CENG420

Chapter 3 CSS

Cascading Style Sheets

Fall 2019-2020 LIU/BIU Dr. Ghantous

Some notes about HTML

- Most HTML tags have properties that store presentation formats
- For example, ,<h2> has font-size
- If not specified, browsers choose default value
- If specified, it overrides the default

Cascaded because they can be defined at different levels

- Cascaded Style sheets are a syntactic mechanism to specify the style

Lower levels override higher levels

- CSS1 in 1996, CSS2 in 1998, CSS2.1 in 2009, CSS3

Levels of style sheets

- Lowest → *inline*
- *Document level*
- Highest → *External*

Applied to single HTML element

Applies to whole body of document

Applies to whole bodies of any number of documents

Inline style sheets

- Defeat the purpose of using a style sheet
- Deprecated in XHTML 1.1

Document level style sheets

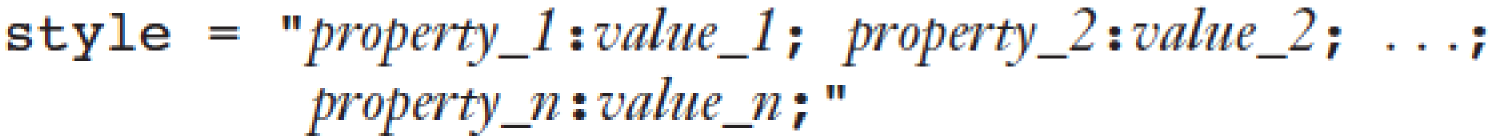
- Appear in the head section
- Applied to entire body

External style sheets

- Leave it for later!

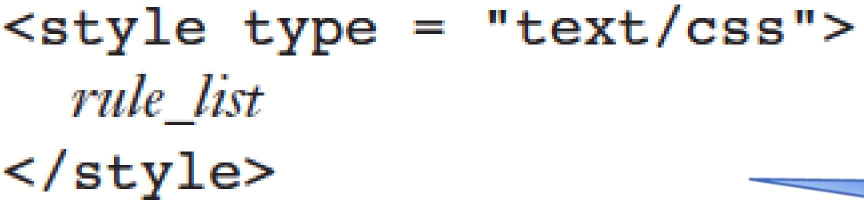
Style specification formats

- *Inline*:



```
style = "property_1:value_1; property_2:value_2; ...;
         property_n:value_n;"
```

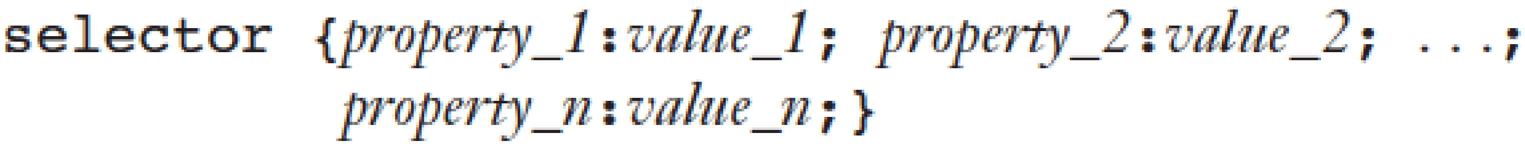
- *Document-leve (inside the <head> section:*



```
<style type = "text/css">
  rule_list
</style>
```

Comments /* … */

  - Each rule:



```
selector {property_1:value_1; property_2:value_2; ...;
          property_n:value_n;}
```

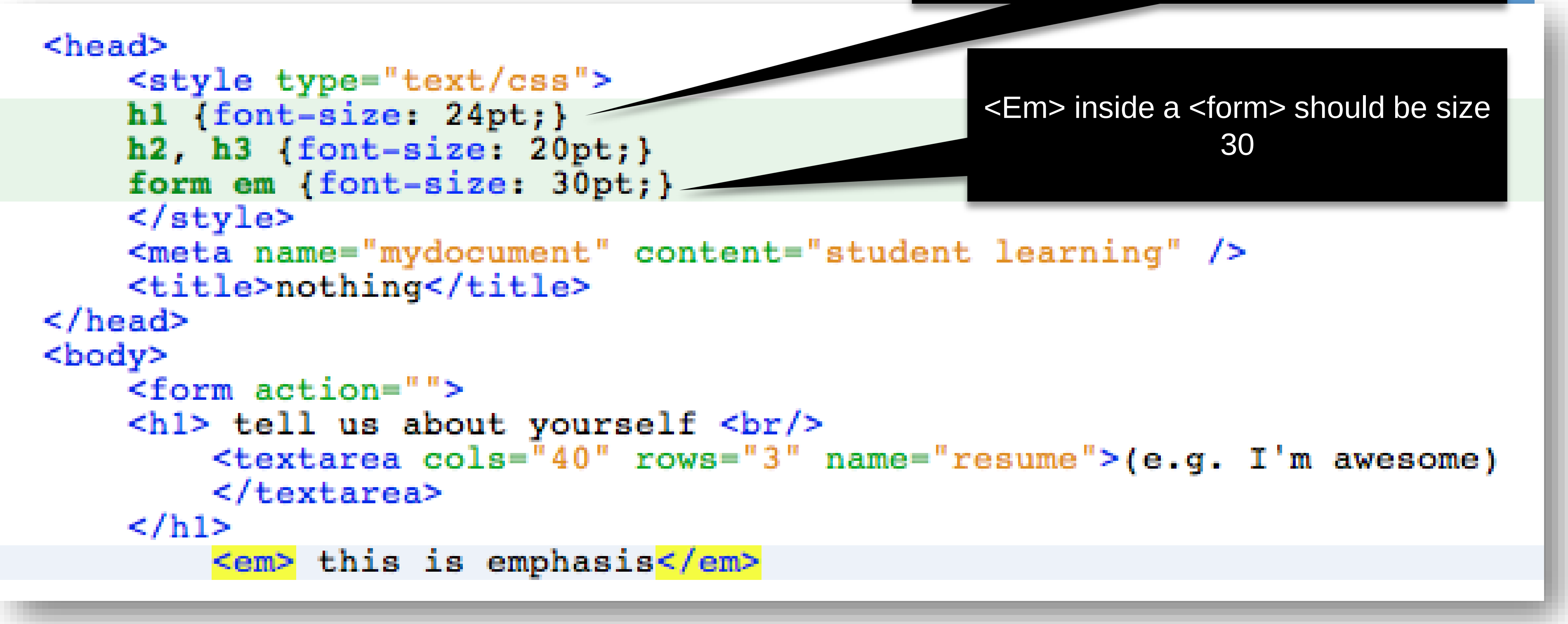
Example

```
<style type=’text/css’>

    p {color: red;}

</style>
```

This means, all “p” elements in the page will be red.

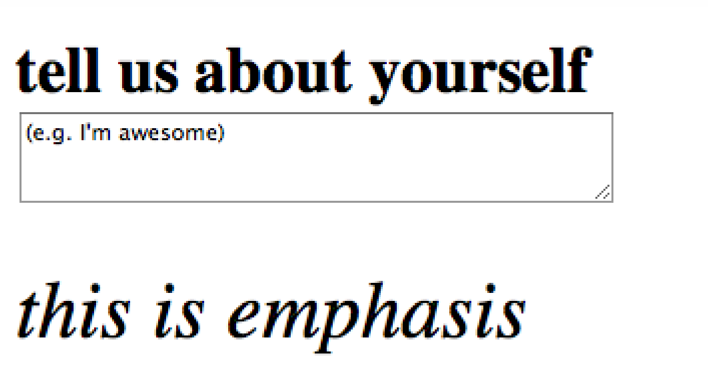
Selector forms

H1 follows font size of 24
H2, and h3 have font size of 20



```
<head>
    <style type="text/css">
    h1 {font-size: 24pt;}
    h2, h3 {font-size: 20pt;}
    form em {font-size: 30pt;}
    </style>
    <meta name="mydocument" content="student learning" />
    <title>nothing</title>
</head>
<body>
    <form action="">
    <h1> tell us about yourself <br/>
        <textarea cols="40" rows="3" name="resume">(e.g. I'm awesome)
        </textarea>
    </h1>
        <em> this is emphasis</em>
```

<Em> inside a <form> should be size 30



tell us about yourself

(e.g. I'm awesome)

this is emphasis

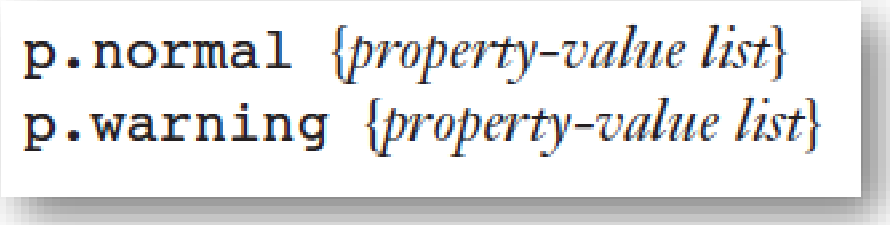
Even if inline is not used anymore but we can still use it for quick styles

- **Example:**

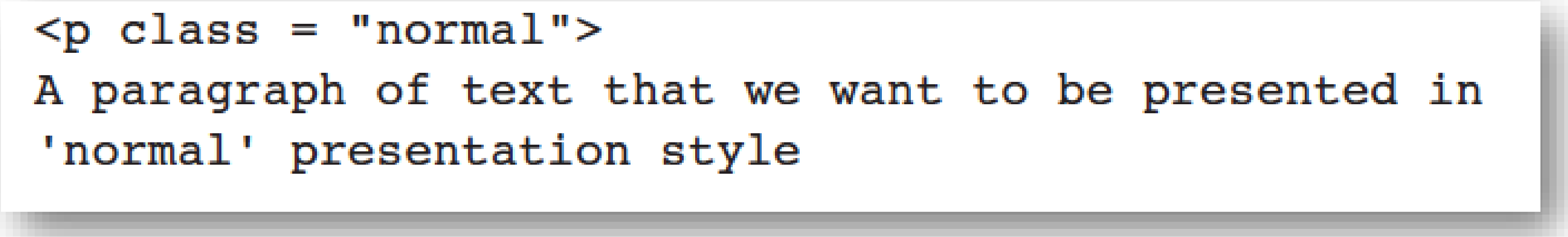
```
<div id="chat"
     style="position:absolute;
            height:80%;
        width:100%;
        overflow-y: scroll;”>
</div>
```

Class selectors

- Create **`classes`** to allow different occurrences of the **same** tag with different styles.
- You choose the class name, for example normal, warning… and you use the dot after the element name.
- Then use the property **`class`** inside your elements.



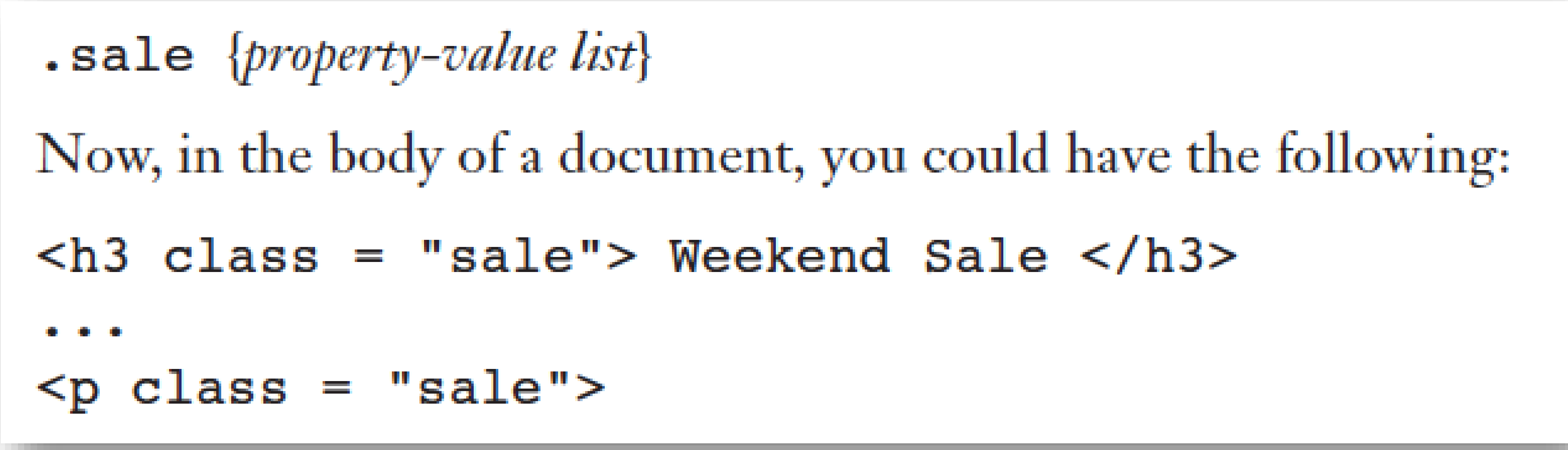
```
p.normal {property-value list}
p.warning {property-value list}
```



```
<p class = "normal">
A paragraph of text that we want to be presented in
'normal' presentation style
```

Generic selectors

- Create a class that applied to **more than one** kind of tag.

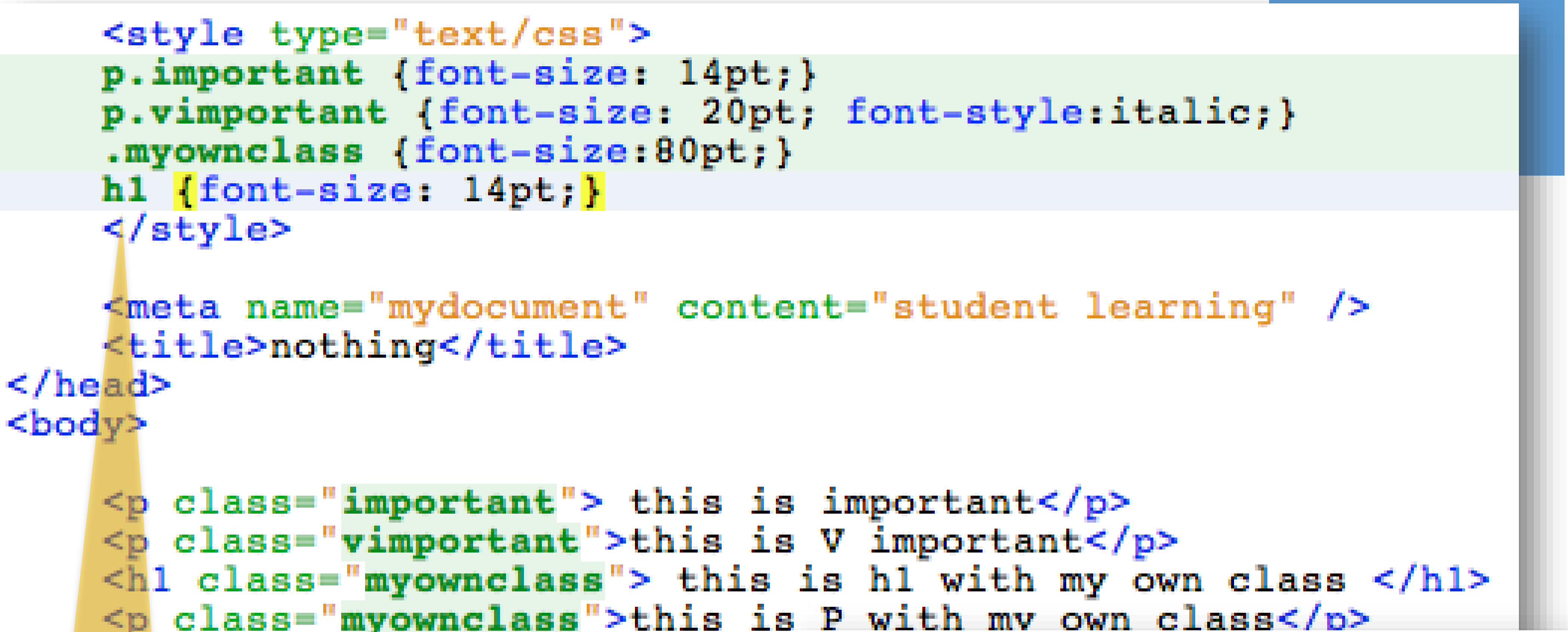


```
.sale {property-value list}
```

Now, in the body of a document, you could have the following:

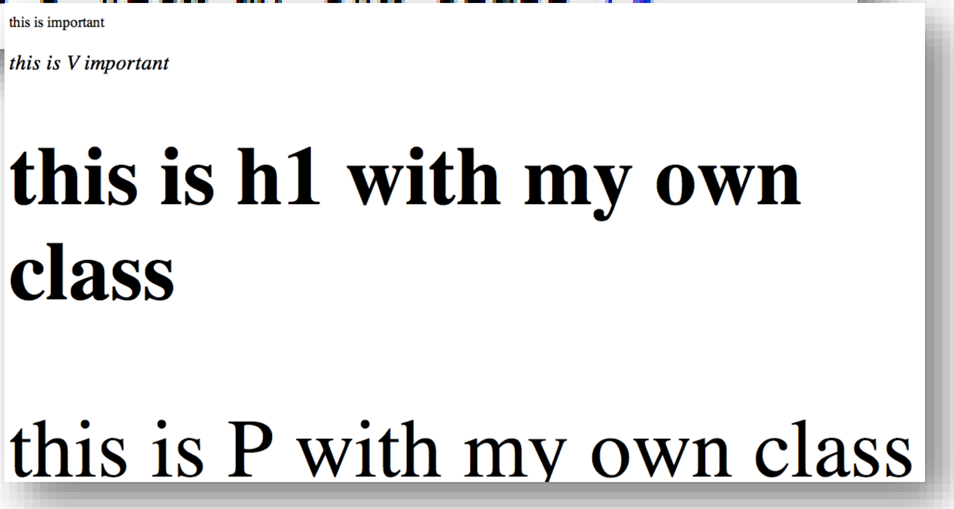
```
<h3 class = "sale"> Weekend Sale </h3>
...
<p class = "sale">
```

Selectors examples



```
    <style type="text/css">
    p.important {font-size: 14pt;}
    p.vimportant {font-size: 20pt; font-style:italic;}
    .myownclass {font-size:80pt;}
    h1 {font-size: 14pt;}
    </style>

    <meta name="mydocument" content="student learning" />
    <title>nothing</title>
</head>
<body>

    <p class="important"> this is important</p>
    <p class="vimportant">this is V important</p>
    <h1 class="myownclass"> this is h1 with my own class </h1>
    <p class="myownclass">this is P with my own class</p>
```

Both work on h1, seems like the the class overrides the h1 style



this is important

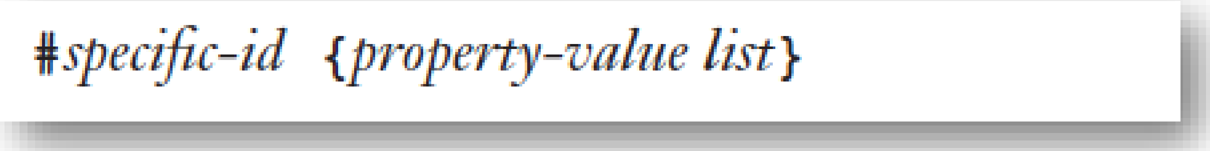
this is V important

this is h1 with my own class

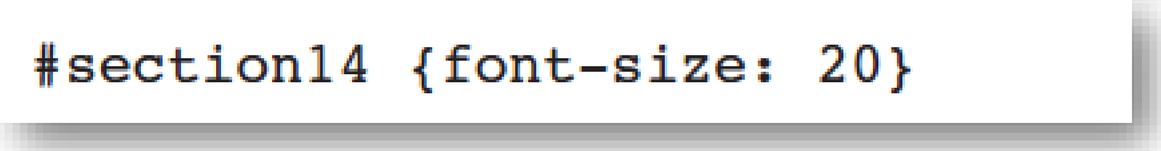
this is P with my own class

Id selector

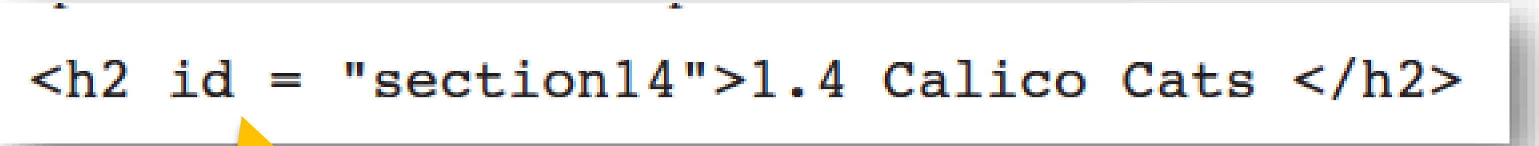
- Application of a style to 1 specific element



#specific-id {property-value list}



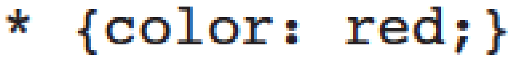
```
#section14 {font-size: 20}
```



```
<h2 id = "section14">1.4 Calico Cats </h2>
```

ID here is used above in the style

Universal selec

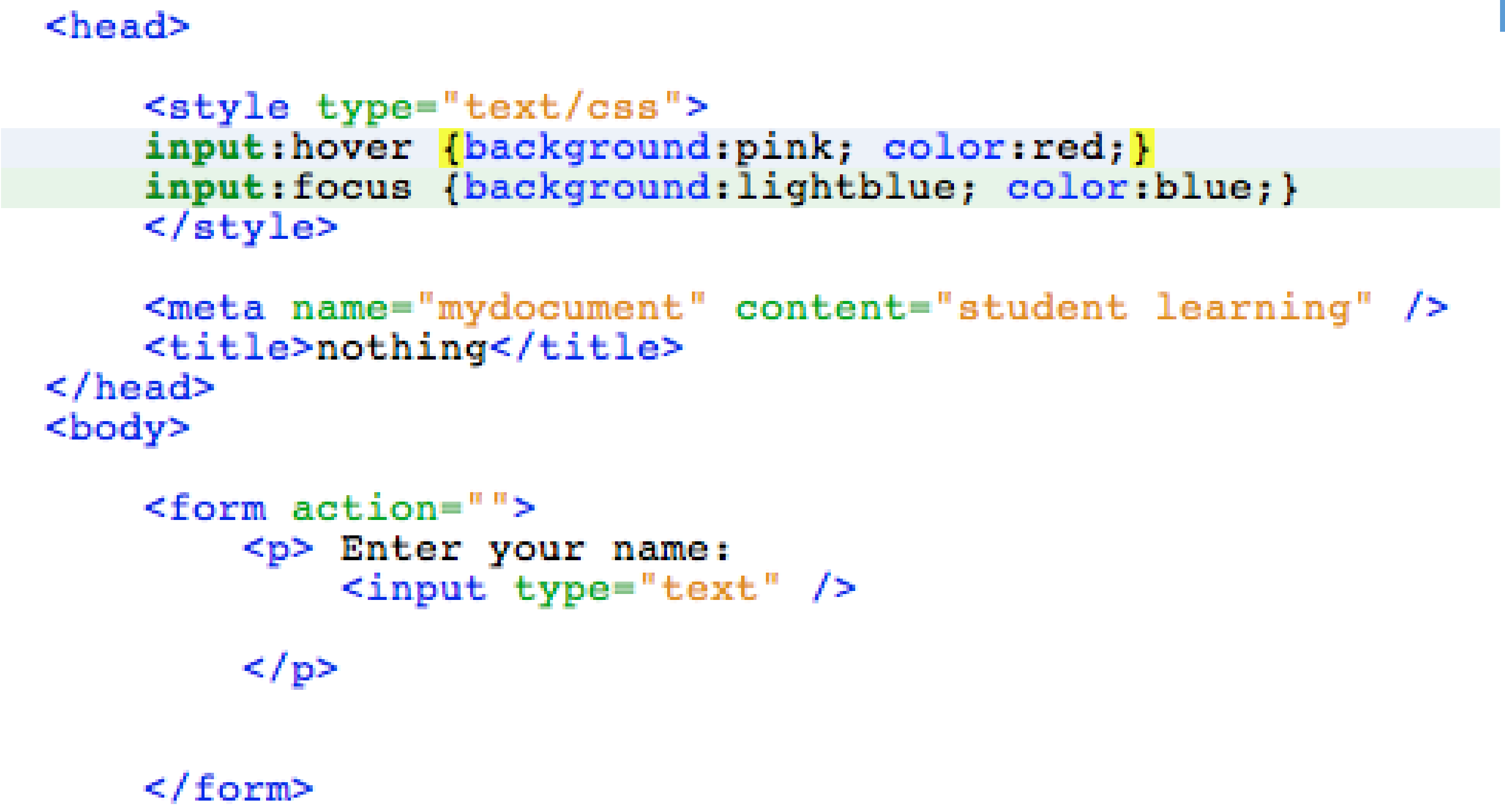


```
* {color: red;}
```

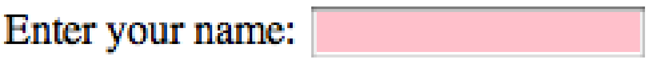
Makes all document red

Pseudo Classes

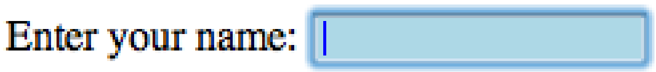
- Examples are **hover** and **focus**: activated only when triggered



```
<head>

    <style type="text/css">
    input:hover {background:pink; color:red;}
    input:focus {background:lightblue; color:blue;}
    </style>

    <meta name="mydocument" content="student learning" />
    <title>nothing</title>
</head>
<body>

    <form action="">
        <p> Enter your name:
            <input type="text" />

        </p>


    </form>
```



Enter your name:

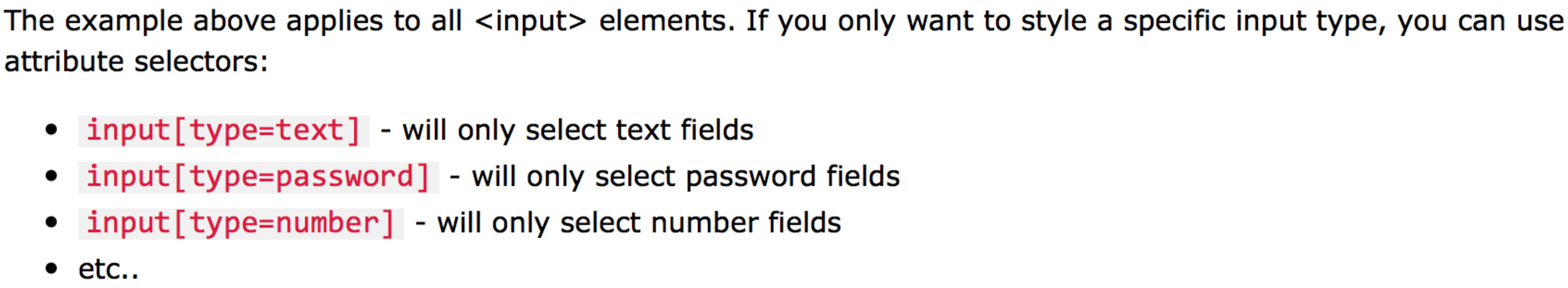


Enter your name:

Other examples include: `a:link, a:visited, a.someclass:selected,….`

specific **input** css

- if you use **input {…}** this applies to all input styles



The example above applies to all <input> elements. If you only want to style a specific input type, you can use attribute selectors:

- `input[type=text]` - will only select text fields
- `input[type=password]` - will only select password fields
- `input[type=number]` - will only select number fields
- etc..

Example

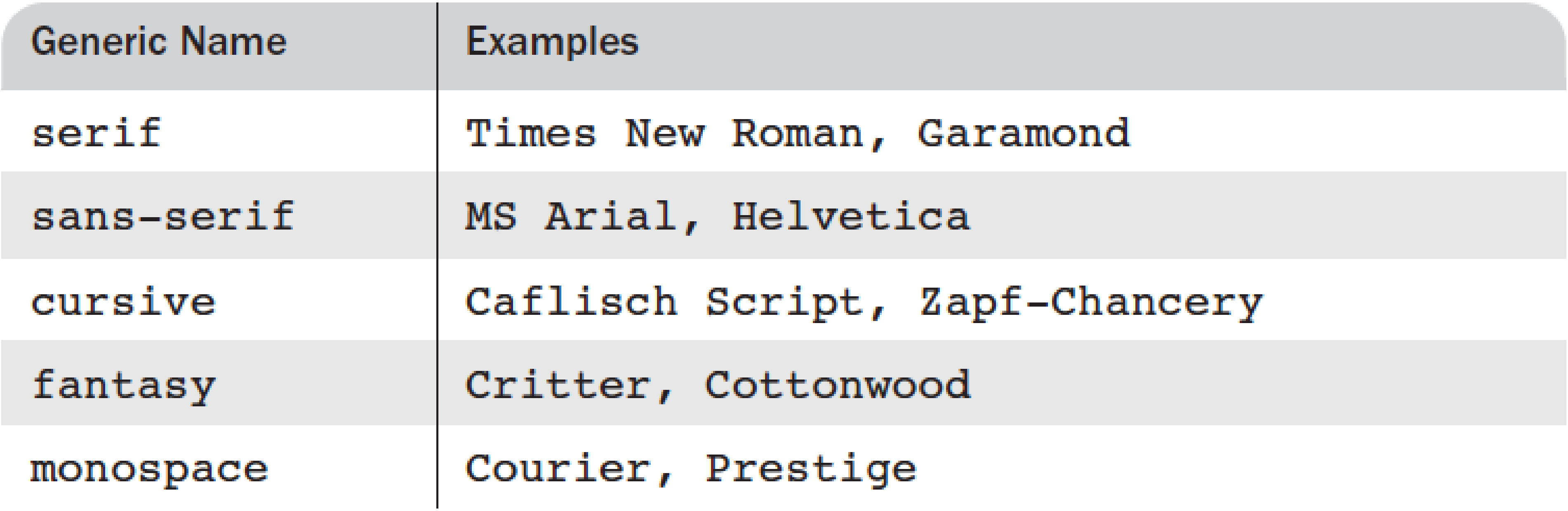
```
input[type=text] {
    width: 100%;
    padding: 12px 20px;
    margin: 8px 0;
    box-sizing: border-box;
}
```

3.6 Font properties

- Font-family

```
font-family: Arial, Helvetica, Futura
```

Tells the browser to use Arial is supported, if not then helvetica, and so on



Generic Name	Examples
serif	Times New Roman, Garamond
sans-serif	MS Arial, Helvetica
cursive	Caflisch Script, Zapf-Chancery
fantasy	Critter, Cottonwood
monospace	Courier, Prestige

We can use a generic name at the end

```
font-family: Arial, Helvetica, Futura, sans-serif
```

If font is more than 1 word

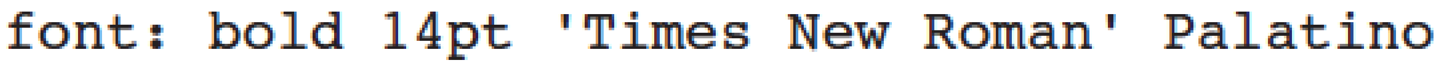
```
font-family: 'Times New Roman'
```

3.6 Font properties

- Font-size
- Can be defined as xx-small, x-small, small, xx-large, or as percentage, or numbers.
- Font-variant → small-caps to show small capital letters
- Font-Style:italic
- Font-weight:bold (can also be normal, bolder, lighter that are relative, or number between 100 and 900 mutliple of 100. 400 normal and 700 is bold

3.6 Font properties

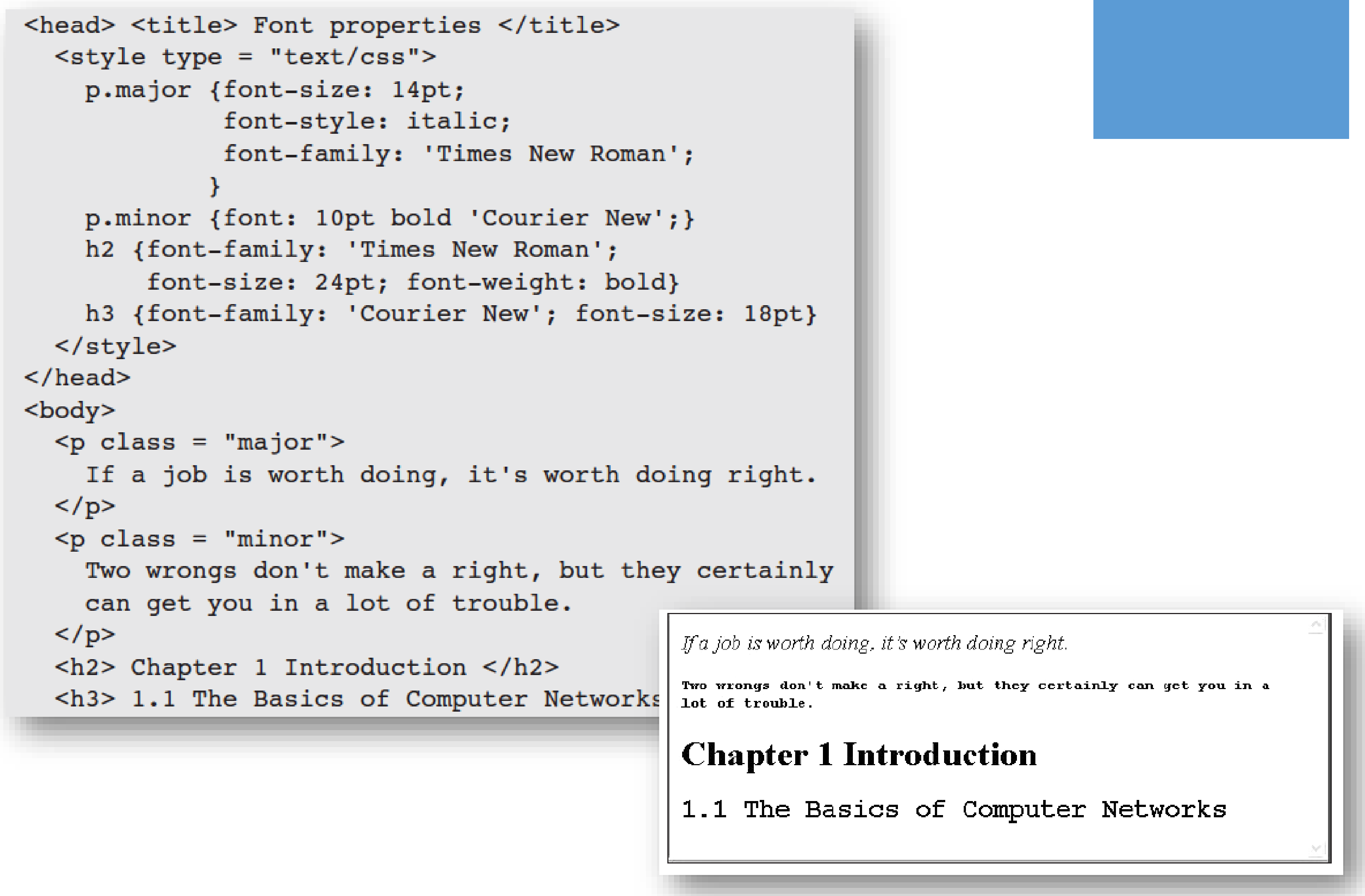
- Font shorthand
- Used when multiple properties



```
font: bold 14pt 'Times New Roman' Palatino
```

Font family must be last, size must be second to last, others can be in any order but before family and size.
When we use font:, the only required are family and size

Example



```
<head> <title> Font properties </title>
  <style type = "text/css">
    p.major {font-size: 14pt;
             font-style: italic;
             font-family: 'Times New Roman';
            }
    p.minor {font: 10pt bold 'Courier New';}
    h2 {font-family: 'Times New Roman';
        font-size: 24pt; font-weight: bold}
    h3 {font-family: 'Courier New'; font-size: 18pt}
  </style>
</head>
<body>
  <p class = "major">
    If a job is worth doing, it's worth doing right.
  </p>
  <p class = "minor">
    Two wrongs don't make a right, but they certainly
    can get you in a lot of trouble.
  </p>
  <h2> Chapter 1 Introduction </h2>
  <h3> 1.1 The Basics of Computer Networks
```

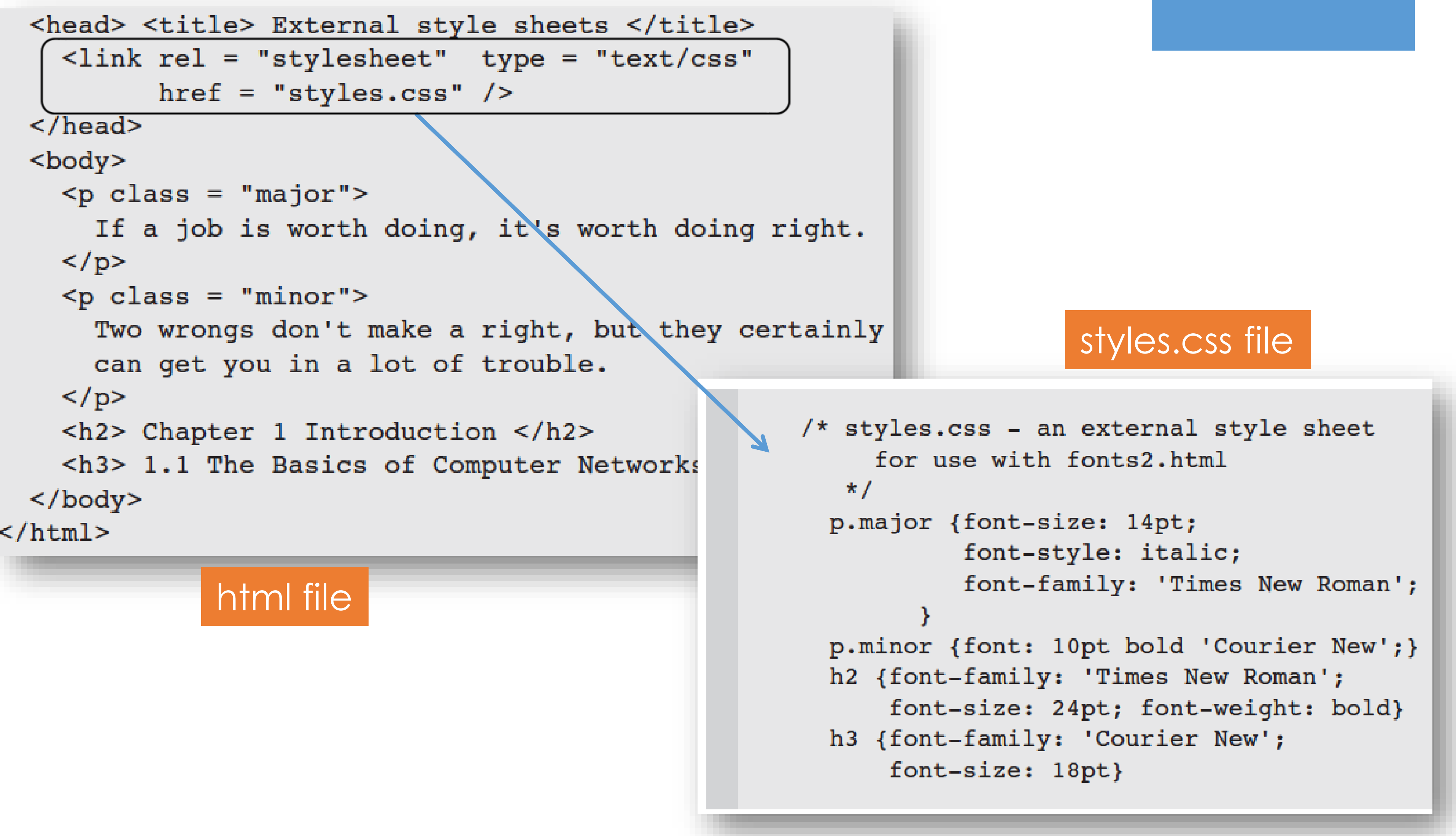
If a job is worth doing, it's worth doing right.

Two wrongs don't make a right, but they certainly can get you in a lot of trouble.

Chapter 1 Introduction

1.1 The Basics of Computer Networks

External style sheet



```
<head> <title> External style sheets </title>
  <link rel = "stylesheet"  type = "text/css"
        href = "styles.css" />
</head>
<body>
  <p class = "major">
    If a job is worth doing, it's worth doing right.
  </p>
  <p class = "minor">
    Two wrongs don't make a right, but they certainly
    can get you in a lot of trouble.
  </p>
  <h2> Chapter 1 Introduction </h2>
  <h3> 1.1 The Basics of Computer Networks
</body>
</html>
```

html file

styles.css file

```
/* styles.css - an external style sheet
     for use with fonts2.html
  */
 p.major {font-size: 14pt;
          font-style: italic;
          font-family: 'Times New Roman';
       }
 p.minor {font: 10pt bold 'Courier New';}
 h2 {font-family: 'Times New Roman';
     font-size: 24pt; font-weight: bold}
 h3 {font-family: 'Courier New';
     font-size: 18pt}
```

The ***em*** unit

- **em** unit is very useful for scalable style sheets.
- This is important when modifying the size of the browser window. This keeps everything relative.

- The **em** unit, when used with font sizes, refers to the font size of the parent element.
- **1em** is equal to the current font size. The default text size in browsers is 16px. So, the default size of 1em is 16px.
- The size can be calculated from pixels to em using this formula: *pixels*/16=*em*
- For example 2em, 1.5em, … and so on
- Can be used with padding, margins,...
- Try it!

To use your **own font**

- Download any font you want (it's a file with either TFF or OTF extension). You can use *google fonts*, *awesome fonts…*
- Add this tff or otf file to your folder or subfolder with your *html*, *php* and *css* files.
- For example, if you download a font called **sansation_light.tff**

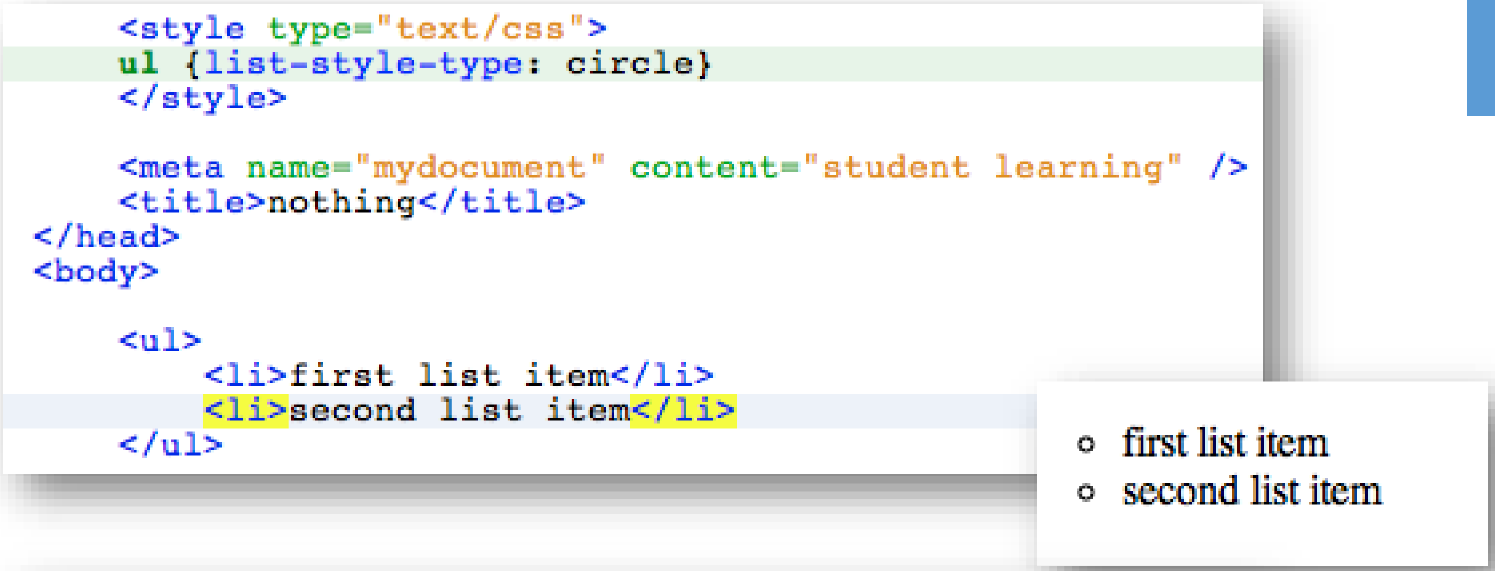
```
@font-face {
    font-family: myFirstFont;
    src: url(sansation_light.tff);
}

div {
    font-family: myFirstFont;
}
```

3.7 List Properties

- Unordered lists can have `list-style-type` to specify the bullet shape: `Disc, circle, square, and none.`
- Default is `Disc`

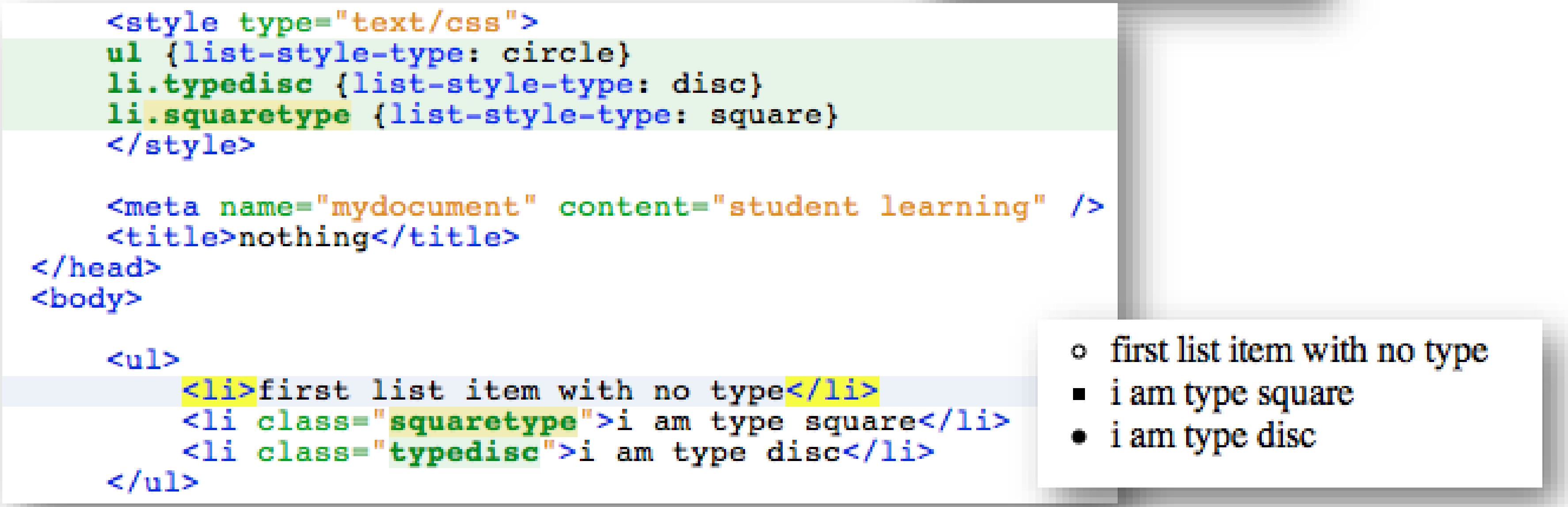
- Also it can be an image defined as `list-style-image: url (myimage.jpg)`

3.7 List Properties: examples



```
    <style type="text/css">
    ul {list-style-type: circle}
    </style>

    <meta name="mydocument" content="student learning" />
    <title>nothing</title>
</head>
<body>

    <ul>
        <li>first list item</li>
        <li>second list item</li>
    </ul>
```

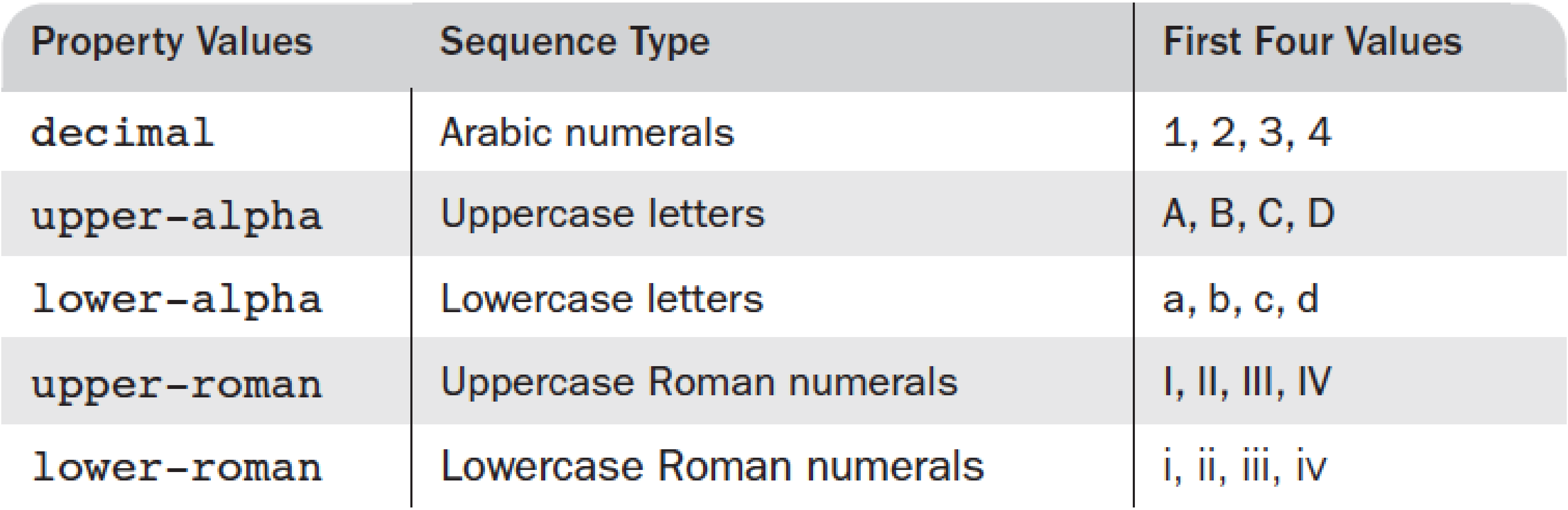
- ○ first list item
- ○ second list item



```
    <style type="text/css">
    ul {list-style-type: circle}
    li.typedisc {list-style-type: disc}
    li.squaretype {list-style-type: square}
    </style>

    <meta name="mydocument" content="student learning" />
    <title>nothing</title>
</head>
<body>

    <ul>
        <li>first list item with no type</li>
        <li class="squaretype">i am type square</li>
        <li class="typedisc">i am type disc</li>
    </ul>
```

- ○ first list item with no type
- ■ i am type square
- ● i am type disc

3.7 Ordered lists

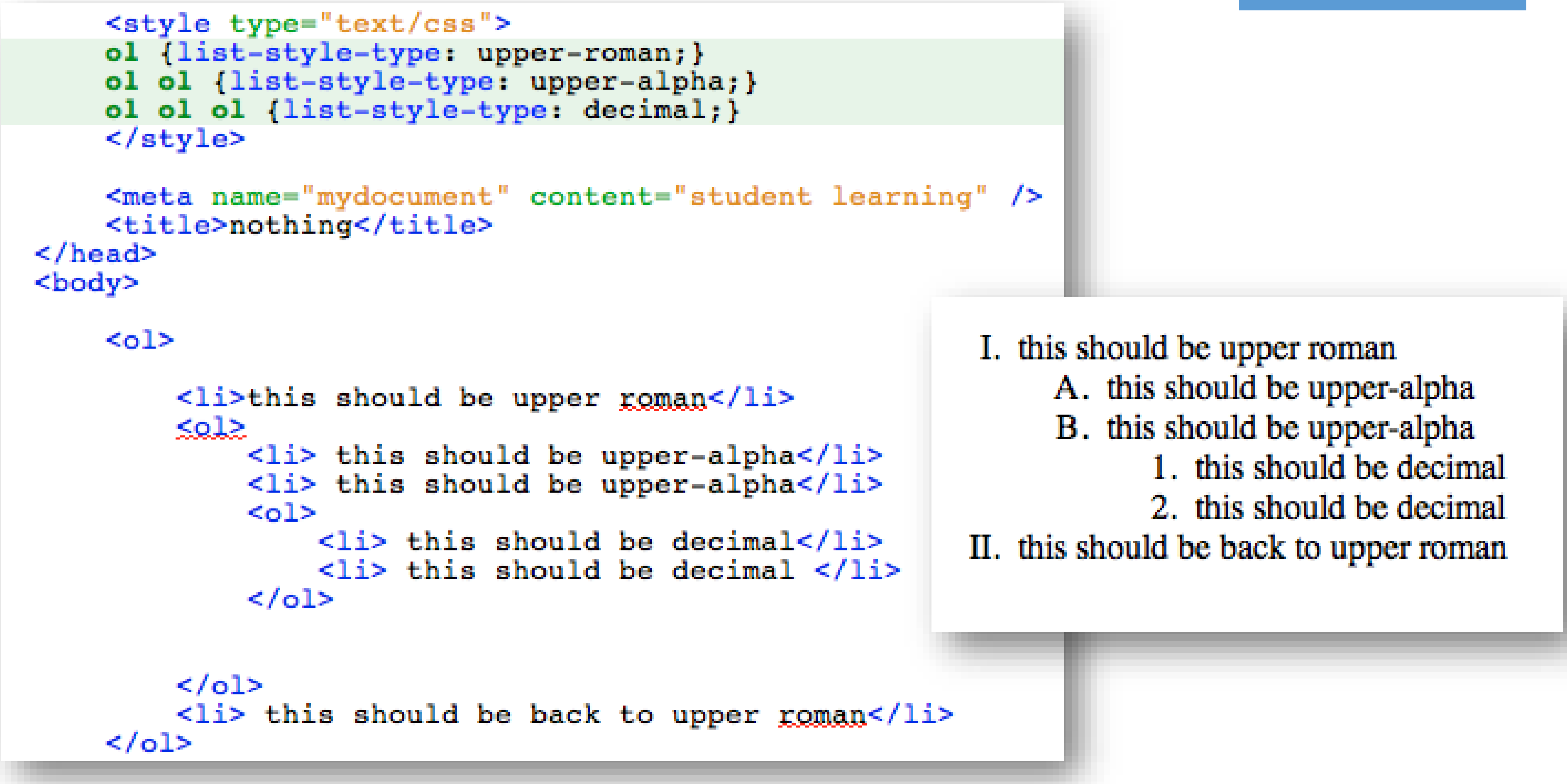


Property Values	Sequence Type	First Four Values
`decimal`	Arabic numerals	1, 2, 3, 4
`upper-alpha`	Uppercase letters	A, B, C, D
`lower-alpha`	Lowercase letters	a, b, c, d
`upper-roman`	Uppercase Roman numerals	I, II, III, IV
`lower-roman`	Lowercase Roman numerals	i, ii, iii, iv

More styles in the book

3.7 Nested Ordered lists

Ol
Ol ol
Ol ol ol

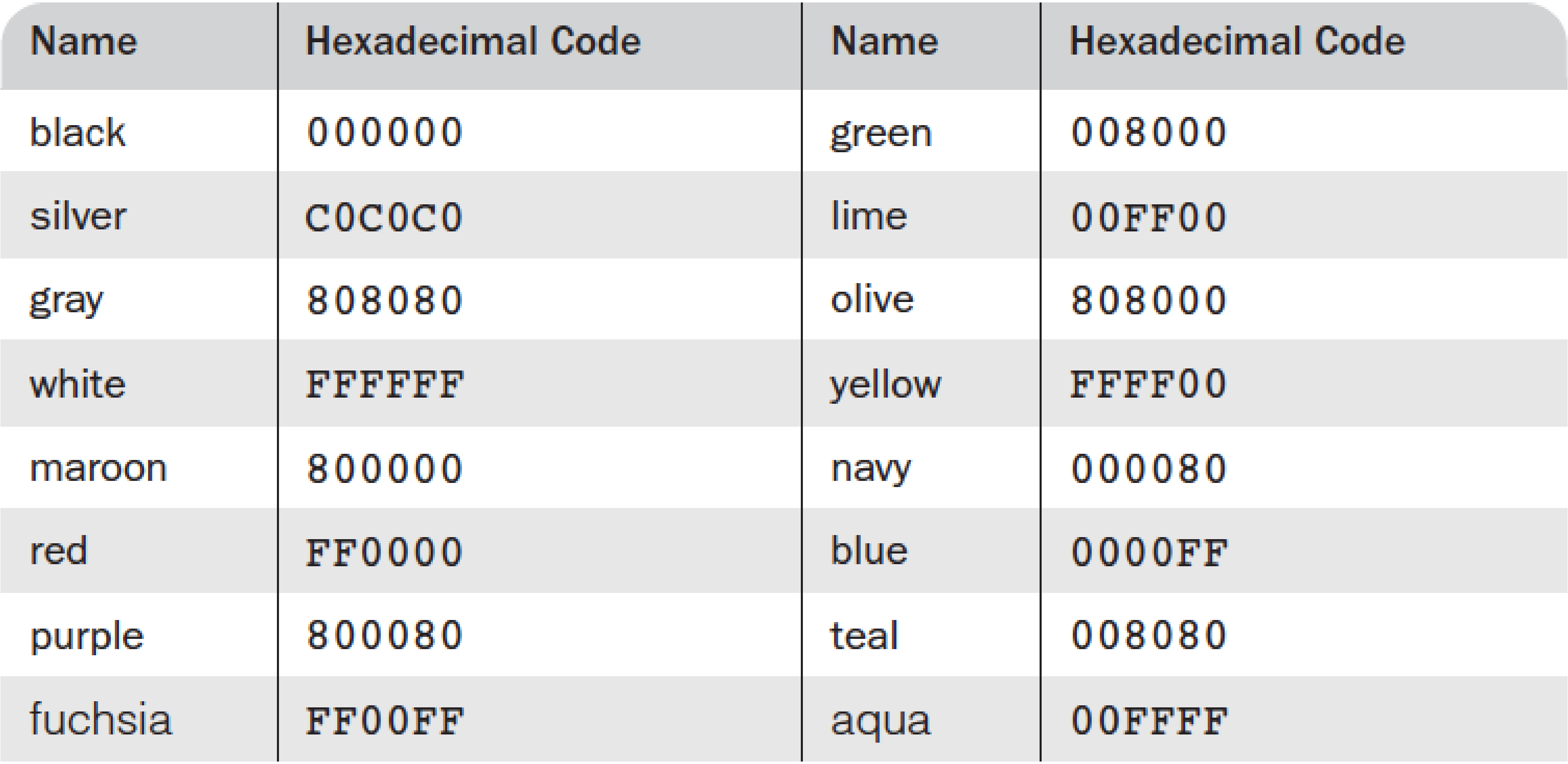


```
    <style type="text/css">
    ol {list-style-type: upper-roman;}
    ol ol {list-style-type: upper-alpha;}
    ol ol ol {list-style-type: decimal;}
    </style>

    <meta name="mydocument" content="student learning" />
    <title>nothing</title>
</head>
<body>

    <ol>

        <li>this should be upper roman</li>
        <ol>
            <li> this should be upper-alpha</li>
            <li> this should be upper-alpha</li>
            <ol>
                <li> this should be decimal</li>
                <li> this should be decimal </li>
            </ol>


        </ol>
        <li> this should be back to upper roman</li>
    </ol>
```

```
  I.  this should be upper roman
        A.  this should be upper-alpha
        B.  this should be upper-alpha
              1.  this should be decimal
              2.  this should be decimal
 II.  this should be back to upper roman
```

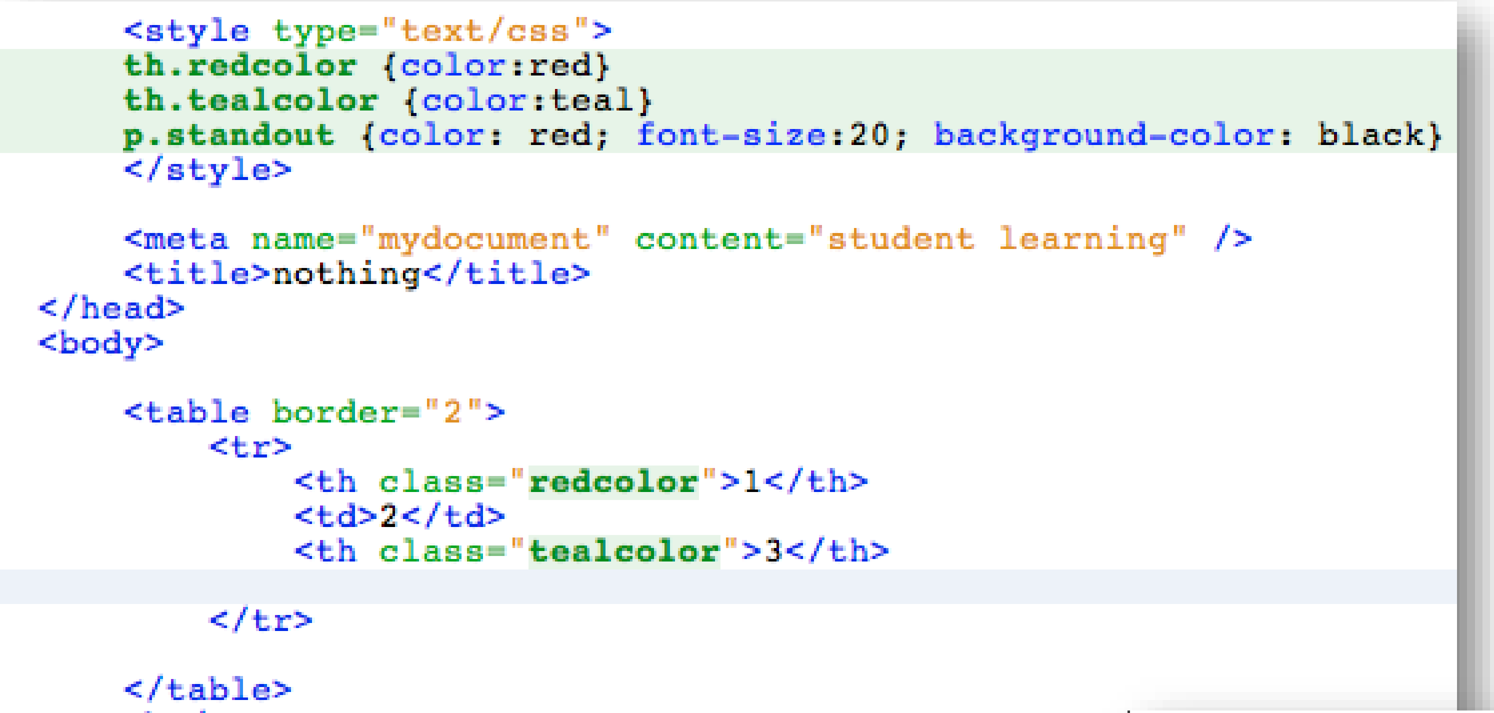
3.8 Colors

Named colors (17 colors) are displayed by all browsers and can be CSS validated

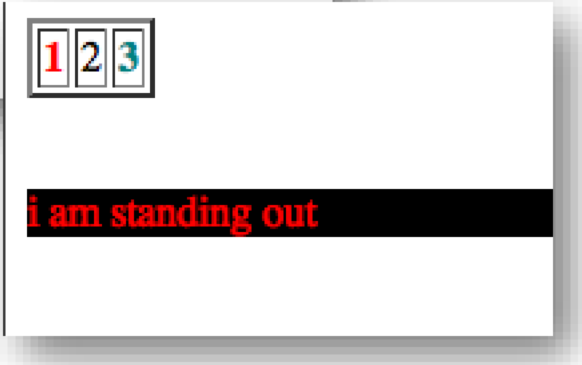


Name	Hexadecimal Code	Name	Hexadecimal Code
black	000000	green	008000
silver	C0C0C0	lime	00FF00
gray	808080	olive	808000
white	FFFFFF	yellow	FFFF00
maroon	800000	navy	000080
red	FF0000	blue	0000FF
purple	800080	teal	008080
fuchsia	FF00FF	aqua	00FFFF

3.8 Examples: color and background-color



```
    <style type="text/css">
    th.redcolor {color:red}
    th.tealcolor {color:teal}
    p.standout {color: red; font-size:20; background-color: black}
    </style>

    <meta name="mydocument" content="student learning" />
    <title>nothing</title>
</head>
<body>

    <table border="2">
        <tr>
            <th class="redcolor">1</th>
            <td>2</td>
            <th class="tealcolor">3</th>

        </tr>

    </table>
    <br/>
    <p class="standout"> i am standing out</p>
```



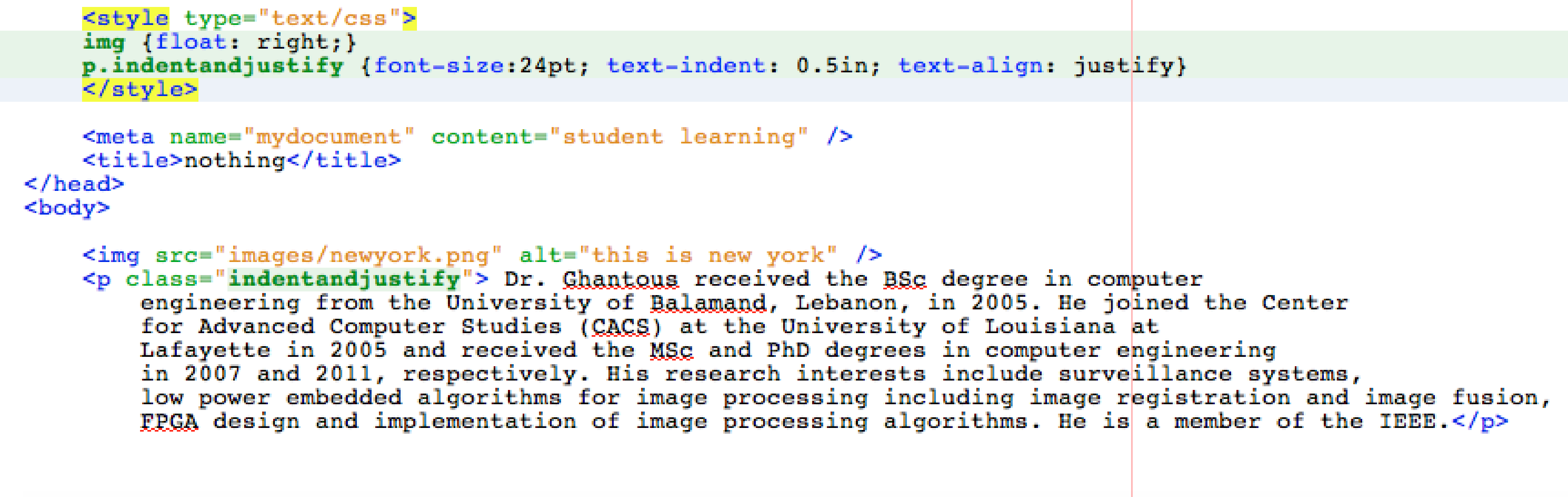
1	2	3

i am standing out

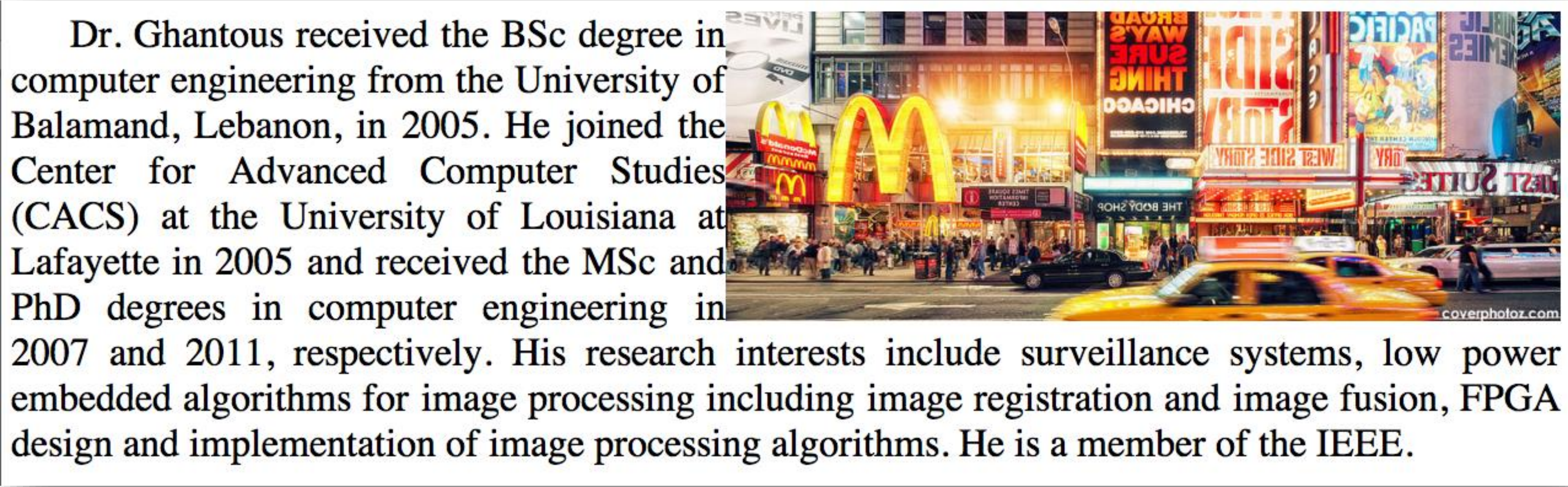
3.9 Alignment of text

- **text-indent** takes a number in inches
- **text-align** takes left, center, right and justify
- **float** is used to let text flow around some element

3.9 Alignment of text

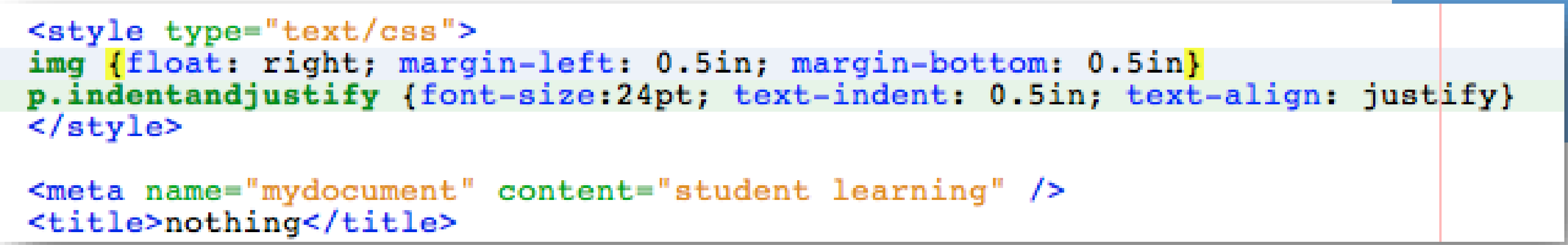


```
    <style type="text/css">
    img {float: right;}
    p.indentandjustify {font-size:24pt; text-indent: 0.5in; text-align: justify}
    </style>

    <meta name="mydocument" content="student learning" />
    <title>nothing</title>
</head>
<body>

    <img src="images/newyork.png" alt="this is new york" />
    <p class="indentandjustify"> Dr. Ghantous received the BSc degree in computer
        engineering from the University of Balamand, Lebanon, in 2005. He joined the Center
        for Advanced Computer Studies (CACS) at the University of Louisiana at
        Lafayette in 2005 and received the MSc and PhD degrees in computer engineering
        in 2007 and 2011, respectively. His research interests include surveillance systems,
        low power embedded algorithms for image processing including image registration and image fusion,
        FPGA design and implementation of image processing algorithms. He is a member of the IEEE.</p>
```

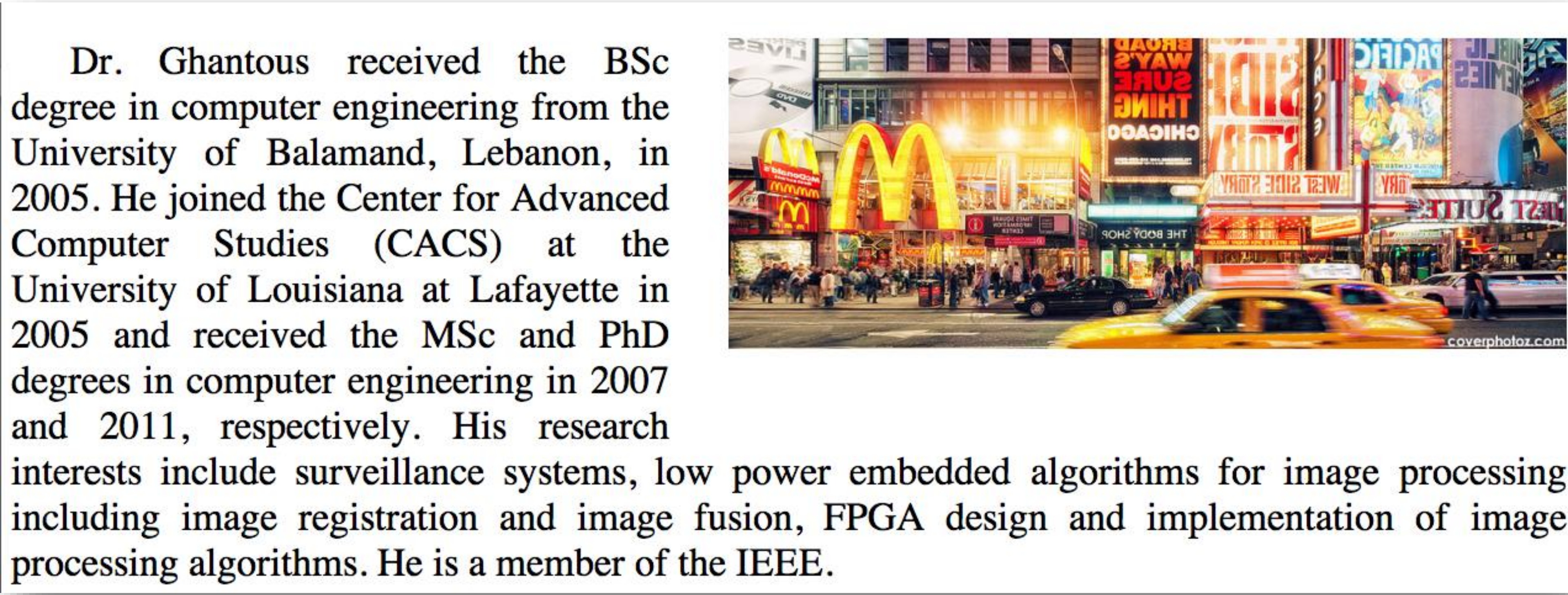


Dr. Ghantous received the BSc degree in computer engineering from the University of Balamand, Lebanon, in 2005. He joined the Center for Advanced Computer Studies (CACS) at the University of Louisiana at Lafayette in 2005 and received the MSc and PhD degrees in computer engineering in 2007 and 2011, respectively. His research interests include surveillance systems, low power embedded algorithms for image processing including image registration and image fusion, FPGA design and implementation of image processing algorithms. He is a member of the IEEE.

3.9 Alignment of text



```
<style type="text/css">
img {float: right; margin-left: 0.5in; margin-bottom: 0.5in}
p.indentandjustify {font-size:24pt; text-indent: 0.5in; text-align: justify}
</style>

<meta name="mydocument" content="student learning" />
<title>nothing</title>
```



Dr. Ghantous received the BSc degree in computer engineering from the University of Balamand, Lebanon, in 2005. He joined the Center for Advanced Computer Studies (CACS) at the University of Louisiana at Lafayette in 2005 and received the MSc and PhD degrees in computer engineering in 2007 and 2011, respectively. His research interests include surveillance systems, low power embedded algorithms for image processing including image registration and image fusion, FPGA design and implementation of image processing algorithms. He is a member of the IEEE.

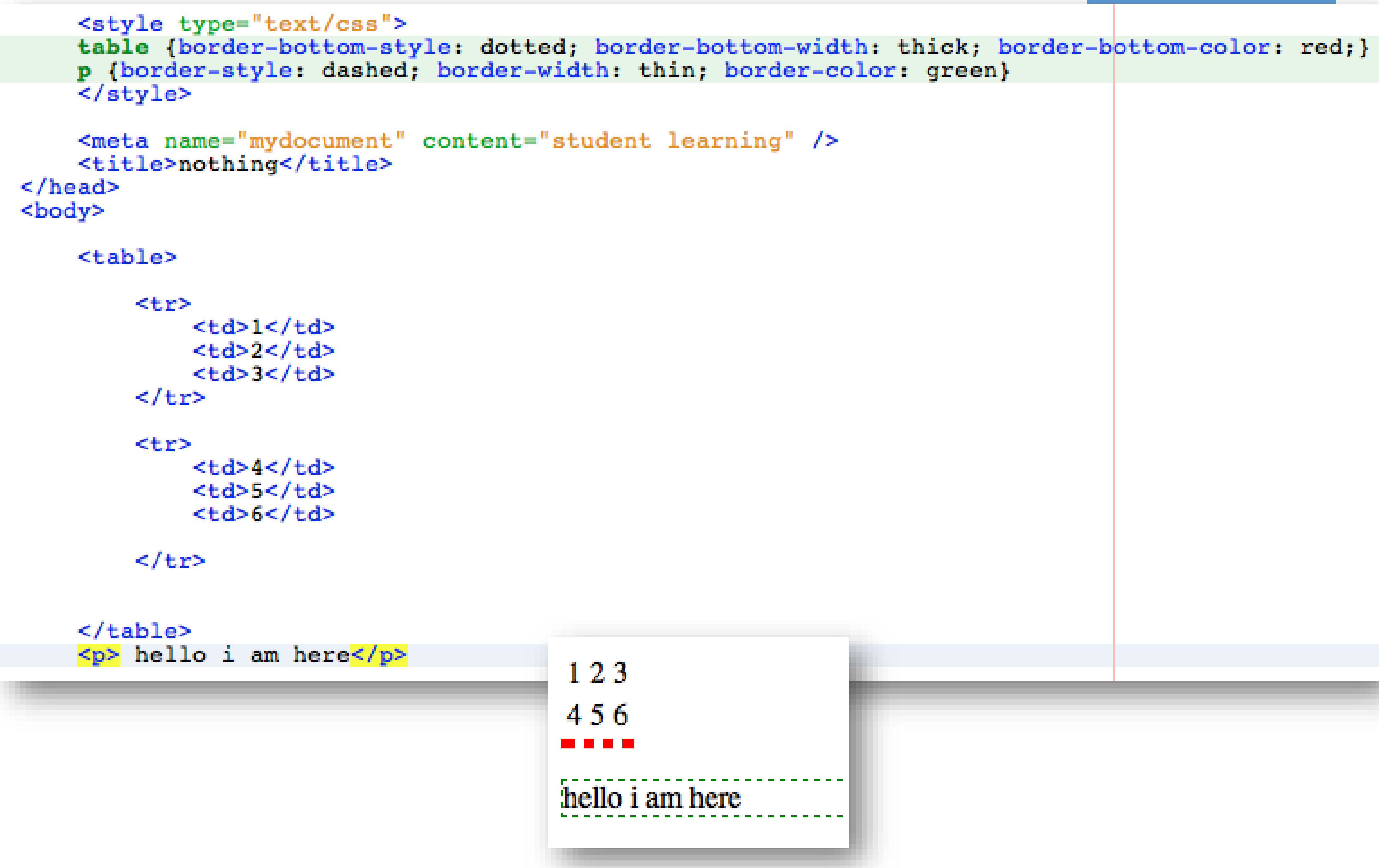
3.9 Borders

Border-style, border-top-style, border-bottom-style, border-left-style, …

Border-width, border-top-width, …

Border-color, …

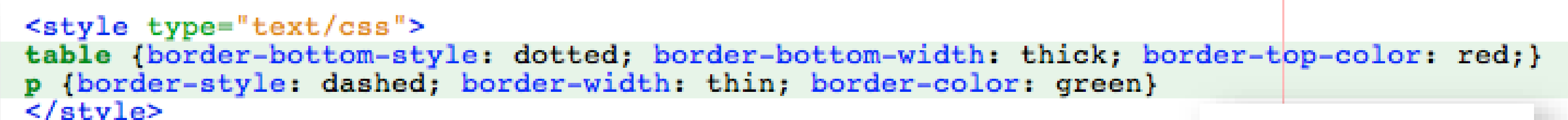
3.9 Borders



```
    <style type="text/css">
    table {border-bottom-style: dotted; border-bottom-width: thick; border-bottom-color: red;}
    p {border-style: dashed; border-width: thin; border-color: green}
    </style>

    <meta name="mydocument" content="student learning" />
    <title>nothing</title>
</head>
<body>

    <table>

        <tr>
            <td>1</td>
            <td>2</td>
            <td>3</td>
        </tr>

        <tr>
            <td>4</td>
            <td>5</td>
            <td>6</td>

        </tr>


    </table>
    <p> hello i am here</p>
```

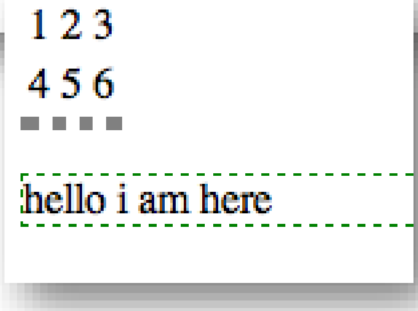
```
1 2 3
4 5 6

hello i am here
```

3.9 Borders



```
<style type="text/css">
table {border-bottom-style: dotted; border-bottom-width: thick; border-top-color: red;}
p {border-style: dashed; border-width: thin; border-color: green}
</style>
```

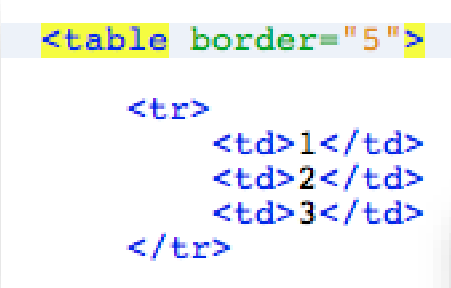
Top border color only is specified: no style is defined, and no border is specified either inside the table tag <table>, so NO top border appears



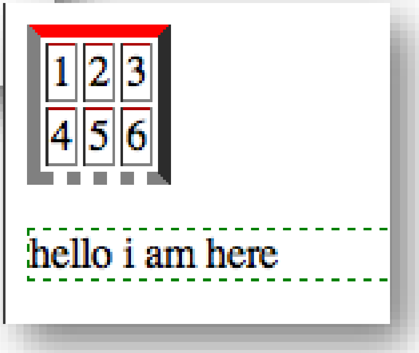
1 2 3

4 5 6

hello i am here



```
<table border="5">

    <tr>
        <td>1</td>
        <td>2</td>
        <td>3</td>
    </tr>
```



1 2 3

4 5 6

hello i am here

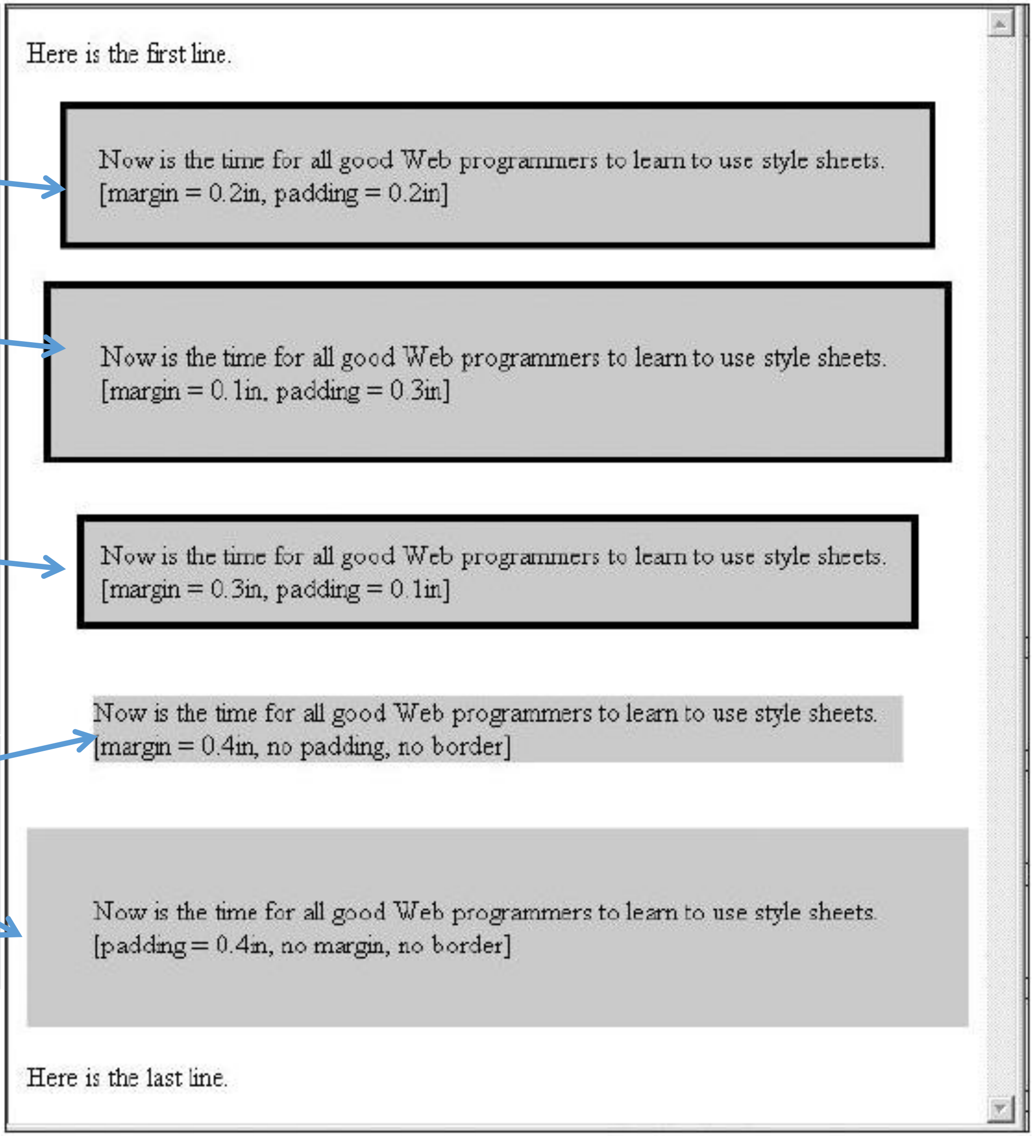
3.9 Margins and Padding

Margin-left, right, …

Padding-left, right, …

3.9 Margins and Padding: example

```
p.one   {margin: 0.2in;
         padding: 0.2in;
         background-color: #C0C0C0;
         border-style: solid;
        }
p.two   {margin: 0.1in;
         padding: 0.3in;
         background-color: #C0C0C0;
         border-style: solid;
        }
p.three {margin: 0.3in;
         padding: 0.1in;
         background-color: #C0C0C0;
         border-style: solid;
        }
p.four  {margin:0.4in;
         background-color: #C0C0C0;}
p.five  {padding: 0.4in;
         background-color: #C0C0C0;
        }
```



Here is the first line.

Now is the time for all good Web programmers to learn to use style sheets. [margin = 0.2in, padding = 0.2in]

Now is the time for all good Web programmers to learn to use style sheets. [margin = 0.1in, padding = 0.3in]

Now is the time for all good Web programmers to learn to use style sheets. [margin = 0.3in, padding = 0.1in]

Now is the time for all good Web programmers to learn to use style sheets. [margin = 0.4in, no padding, no border]

Now is the time for all good Web programmers to learn to use style sheets. [padding = 0.4in, no margin, no border]

Here is the last line.

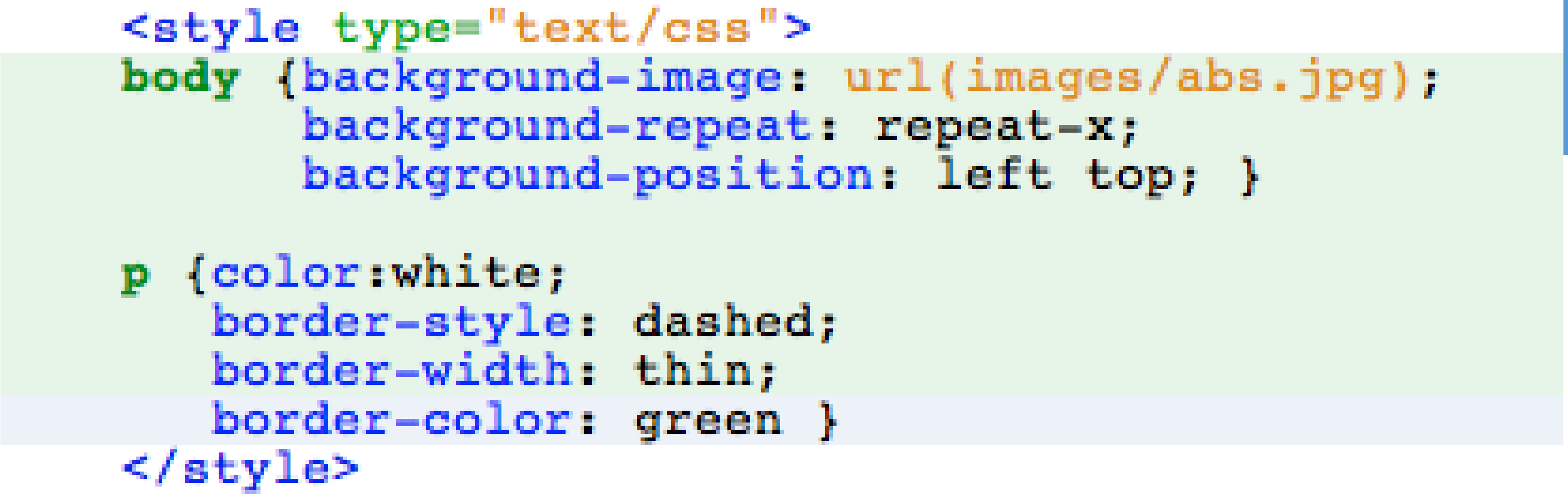
3.9 Background-image

```
body { background-image:
url([imageName]) }
```

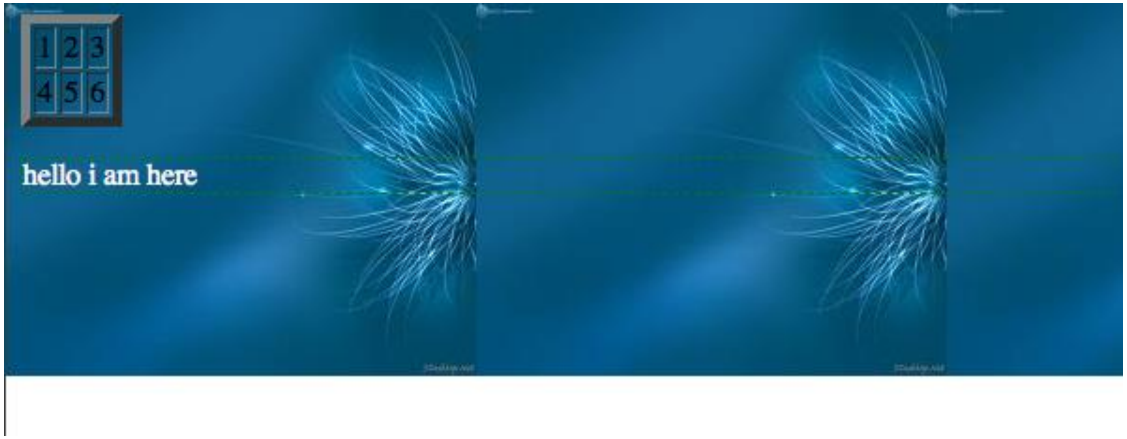
background-repeat: repeat, repeat-x, repeat-y, no-repeat

Background-position: left, right, center, bottom, bottom right, center top. (horizontal then vertical, center is default, so left will be automatically left center)

3.9 Background-image



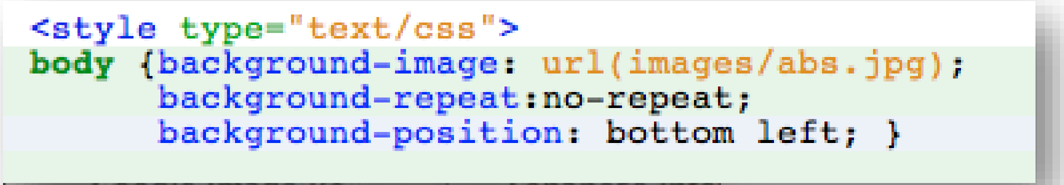
```
<style type="text/css">
body {background-image: url(images/abs.jpg);
      background-repeat: repeat-x;
      background-position: left top; }

p {color:white;
   border-style: dashed;
   border-width: thin;
   border-color: green }
</style>
```



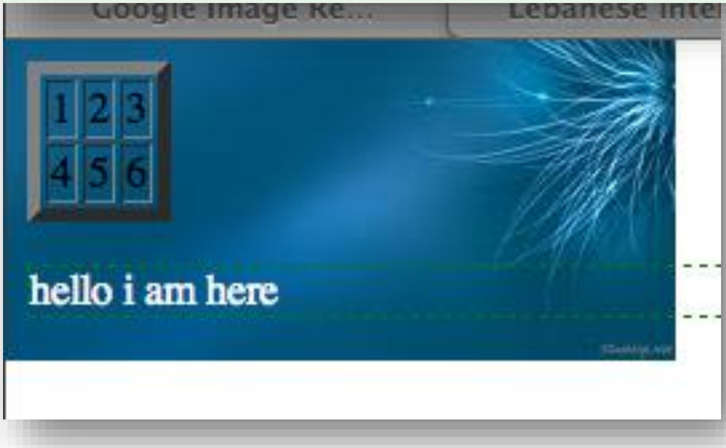
hello i am here

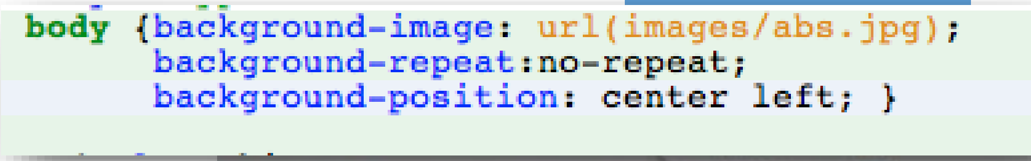
When repeat is used, no meaning for position

3.9 Background-image

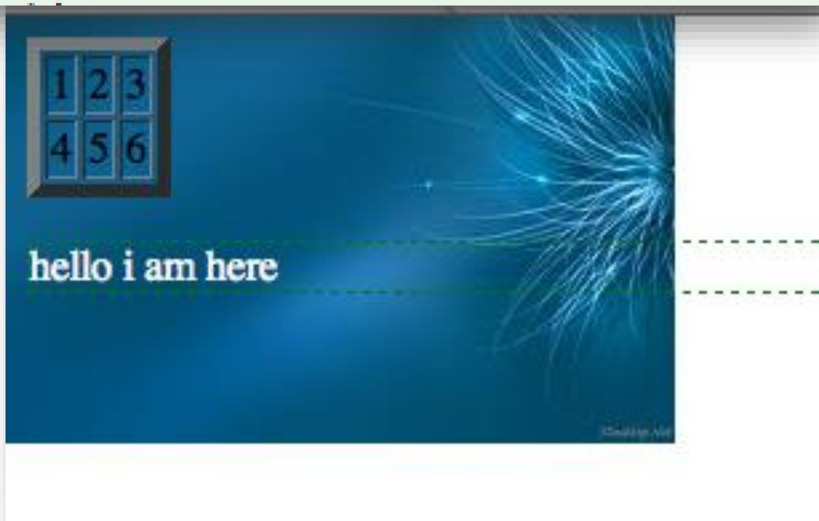


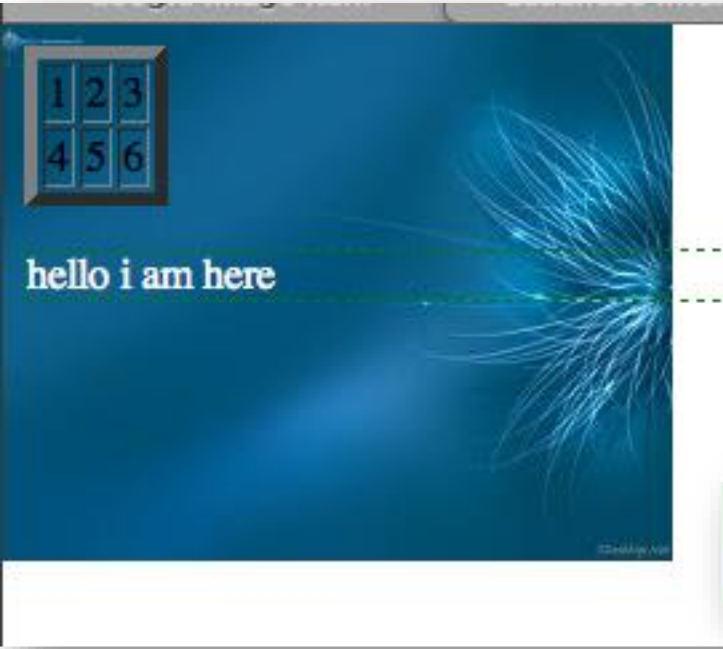
```
<style type="text/css">
body {background-image: url(images/abs.jpg);
      background-repeat:no-repeat;
      background-position: bottom left; }
```

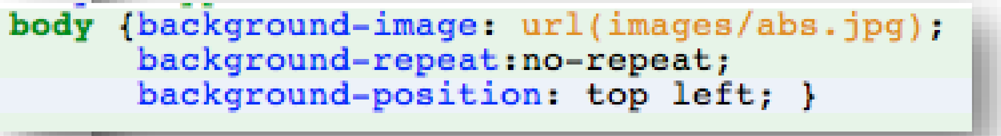




```
body {background-image: url(images/abs.jpg);
      background-repeat:no-repeat;
      background-position: center left; }
```





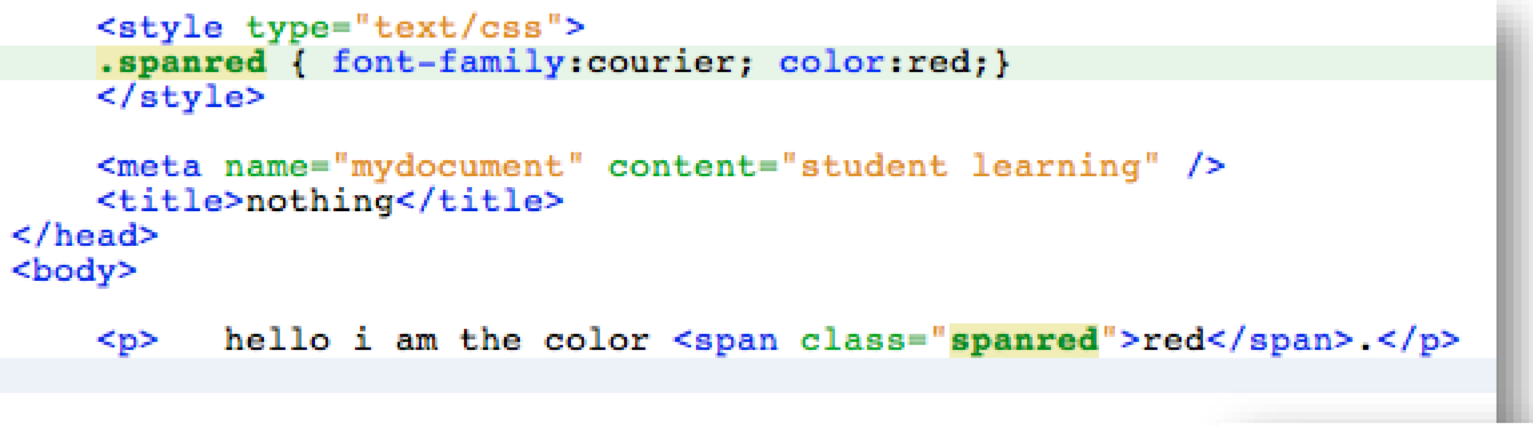


```
body {background-image: url(images/abs.jpg);
      background-repeat:no-repeat;
      background-position: top left; }
```

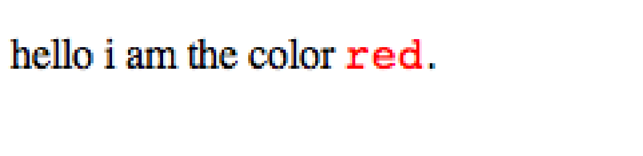
3.9 span and div

span is used to apply a style to 1 word or a section inside a paragraph

div is used to apply style to a number of paragraphs without the need to specify the style for each

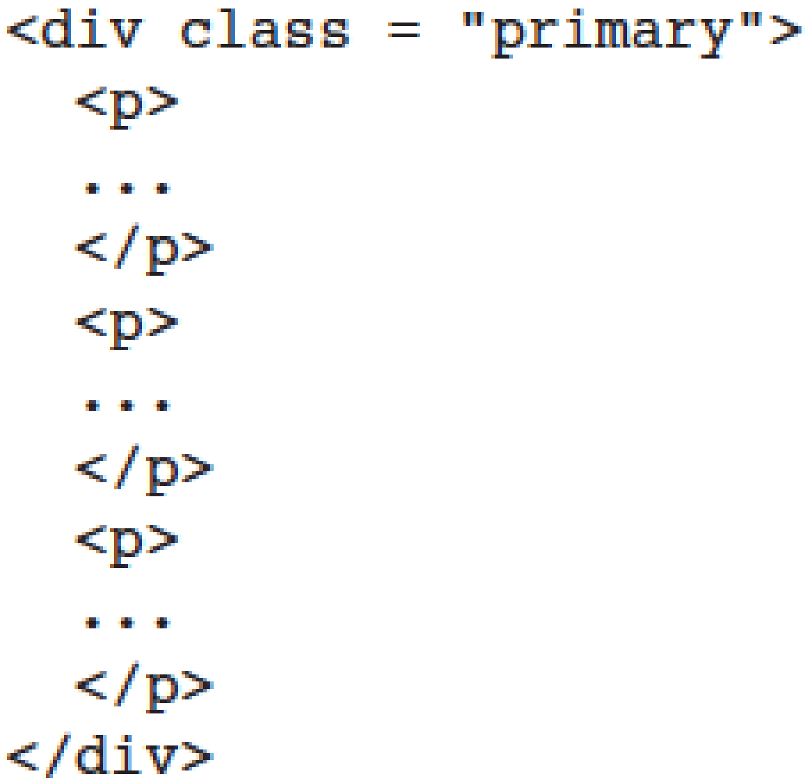


```
    <style type="text/css">
    .spanred { font-family:courier; color:red;}
    </style>

    <meta name="mydocument" content="student learning" />
    <title>nothing</title>
</head>
<body>

    <p>   hello i am the color <span class="spanred">red</span>.</p>
```



hello i am the color red.

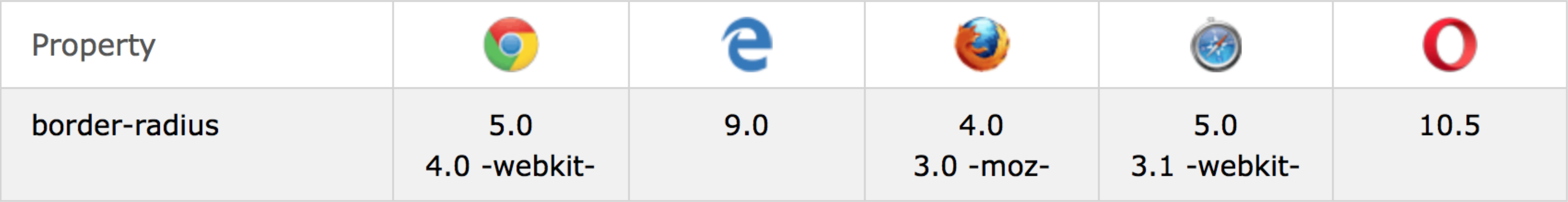
3.9 span and div



```
<div class = "primary">
  <p>
  ...
  </p>
  <p>
  ...
  </p>
  <p>
  ...
  </p>
</div>
```

Rounded corners

- With the CSS3 **`border-radius`** property, you can give any element "rounded corners".

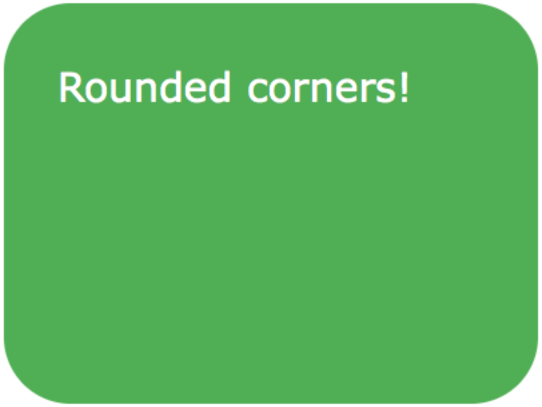


Property		e			O
border-radius	5.0 4.0 -webkit-	9.0	4.0 3.0 -moz-	5.0 3.1 -webkit-	10.5

- Some CSS3 properties were not available until a specific version of the browser.
- For them to work on all versions, you need to add a prefix such as –webkit- or –moz-

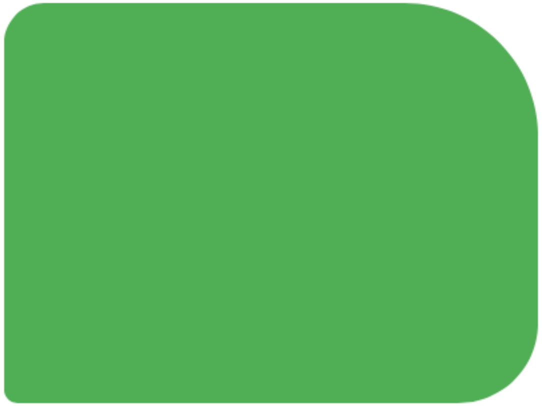
Rounded corners

```
#rcorners1 {
    border-radius: 25px;
    background: #73AD21;
    padding: 20px;
    width: 200px;
    height: 150px;
}
```



Four values: first value applies to top-left, second value applies to top-right, third value applies to bottom-right, and fourth value applies to bottom-left corner

```
#rcorners4 {
    border-radius: 15px 50px 30px 5px;
    background: #73AD21;
    padding: 20px;
    width: 200px;
    height: 150px;
}
```

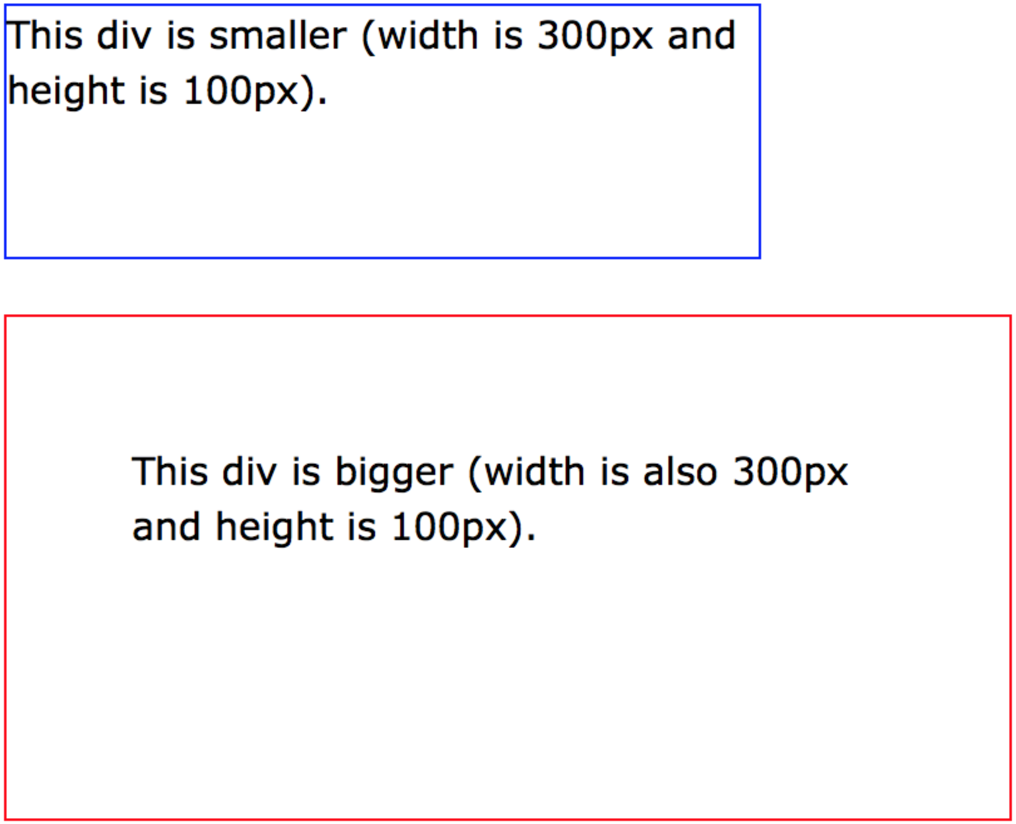


`box-sizing` property

- By default, the width and height of an element is calculated like this:
- width + padding + border = actual width of an element
- height + padding + border = actual height of an element
- This means: When you set the width/height of an element, the element often appear bigger than you have set (because the element's border and padding are added to the element's specified width/height).

Example: same width/height

```
.div1 {
    width: 300px;
    height: 100px;
    border: 1px solid blue;
}

.div2 {
    width: 300px;
    height: 100px;
    padding: 50px;
    border: 1px solid red;
}
```



This div is smaller (width is 300px and height is 100px).

This div is bigger (width is also 300px and height is 100px).

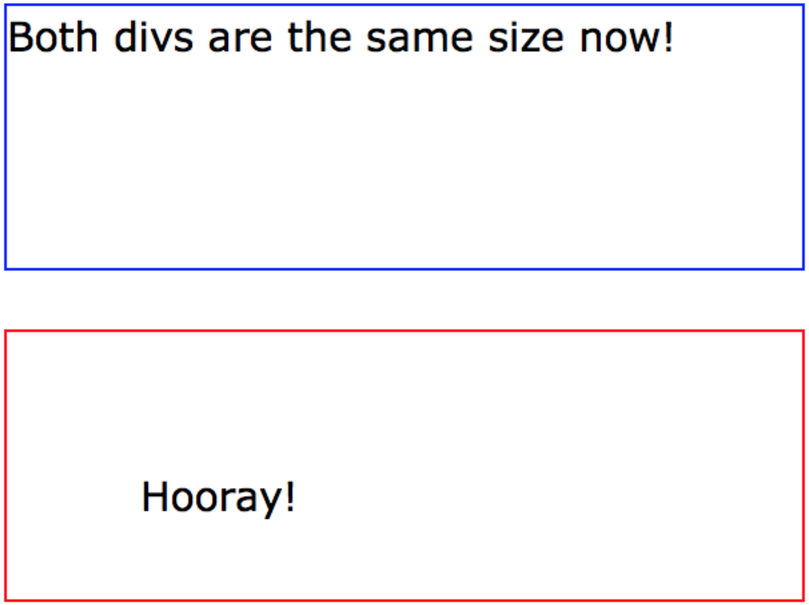
Solution:

- Use the box-sizing property

- The CSS3 box-sizing property allows us to include the padding and border in an element's total width and height.

- If you set box-sizing: border-box; on an element padding and border are included in the width and height:

```
.div1 {
    width: 300px;
    height: 100px;
    border: 1px solid blue;
    box-sizing: border-box;
}
.div2 {
    width: 300px;
    height: 100px;
    padding: 50px;
    border: 1px solid red;
    box-sizing: border-box;}
```



Both divs are the same size now!

Hooray!

Tip:

- Since we want this for all elements, it's good to do this

```
* {
    box-sizing: border-box;
}
```

Another Tip:

- Sometimes, we need to apply more than 1 class on an element. This is ok!

```
<div class=”class1 class2”>
```

Reminder: write all your css in one or more file with **.css** extension

- Add them to any html you want

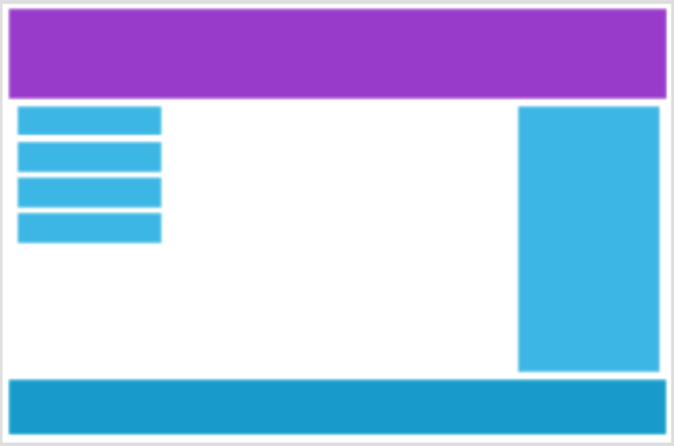
```
<head>
<link rel="stylesheet"
      type="text/css"
      href=“YourFileName.css">
</head>
```

Part 2

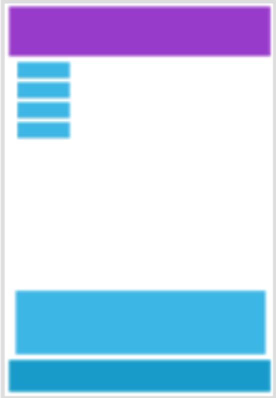
Responsive web design

Responsive web design

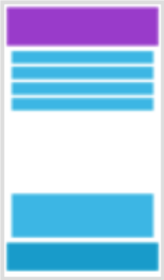
- Responsive web design makes your web page look good on all devices.
- Responsive web design uses only HTML and CSS.
- Responsive web design is not a program or a JavaScript.
- Web pages should not leave out information to fit smaller devices, but rather adapt its content to fit any device



Desktop



Tablet



Phone

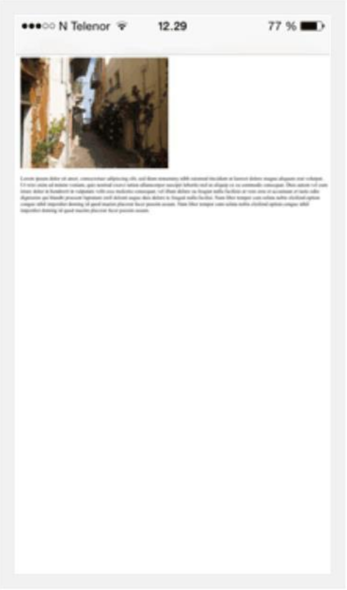
Responsive web design

The Viewport

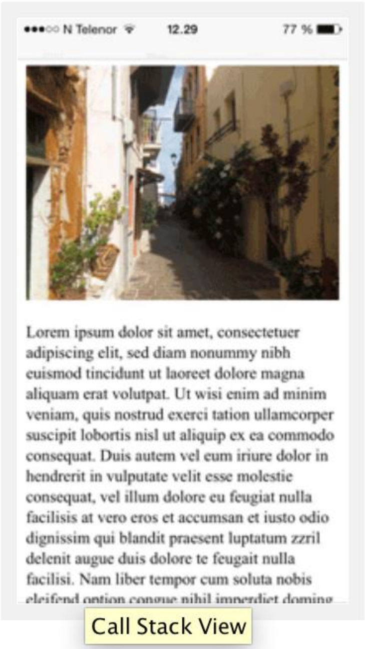
- The viewport is the user's visible area of a web page.
- The viewport varies with the device, and will be smaller on a mobile phone than on a computer screen.
- Add this to the <head> section

```
<meta name="viewport"
      content="width=device-width, initial-scale=1.0">
```

- When specifying sizes in your CSS, always use relative sizes such as *em* or percentages like 50%, 100%
- **Now you can also use vw, for example 1.5 vw for font sizes. It will scale by itself.**



Without the viewport meta tag

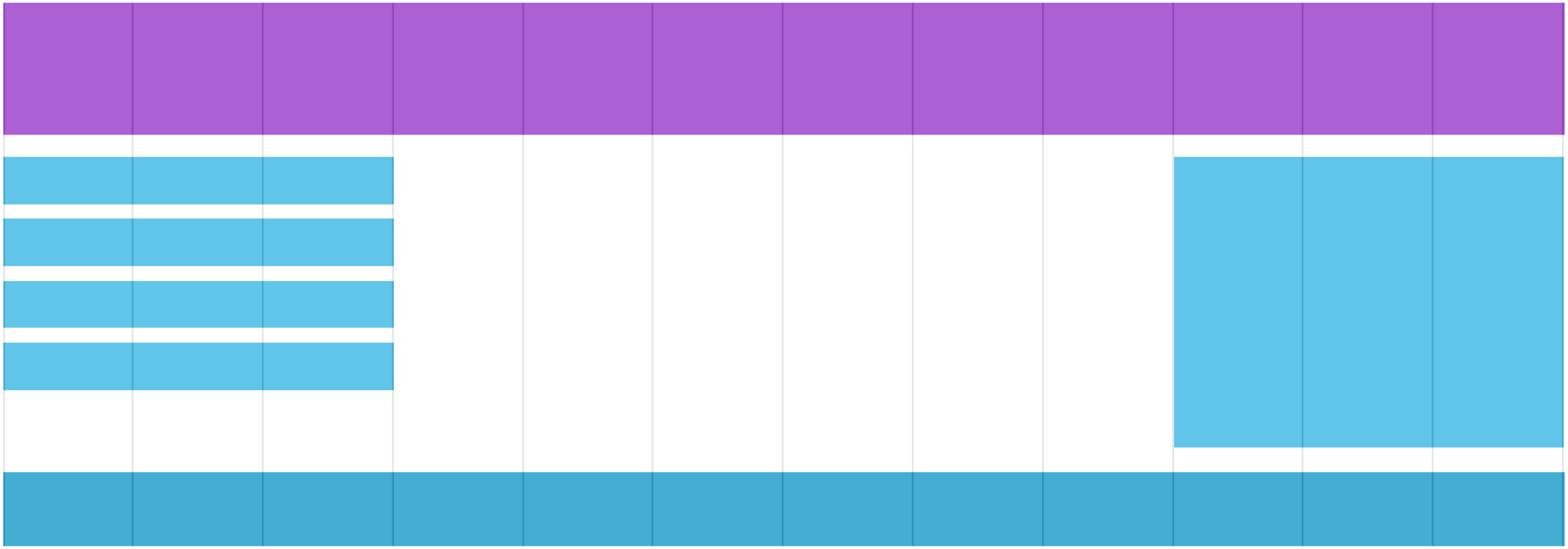


Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming

With the viewport meta tag

Grid view

- Using a grid-view is very helpful when designing web pages. It makes it easier to place elements on the page.
- A responsive grid-view often has **12 columns**, and has a total width of 100%, and will shrink and expand as you resize the browser window.



Full example

- Check the responsive webpage on NetBeans example.

- Main ideas: columns, @media condition to add a breakpoint, and ::after

- Try to resize the browser.

The columns

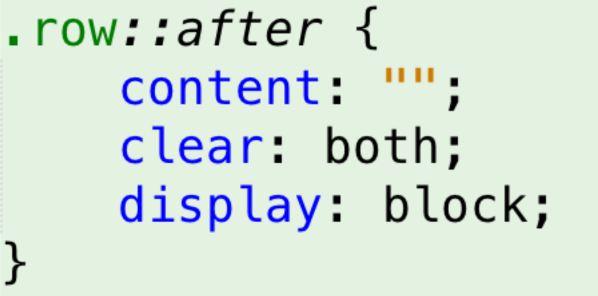
All classes that start with col-

```
[class*="col-"]
{    float: left;    padding: 15px;}


/* For mobile phones: */
[class*="col-"] {    width: 100%;}

@media only screen and (min-width: 768px)
{    /* For desktop: */
.col-1 {width: 8.33%;}
.col-2 {width: 16.66%;}
.col-3 {width: 25%;}
.col-4 {width: 33.33%;}
.col-5 {width: 41.66%;}
.col-6 {width: 50%;}
.col-7 {width: 58.33%;}
.col-8 {width: 66.66%;}
.col-9 {width: 75%;}
.col-10 {width: 83.33%;}
.col-11 {width: 91.66%;}
.col-12 {width: 100%;}}
```

@media only screen means condition if the width is larger than 768 then the columns width not 100% anymore but we have 12 columns per row, each with a specified width

But where to add these divs with **col-** classes?

- we do that Inside a <**`div`**> with class ***`row`*** that we will create.

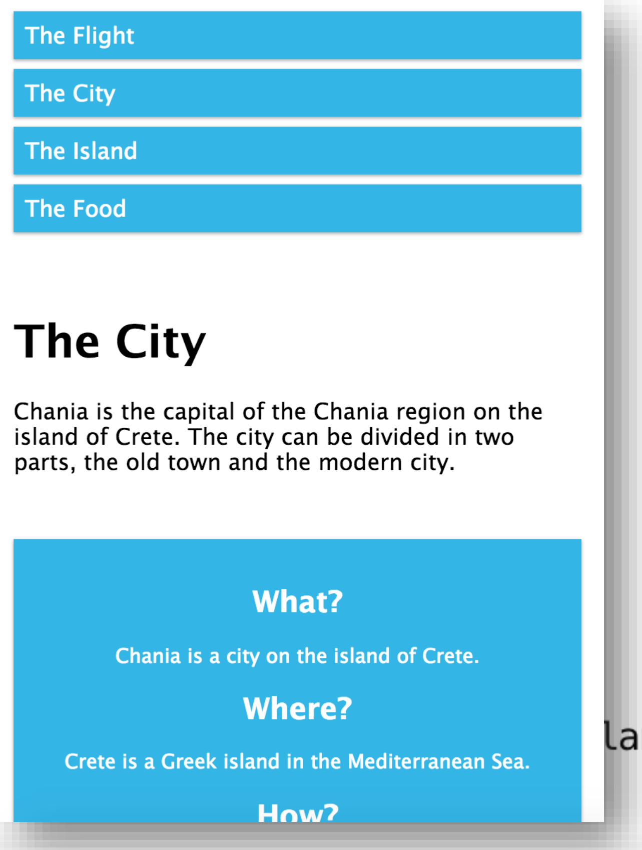


```
.row::after {
    content: "";
    clear: both;
    display: block;
}
```

After our header, we can embed everything into a <div> with row class. The **`::after`** pseudo selector means that after each element with class row:

- Content is empty.
- No floating to the right or left of it. (clear both sides)
- And display block means if another element comes after it, it will be on a new line.

```
<div class="row">
<div class="col-3 menu">
<ul>
<li>The Flight</li>
<li>The City</li>
<li>The Island</li>
<li>The Food</li>
</ul>
</div>
```



The Flight

The City

The Island

The Food

The City

Chania is the capital of the Chania region on the island of Crete. The city can be divided in two parts, the old town and the modern city.

What?

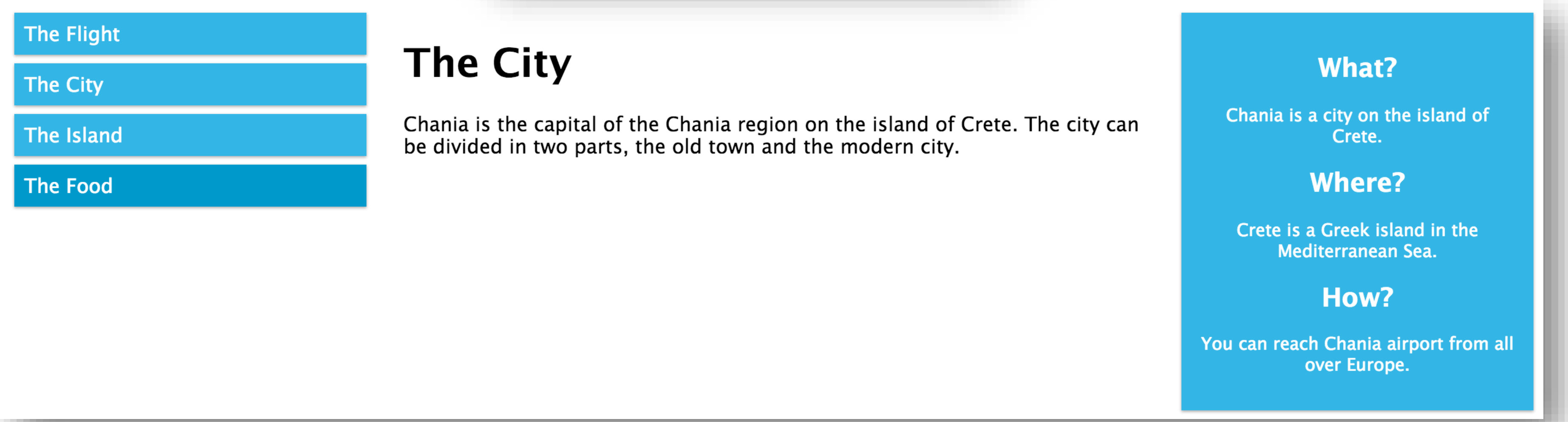
Chania is a city on the island of Crete.

Where?

Crete is a Greek island in the Mediterranean Sea.

How?

```
<div class="col-6">
<h1>The City</h1>
<p>Chania is the capital of                             land of
</div>
```



The Flight

The City

The Island

The Food

The City

Chania is the capital of the Chania region on the island of Crete. The city can be divided in two parts, the old town and the modern city.

What?

Chania is a city on the island of Crete.

Where?

Crete is a Greek island in the Mediterranean Sea.

How?

You can reach Chania airport from all over Europe.

```
</div>
```

Bootstrap

- popular framework is Bootstrap, it uses HTML, CSS and jQuery to make responsive web pages.

- It hides some CSS details from you like the previous example

To use bootstrap, include this in your page inside <head>

```html
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">

<!-- jQuery library -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">
</script>

<!-- Popper JS -->
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js">
</script>

<!-- Latest compiled JavaScript -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js">
</script>
```

Classes

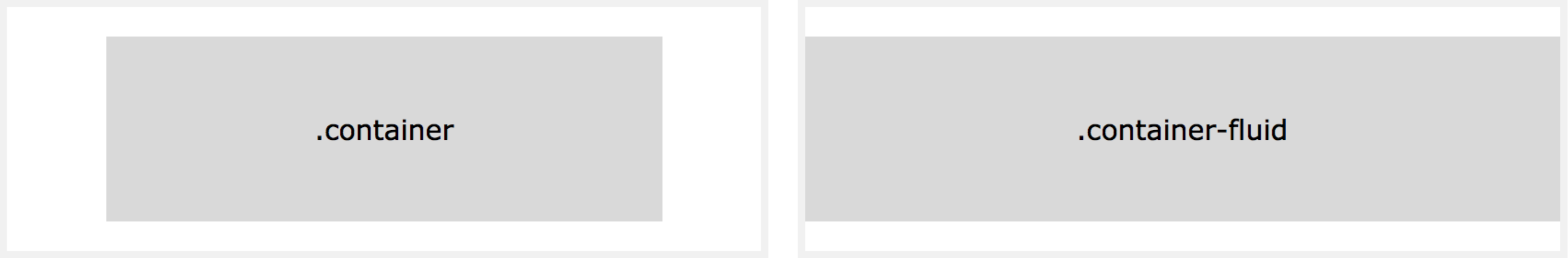
- Think about bootsrap as a CSS library
- this means it has classes. But this time, they have written the classes for us.

- One of the most famous classes are container classes ***inside which*** we place all our components: the `.container` class

container classes

Bootstrap 4 also requires a containing element to wrap site contents.
There are two container classes to choose from:

1. The `.container` class provides a responsive **fixed width container**
2. The `.container-fluid` class provides a **full width container**, spanning the entire width of the viewport



Examples

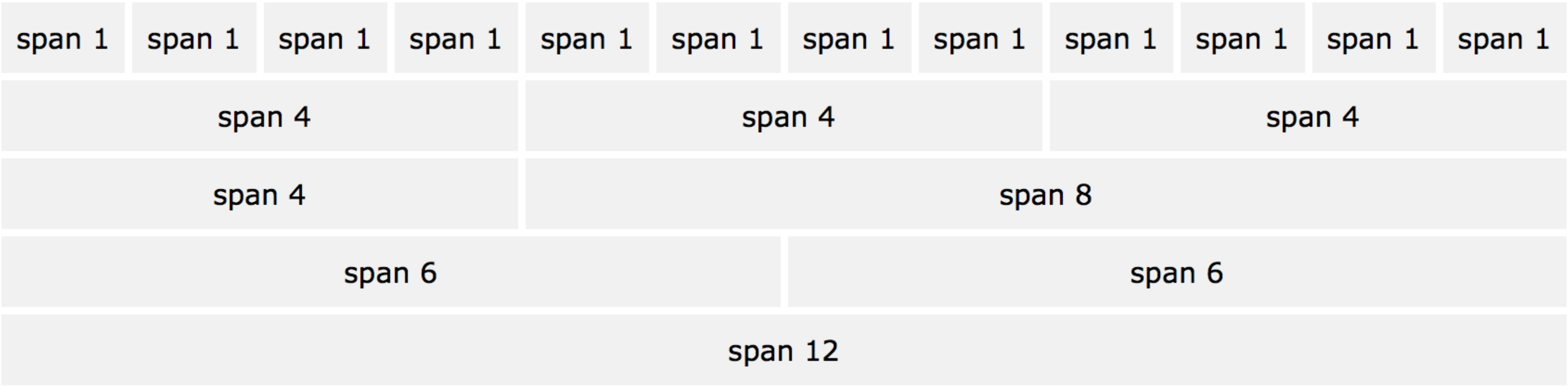
```
<div class="container">
  <h1>My First Bootstrap Page</h1>
  <p>This is some text.</p>
</div>
```

OR

```
<div class="container-fluid">
  <h1>My First Bootstrap Page</h1>
  <p>This is some text.</p>
</div>
```

Bootstrap **Grids**

- Bootstrap's grid system is built with flexbox and allows up to **12 columns** across the page.
- If you do not want to use all 12 columns individually, you can group the columns together to create wider columns:



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

Grid classes

The Bootstrap 4 grid system has five classes:

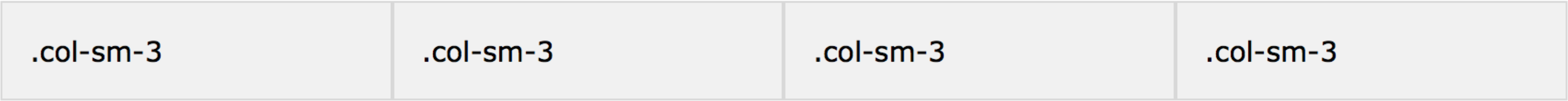
- **`.col-`** (extra small devices - screen width less than 576px)
- **`.col-sm-`** (small devices - screen width equal to or greater than 576px)
- **`.col-md-`** (medium devices - screen width equal to or greater than 768px)
- **`.col-lg-`** (large devices - screen width equal to or greater than 992px)
- **`.col-xl-`** (xlarge devices - screen width equal to or greater than 1200px)

how to create a grid?

- Create a row `(<div class="row">` inside a div with class `container`).
- Then, add the desired number of columns (tags with appropriate .col-*-* classes).
- The first star (*) represents the responsiveness: sm, md, lg or xl
- while the second star (*) represents a number, **which should add up to 12 for each row**.

Example

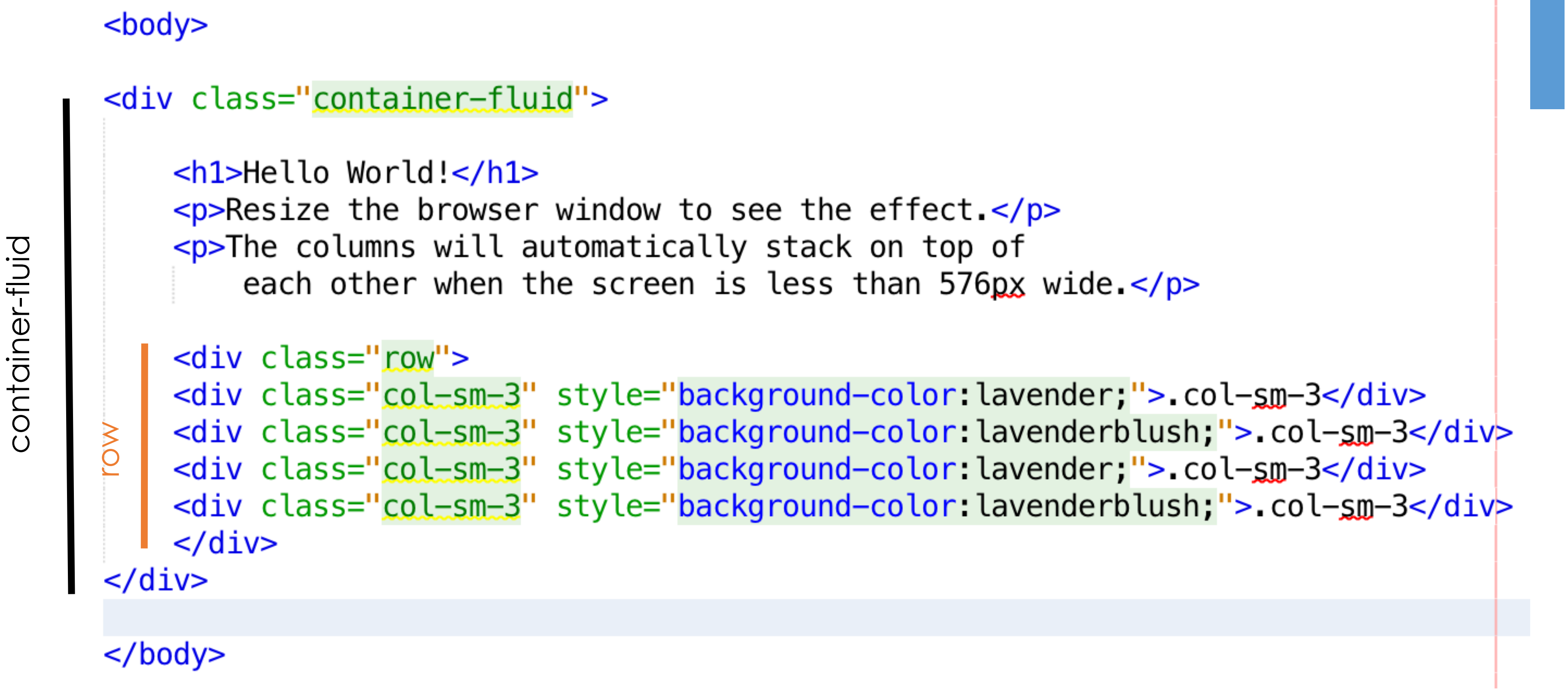
```
<div class="row">
  <div class="col-sm-3">.col-sm-3</div>
  <div class="col-sm-3">.col-sm-3</div>
  <div class="col-sm-3">.col-sm-3</div>
  <div class="col-sm-3">.col-sm-3</div>
</div>
```



.col-sm-3	.col-sm-3	.col-sm-3	.col-sm-3

when you resize the browser or open on phone, these will stack up on top of each other

<head> part is hidden here containing the CDN

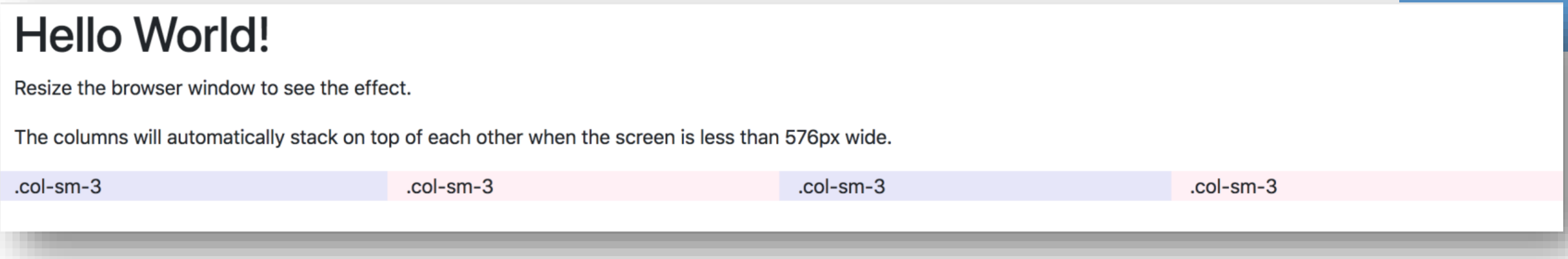


```
<body>

<div class="container-fluid">

    <h1>Hello World!</h1>
    <p>Resize the browser window to see the effect.</p>
    <p>The columns will automatically stack on top of
        each other when the screen is less than 576px wide.</p>

    <div class="row">
    <div class="col-sm-3" style="background-color:lavender;">.col-sm-3</div>
    <div class="col-sm-3" style="background-color:lavenderblush;">.col-sm-3</div>
    <div class="col-sm-3" style="background-color:lavender;">.col-sm-3</div>
    <div class="col-sm-3" style="background-color:lavenderblush;">.col-sm-3</div>
    </div>
</div>

</body>
```

Notice the col-sm-3, since we have four of them, means their total is equal to 12

when you run this code

desktop/laptop



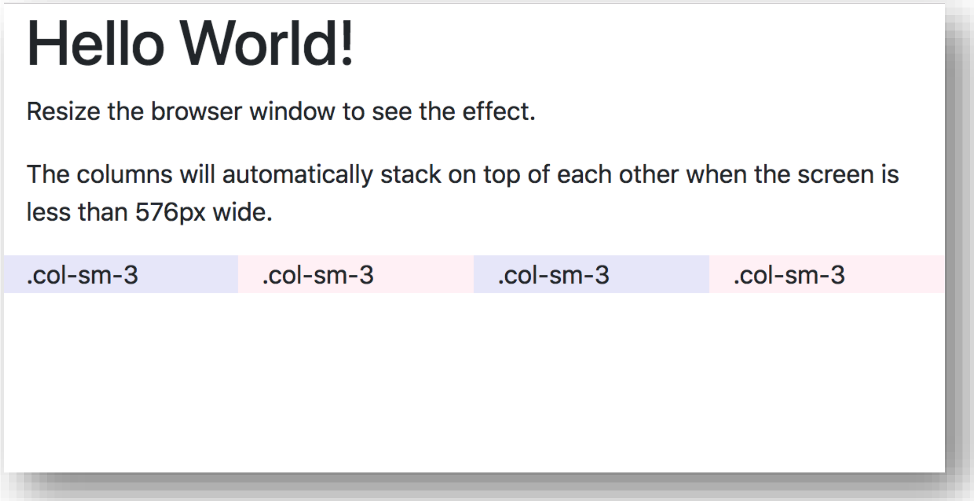
Hello World!

Resize the browser window to see the effect.

The columns will automatically stack on top of each other when the screen is less than 576px wide.

.col-sm-3 .col-sm-3 .col-sm-3 .col-sm-3

Tablet



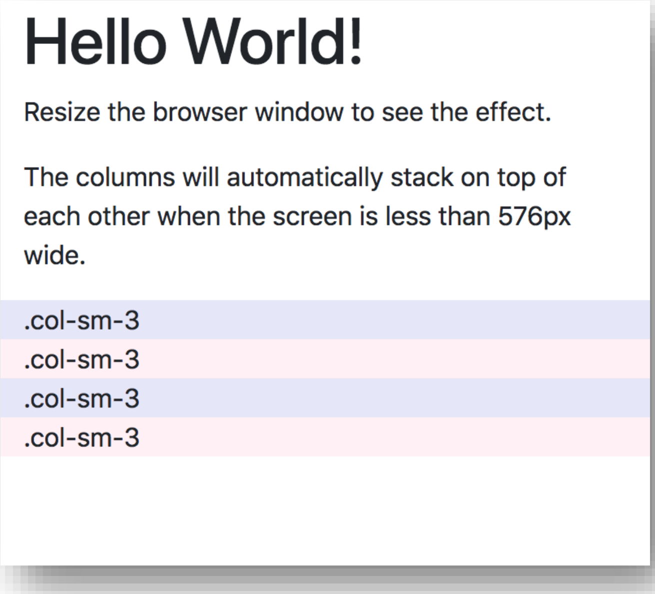
Hello World!

Resize the browser window to see the effect.

The columns will automatically stack on top of each other when the screen is less than 576px wide.

.col-sm-3 .col-sm-3 .col-sm-3 .col-sm-3

Mobile



Hello World!

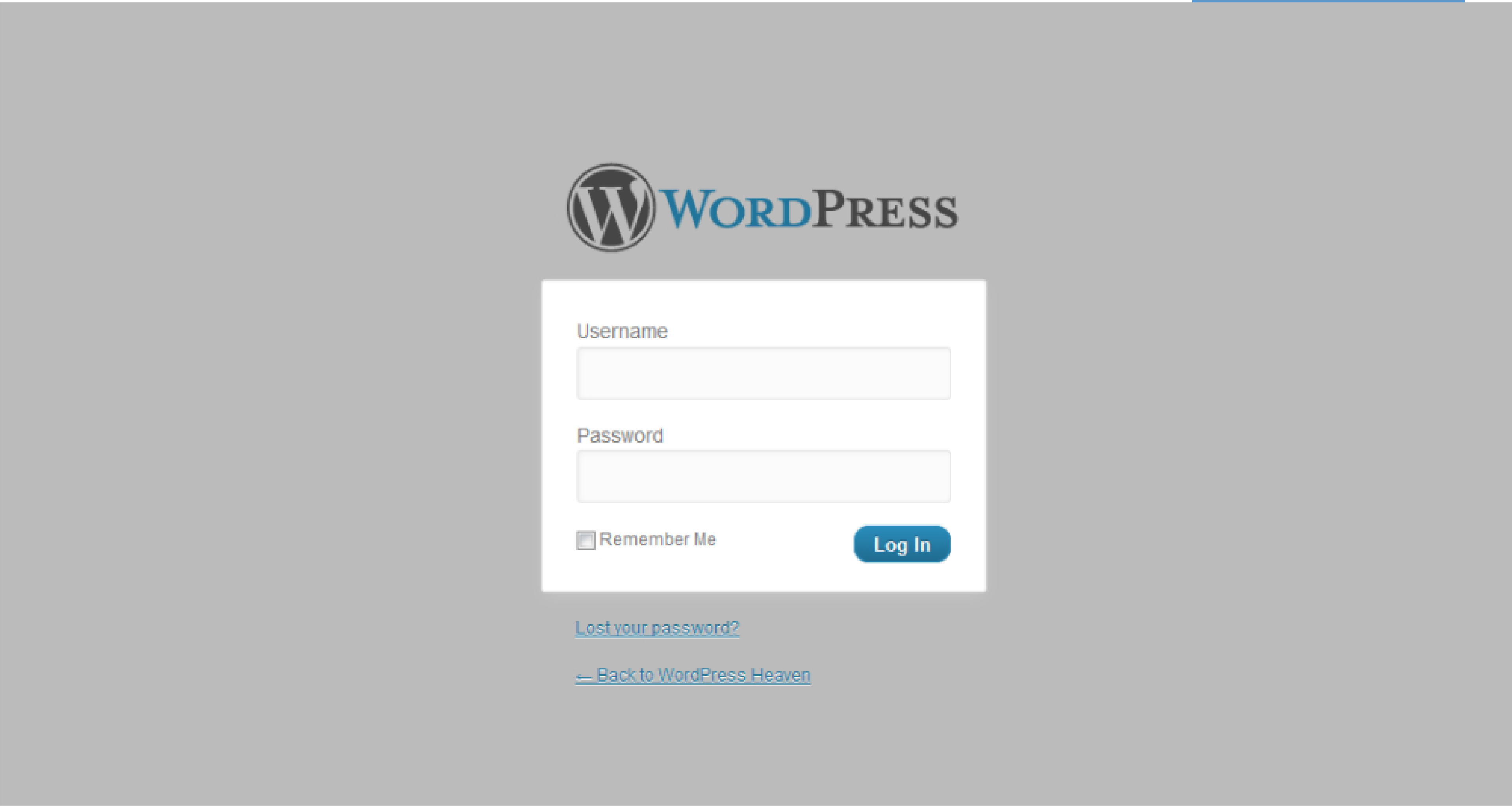
Resize the browser window to see the effect.

The columns will automatically stack on top of each other when the screen is less than 576px wide.

.col-sm-3
.col-sm-3
.col-sm-3
.col-sm-3

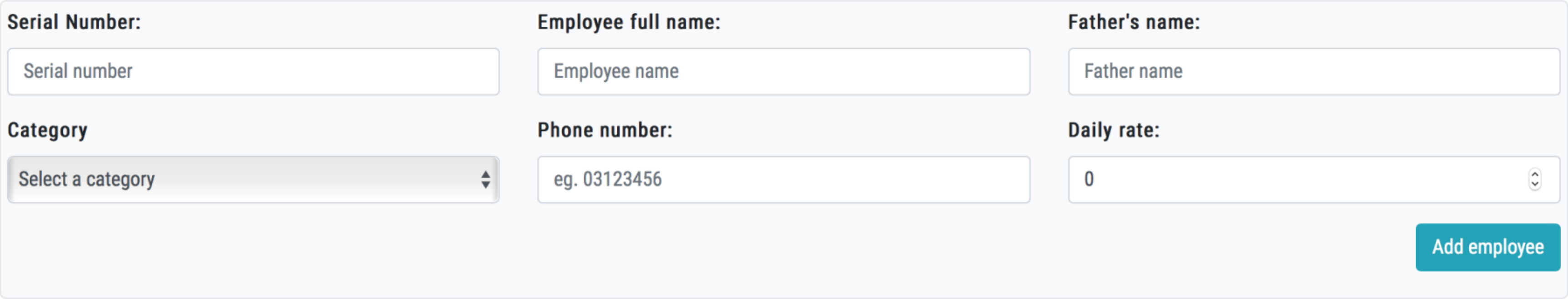
check the **optional** bootstrap lecture for other classes concerning colors, tables, borders, and other stuff

Exercise 1


WordPress
Username
Password
□ Remember Me
Log In
Lost your password?
← Back to WordPress Heaven

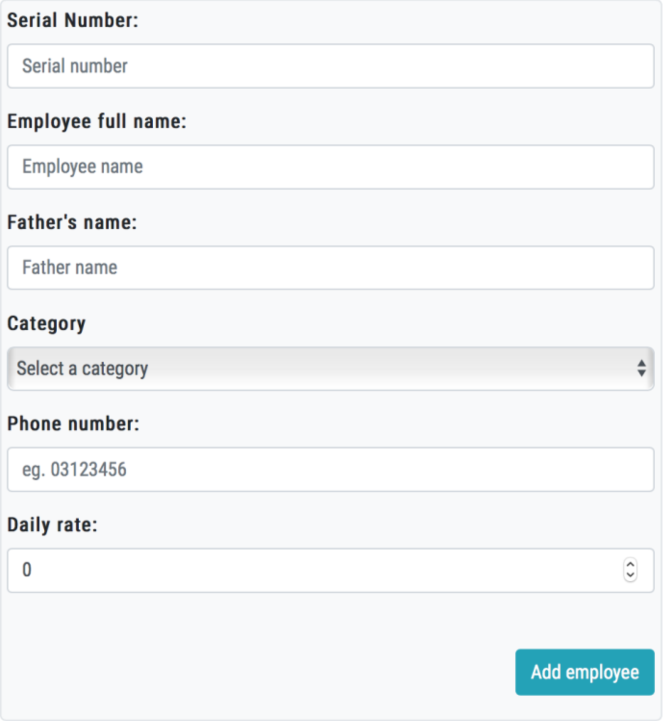
Exercise 2/homework

- Design this responsive form



Serial Number:	**Employee full name:**	**Father's name:**
Serial number	Employee name	Father name
Category	**Phone number:**	**Daily rate:**
Select a category	eg. 03123456	0

on phone
it
becomes:



Serial Number:

Serial number

Employee full name:

Employee name

Father's name:

Father name

Category

Select a category

Phone number:

eg. 03123456

Daily rate:

0