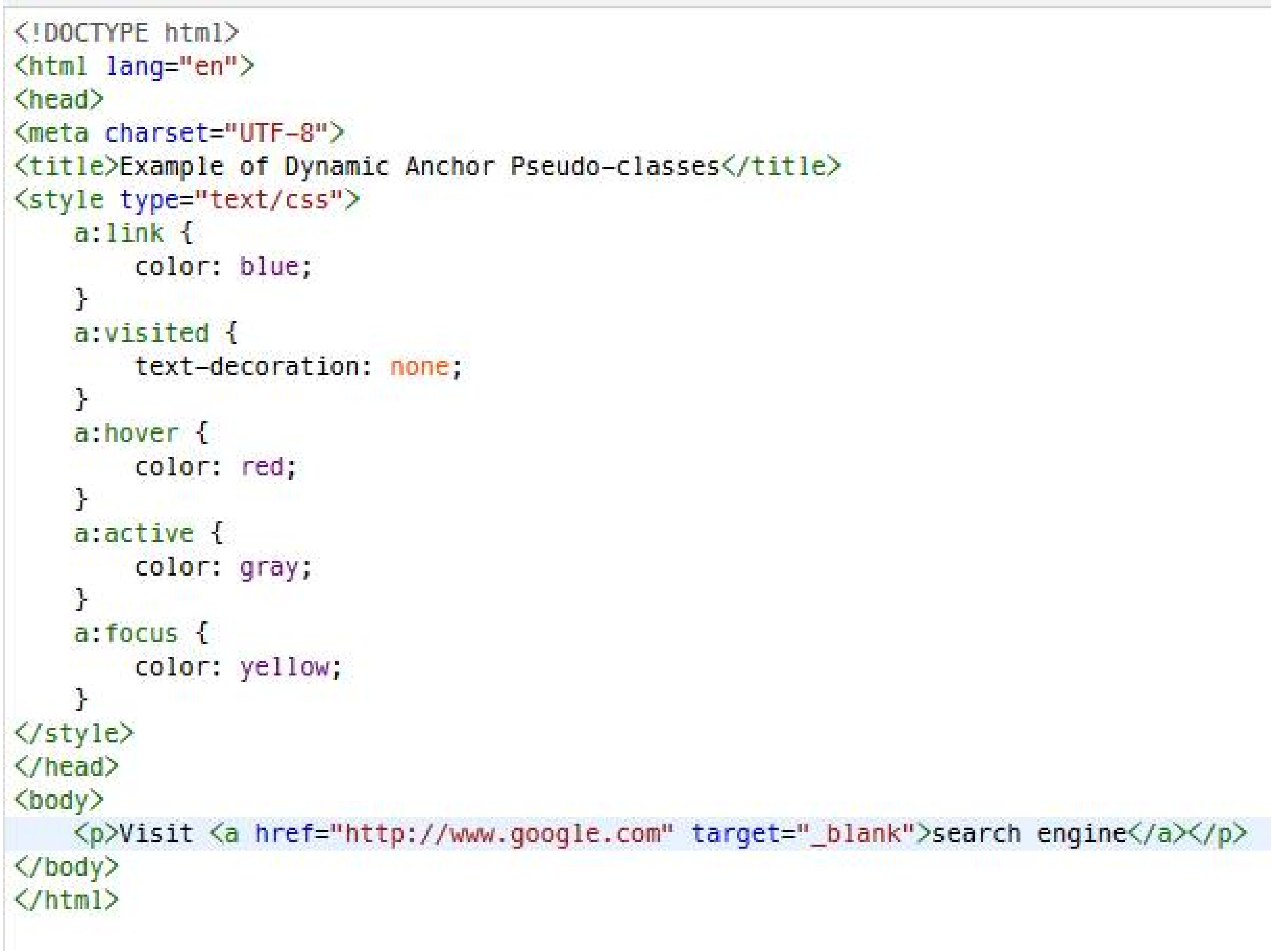


```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Example of CSS floating</title>
<style type="text/css">
    div {
        width: 200px;
        padding: 10px;
    }
    div.red {
        float: left;
        background: #ff0000;
    }
    div.green {
        float: right;
        background: #00ff00;
    }
</style>
</head>
<body>
    <div class="red">Floated to left.</div>
    <div class="green">Floated to right.</div>
</body>
</html>
```

Pseudo-classes

- The CSS pseudo-classes allow you to style the dynamic states of an element such as hover, active and focus state, as well as elements that are existing in the document tree but can't be targeted via the use of other selectors without adding any IDs or classes to them, for example, targeting the first or last child elements.
- A pseudo-class starts with a colon (:). Its syntax can be given with:The syntax of pseudo-classes:

```
selector:pseudo-class { property: value; }
```



```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Example of Dynamic Anchor Pseudo-classes</title>
<style type="text/css">
    a:link {
        color: blue;
    }
    a:visited {
        text-decoration: none;
    }
    a:hover {
        color: red;
    }
    a:active {
        color: gray;
    }
    a:focus {
        color: yellow;
    }
</style>
</head>
<body>
    <p>Visit <a href="http://www.google.com" target="_blank">search engine</a></p>
</body>
</html>
```

Pseudo-classes and CSS Classes

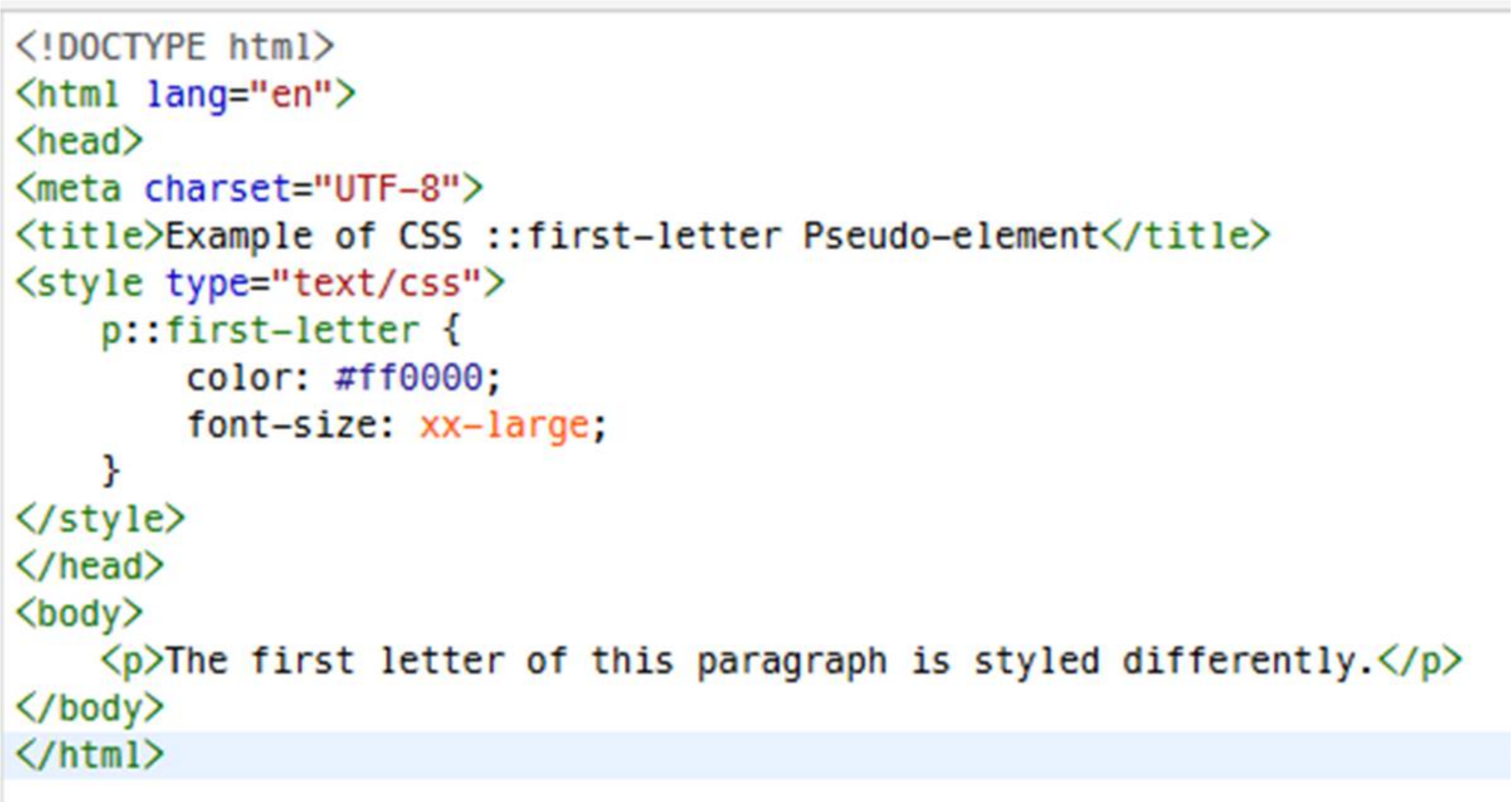
- Pseudo-classes can be combined with CSS classes:

```
<style>
a.highlight:hover {
    color: #ff0000;
}
</style>
</head>
<body>
<p><a class="highlight" href=“test.html">CSS Syntax</a></p>
<p><a href=“css.html">CSS Tutorial</a></p>
</body>
```

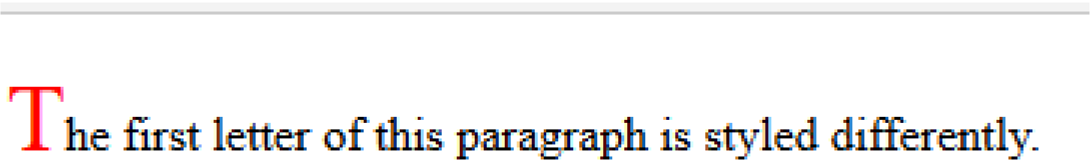
CSS Pseudo-elements

- The CSS pseudo-elements allow you to style the elements or parts of the elements without adding any IDs or classes to them.
- It will be really helpful in the situations when you just want to style the first letter of a paragraph to create the drop cap effect or you want to insert some content before or after an element, etc.
- CSS3 introduced a new double-colon (::) syntax for pseudo-elements to distinguish between them and pseudo-classes. The new syntax of the pseudo-element can be given with:

```
selector::pseudo-element { property: value; }
```



```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Example of CSS ::first-letter Pseudo-element</title>
<style type="text/css">
    p::first-letter {
        color: #ff0000;
        font-size: xx-large;
    }
</style>
</head>
<body>
    <p>The first letter of this paragraph is styled differently.</p>
</body>
</html>
```

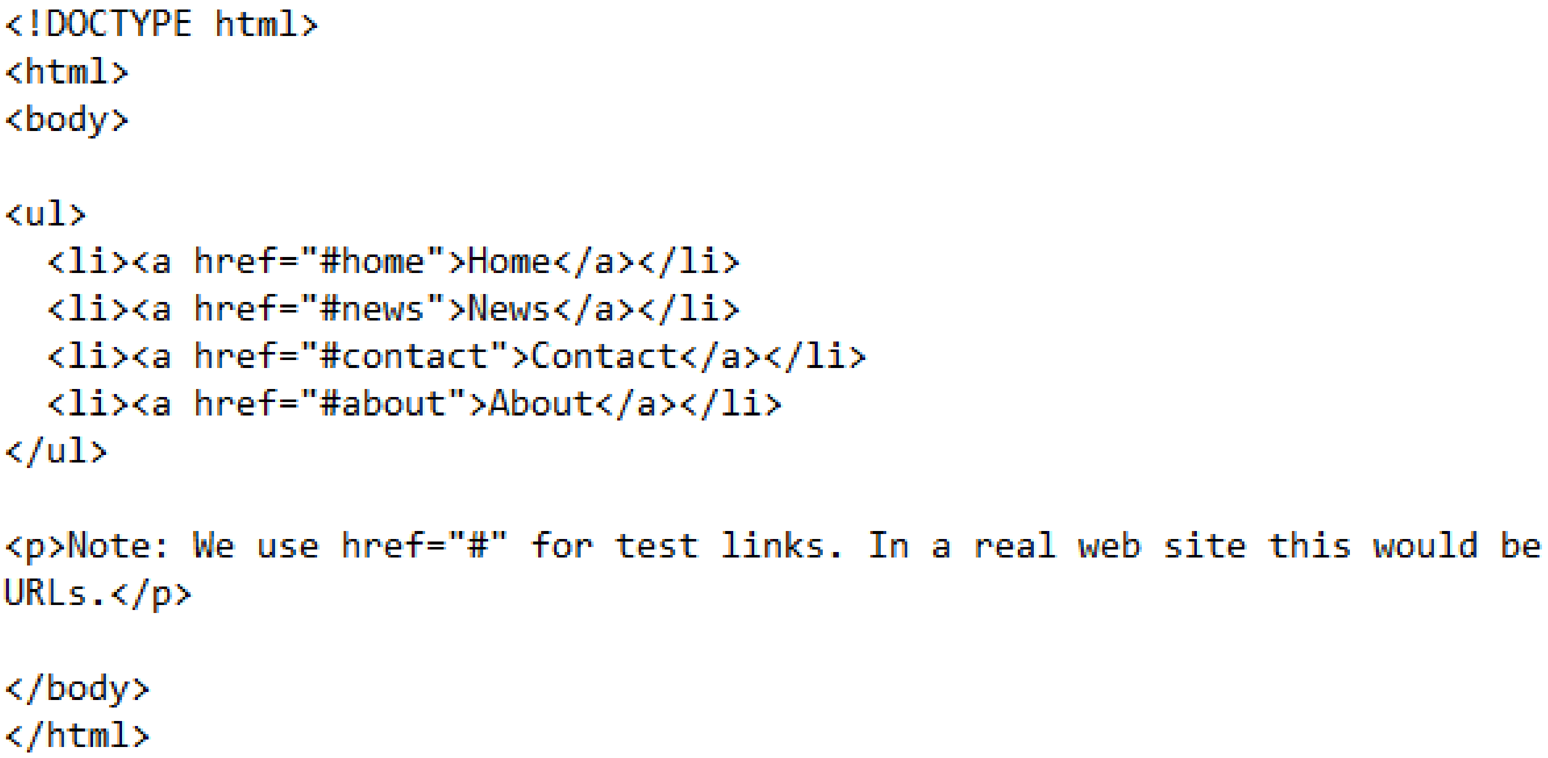


The first letter of this paragraph is styled differently.

Navigation Bars

- A navigation bar needs standard HTML as a base.
- In our examples we will build the navigation bar from a standard HTML list.
- A navigation bar is basically a list of links, so using the <ul> and <li> elements makes perfect sense:

```
<ul>
  <li><a href="default.asp">Home</a></li>
  <li><a href="news.asp">News</a></li>
  <li><a href="contact.asp">Contact</a></li>
  <li><a href="about.asp">About</a></li>
</ul>
```

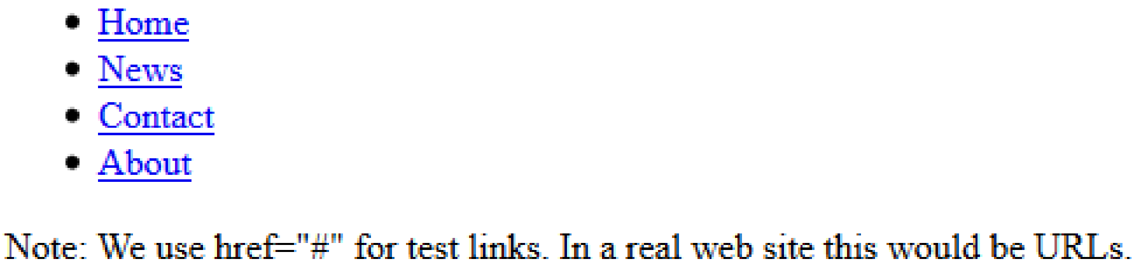


```
<!DOCTYPE html>
<html>
<body>

<ul>
  <li><a href="#home">Home</a></li>
  <li><a href="#news">News</a></li>
  <li><a href="#contact">Contact</a></li>
  <li><a href="#about">About</a></li>
</ul>

<p>Note: We use href="#" for test links. In a real web site this would be
URLs.</p>

</body>
</html>
```



- Home
- News
- Contact
- About

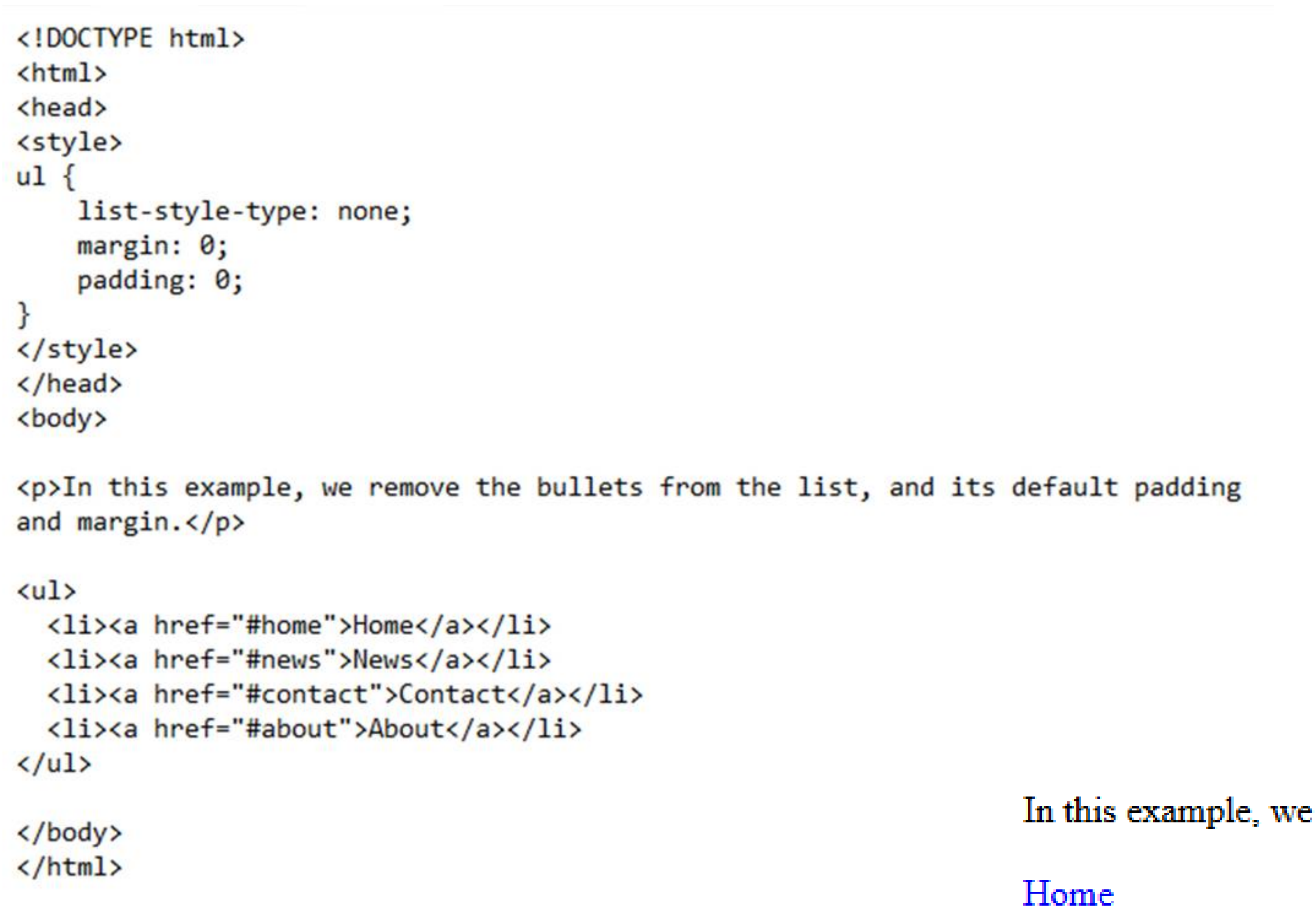
Note: We use href="#" for test links. In a real web site this would be URLs.

Navigation Bars

- Now let's remove the bullets and the margins and padding from the list:

```
ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
}
```

- list-style-type: none - Removes the bullets. A navigation bar does not need list markers
- Setting margins and padding to 0 to remove browser default settings

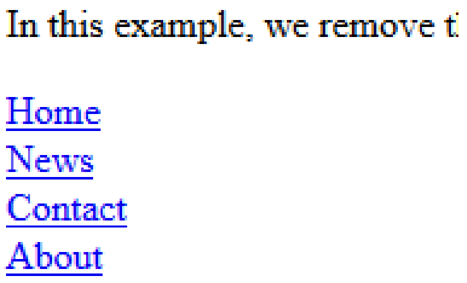


```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
}
</style>
</head>
<body>

<p>In this example, we remove the bullets from the list, and its default padding
and margin.</p>

<ul>
  <li><a href="#home">Home</a></li>
  <li><a href="#news">News</a></li>
  <li><a href="#contact">Contact</a></li>
  <li><a href="#about">About</a></li>
</ul>

</body>
</html>
```



In this example, we remove t

Home
News
Contact
About

Vertical Navigation Bar

- To build a vertical navigation bar we only need to style the <a> elements, in addition to the code above:

```
a {
    display: block;
    width: 60px;
}
```

- display: block - Displaying the links as block elements makes the whole link area clickable (not just the text), and it allows us to specify the width
- width: 60px - Block elements take up the full width available by default. We want to specify a 60 px width

Horizontal Navigation Bar

- There are two ways to create a horizontal navigation bar. Using **inline** or **floating** list items.
- Both methods work fine, but if you want the links to be the same size, you have to use the floating method.

```
li {
    display: inline;
}
```

- display: inline; - By default, <li> elements are block elements. Here, we remove the line breaks before and after each list item, to display them on one line

Horizontal Navigation Bar

Floating List Items

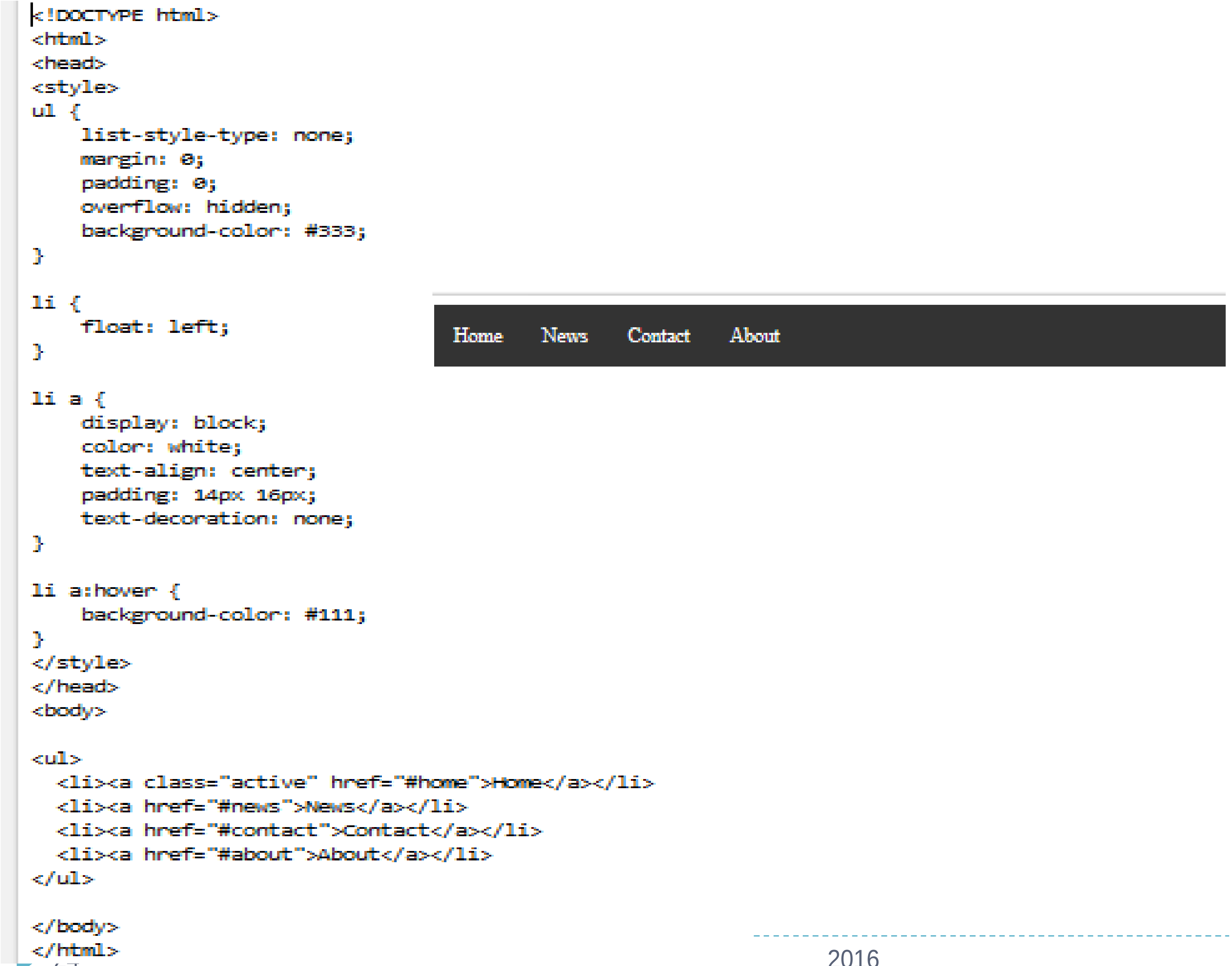
- In the example above the links have different widths.
- For all the links to have an equal width, float the <li> elements and specify a width for the <a> elements:

```
li {
    float: left;
}

a {
    display: block;
    width: 60px;
}
```

Horizontal Navigation Bar

- float: left - use float to get block elements to slide next to each other
- display: block - Displaying the links as block elements makes the whole link area clickable (not just the text), and it allows us to specify the width
- width: 60px - Since block elements take up the full width available, they cannot float next to each other. We specify the width of the links to 60px



```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
    overflow: hidden;
    background-color: #333;
}

li {
    float: left;
}

li a {
    display: block;
    color: white;
    text-align: center;
    padding: 14px 16px;
    text-decoration: none;
}

li a:hover {
    background-color: #111;
}
</style>
</head>
<body>

<ul>
  <li><a class="active" href="#home">Home</a></li>
  <li><a href="#news">News</a></li>
  <li><a href="#contact">Contact</a></li>
  <li><a href="#about">About</a></li>
</ul>

</body>
</html>
```