INTERNET ENGINEERING

Basir University, 2020-2021

By: Prof. Dr. Mohammad Hajarian

- Session 6

WEB DESIGN PRACTICE – CSS AND BOOTSTRAP

CASCADING STYLE SHEETS (CSS)

موسسه آموزش عالی غیر دولتی غیر انتفاعی بصیر دیگ

```
171 #content .article img.left.border {
172     padding: 0 9px 9px 0;
173     border-right: 1px dotted #999;
174     border-bottom: 1px dotted #999; }
175 #content .article blockquote {
176     margin-left: 10px;
177     padding-left: 10px;
178     border-left: 3px solid #252525; }
179 #content .article ul {
180     padding-left: 1em;
181     list-style-type: circle; }
```

TABLE OF CONTENTS

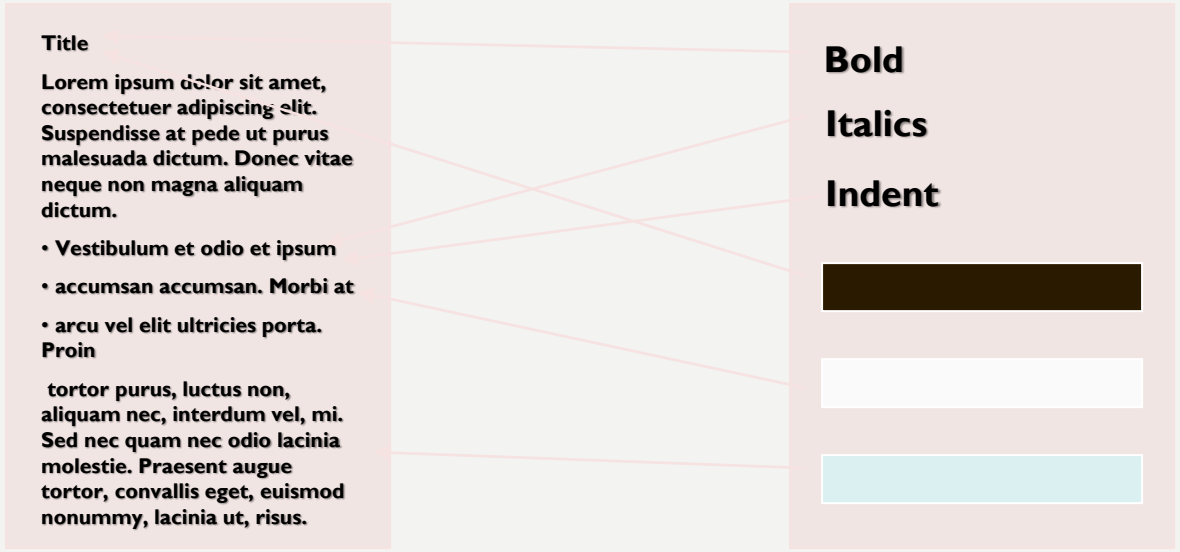
- What is CSS?
- Styling with Cascading Stylesheets (CSS)
- Selectors and style definitions
- Linking HTML and CSS
- Fonts, Backgrounds, Borders
- The Box Model
- Alignment, Z-Index, Margin, Padding
- Positioning and Floating Elements
- Visibility, Display, Overflow
- CSS Development Tools

CSS: A NEW PHILOSOPHY

- Separate content from presentation!

Content (HTML document)

Presentation (CSS Document)



THE RESULTING PAGE

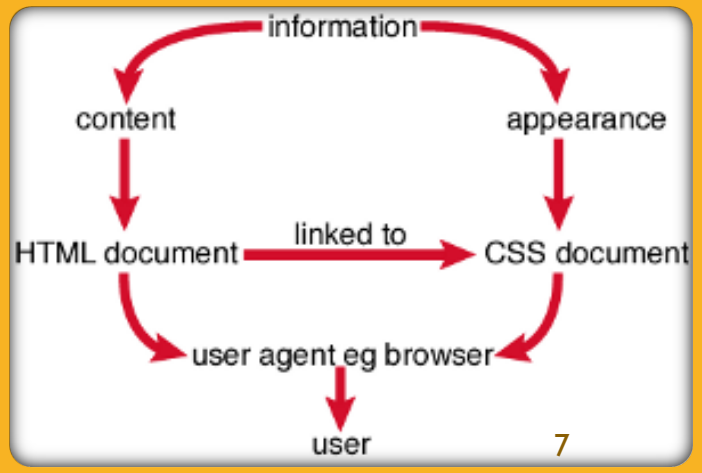
Title

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Suspendisse at pede ut purus malesuada dictum. Donec vitae neque non magna aliquam dictum.

- ***Vestibulum et odio et ipsum***
- ***accumsan accumsan. Morbi at***
- ***arcu vel elit ultricies porta. Proin***

Tortor purus, luctus non, aliquam nec, interdum vel, mi. Sed nec quam nec odio lacinia molestie. Praesent augue tortor, convallis eget, euismod nonummy, lacinia ut, risus.

CSS INTRO



Styling with Cascading Stylesheets

CSS INTRODUCTION

- Cascading Style Sheets (CSS)
  - Used to describe the presentation of documents
  - Define sizes, spacing, fonts, colors, layout, etc.
  - Improve content accessibility
  - Improve flexibility
- Designed to separate presentation from content
- Due to CSS, all HTML presentation tags and attributes are deprecated, e.g. `font`, `center`, etc.

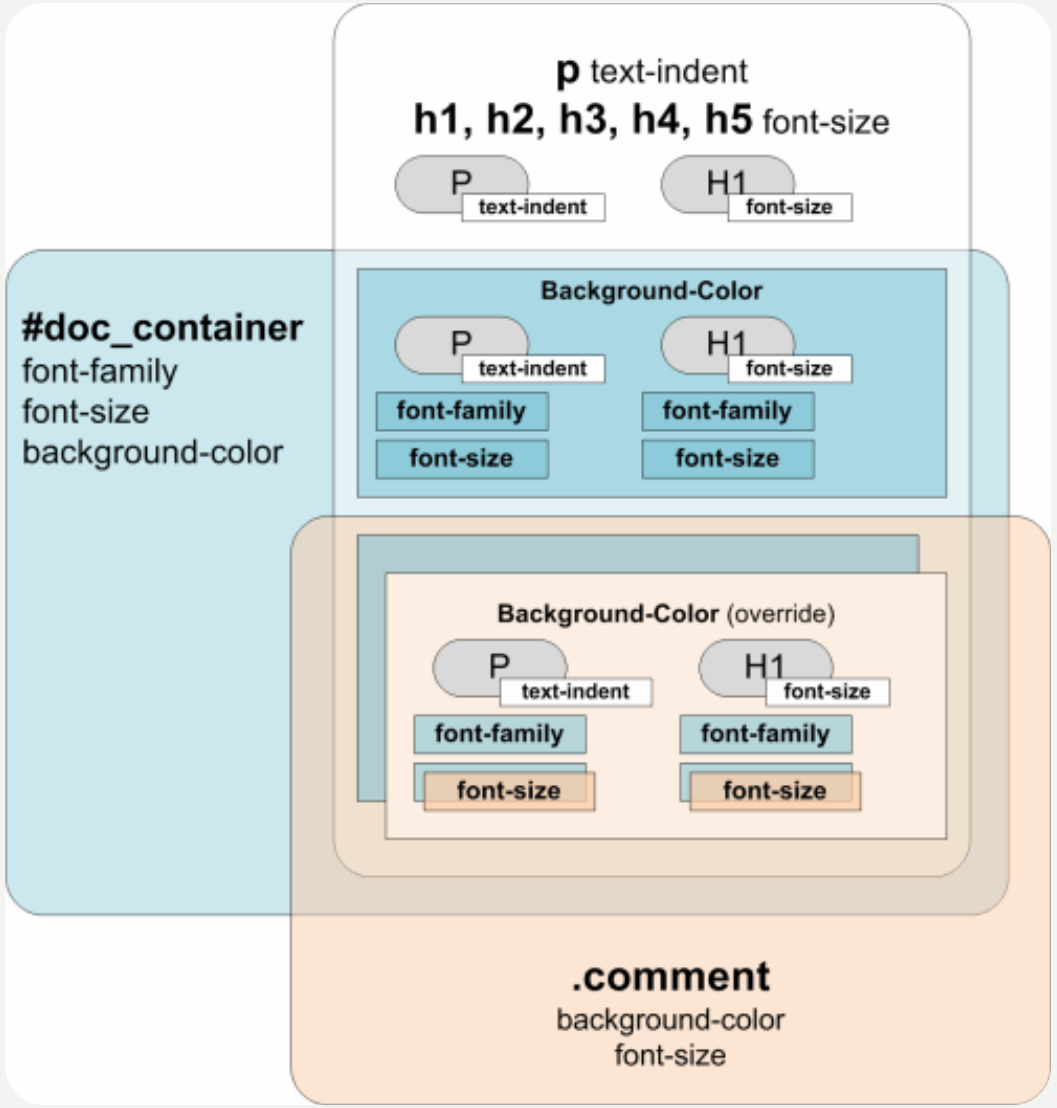
CSS INTRODUCTION (2)

- CSS can be applied to any XML document
  - Not just to HTML / XHTML
- CSS can specify different styles for different media
  - On-screen
  - In print
  - Handheld, projection, etc.
  - … even by voice or Braille-based reader

WHY "CASCADING"?

- Priority scheme determining which style rules apply to element
  - **Cascade priorities** or **specificity (weight)** are calculated and assigned to the rules
  - Child elements in the HTML DOM tree inherit styles from their parent
    - Can override them
    - Control via `!important` rule

WHY "CASCADING"? (2)

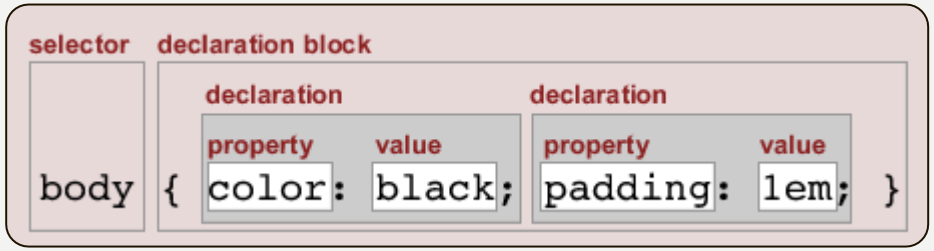


WHY "CASCADING"? (3)

- Some CSS styles are inherited and some not
  - Text-related and list-related properties are inherited - `color`, `font-size`, `font-family`, `line-height`, `text-align`, `list-style`, etc
  - Box-related and positioning styles are not inherited - `width`, `height`, `border`, `margin`, `padding`, `position`, `float`, etc
  - `<a>` elements do not inherit color and text-decoration

STYLE SHEETS SYNTAX

- Stylesheets consist of rules, selectors, declarations, properties and values



http://css.maxdesign.com.au/

- Selectors are separated by commas
- Declarations are separated by semicolons
- Properties and values are separated by colons

```
h1,h2,h3 { color: green; font-weight: bold; }
```

SELECTORS

- Selectors determine which element the rule applies to:
  - All elements of specific type (tag)
  - Those that mach a specific attribute (id, class)
  - Elements may be matched depending on how they are nested in the document tree (HTML)
- Examples:

```
.header a { color: green }
```

```
#menu>li { padding-top: 8px }
```

SELECTORS (2)

- Three primary kinds of selectors:
  - By tag (type selector):

```
h1 { font-family: verdana,sans-serif; }
```

  - By element id:

```
#element_id { color: #ff0000; }
```

  - By element class name (only for HTML):

```
.myClass {border: 1px solid red}
```

- Selectors can be combined with commas:

```
h1, .link, #top-link {font-weight: bold}
```

This will match <h1> tags, elements with class `link`, and element with id `top-link`

SELECTORS (3)

- Pseudo-classes define state
  - `:hover`, `:visited`, `:active` , `:lang`
- Pseudo-elements define element "parts" or are used to generate content
  - `:first-line` , `:before`, `:after`

```
a:hover { color: red; }
p:first-line { text-transform: uppercase; }
.title:before { content: "»"; }
.title:after { content: "«"; }
```

SELECTORS (4)

- Match relative to element placement:

```
p a {text-decoration: underline}
```

This will match all <a> tags that are inside of <p>

- * – universal selector (avoid or use with care!):

```
p * {color: black}
```

This will match all descendants of <p> element

- + selector – used to match "next sibling":

```
img + .link {float:right}
```

This will match all siblings with class name `link` that appear immediately after <img> tag

SELECTORS (5)

- > selector – matches direct child nodes:

```
p > .error {font-size: 8px}
```

This will match all elements with class `error`, direct children of `<p>` tag

- [ ] – matches tag attributes by regular expression:

```
img[alt~=logo] {border: none}
```

This will match all `<img>` tags with `alt` attribute containing the word `logo`

- `.class1.class2` (no space) - matches elements with both (all) classes applied at the same time

VALUES IN THE CSS RULES

- Colors are set in RGB format (decimal or hex):
  - Example: `#a0a6aa = rgb(160, 166, 170)`
  - Predefined color aliases exist: `black`, `blue`, etc.
- Numeric values are specified in:
  - Pixels, ems, e.g. `12px` , `1.4em`
  - Points, inches, centimeters, millimeters
    - E.g. `10pt` , `1in`, `1cm`, `1mm`
  - Percentages, e.g. `50%`
    - Percentage of what?...
  - Zero can be used with no unit: `border: 0;`

DEFAULT BROWSER STYLES

- Browsers have default CSS styles
  - Used when there is no CSS information or any other style information in the document
- Caution: default styles differ in browsers
  - E.g. margins, paddings and font sizes differ most often and usually developers reset them

```
* { margin: 0; padding: 0; }
```

```
body, h1, p, ul, li { margin: 0; padding: 0; }
```

LINKING HTML AND CSS

- HTML (content) and CSS (presentation) can be linked in three ways:
  - Inline: the CSS rules in the `style` attribute
    - No selectors are needed
  - Embedded: in the <head> in a `<style>` tag
  - External: CSS rules in separate file (best)
    - Usually a file with `.css` extension
    - Linked via `<link rel="stylesheet" href=...>` tag or `@import` directive in embedded CSS block

LINKING HTML AND CSS (2)

- Using external files is highly recommended
  - Simplifies the HTML document
  - Improves page load speed as the CSS file is cached

INLINE STYLES: EXAMPLE

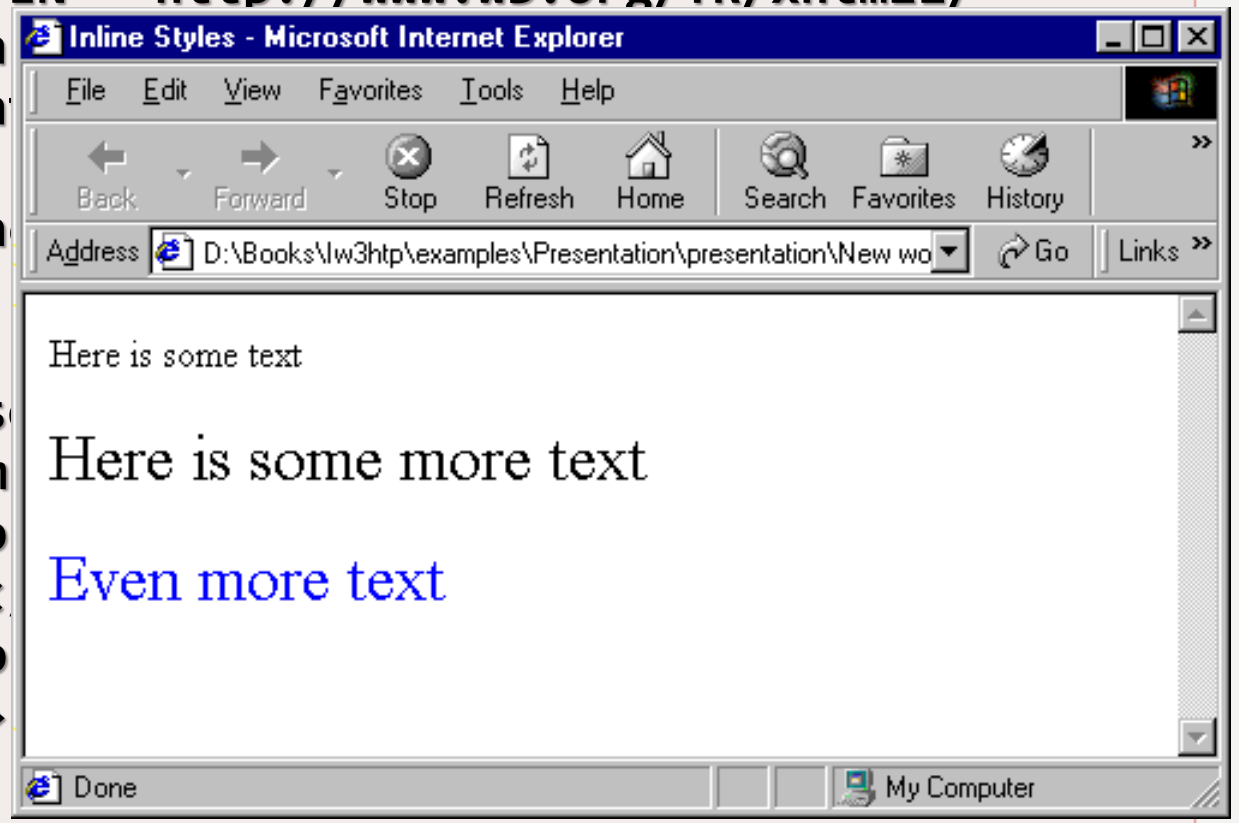
inline-styles.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/
DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Inline Styles</title>
</head>
<body>
  <p>Here is some text</p>
<!--Separate multiple styles with a semicolon-->
  <p style="font-size: 20pt">Here is some
    more text</p>
  <p style="font-size: 20pt;color:
    #0000FF" >Even more text</p>
</body>
</html>
```

INLINE STYLES: EXAMPLE

inline-styles.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/
DTD/xhtml1-tra
<html xmlns="h
<head>
  <title>Inlin
</head>
<body>
  <p>Here is s
<!--Separate m
  <p style="fo
    more text<
  <p style="fo
    #0000FF" >
</body>
</html>
```



CSS CASCADE (PRECEDENCE)

- There are browser, user and author stylesheets with "normal" and "important" declarations
  - Browser styles (least priority)
  - Normal user styles
  - Normal author styles (external, in head, inline)
  - Important author styles
  - Important user styles (max priority)

```
a { color: red !important ; }
```

http://www.slideshare.net/maxdesign/css-cascade-1658158

CSS SPECIFICITY

- CSS specificity is used to determine the precedence of CSS style declarations with the same origin. Selectors are what matters
  - Simple calculation: #id = 100, .class = 10, :pseudo = 10, [attr] = 10, tag = 1, * = 0
  - Same number of points? Order matters.

EMBEDDED STYLES

- Embedded in the HTML in the `<style>` tag:
-

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

  - The `<style>` tag is placed in the `<head>` section of the document
  - `type` attribute specifies the MIME type
    - MIME describes the format of the content
    - Other MIME types include `text/html`, `image/gif`, `text/javascript` …
    - `A media type (also known as a Multipurpose Internet Mail Extensions or MIME type) is a standard that indicates the nature and format of a document, file, or assortment of bytes. It is defined and standardized in IETF's RFC 6838.`
- Used for document-specific styles

EMBEDDED STYLES: EXAMPLE

embedded-stylesheets.html

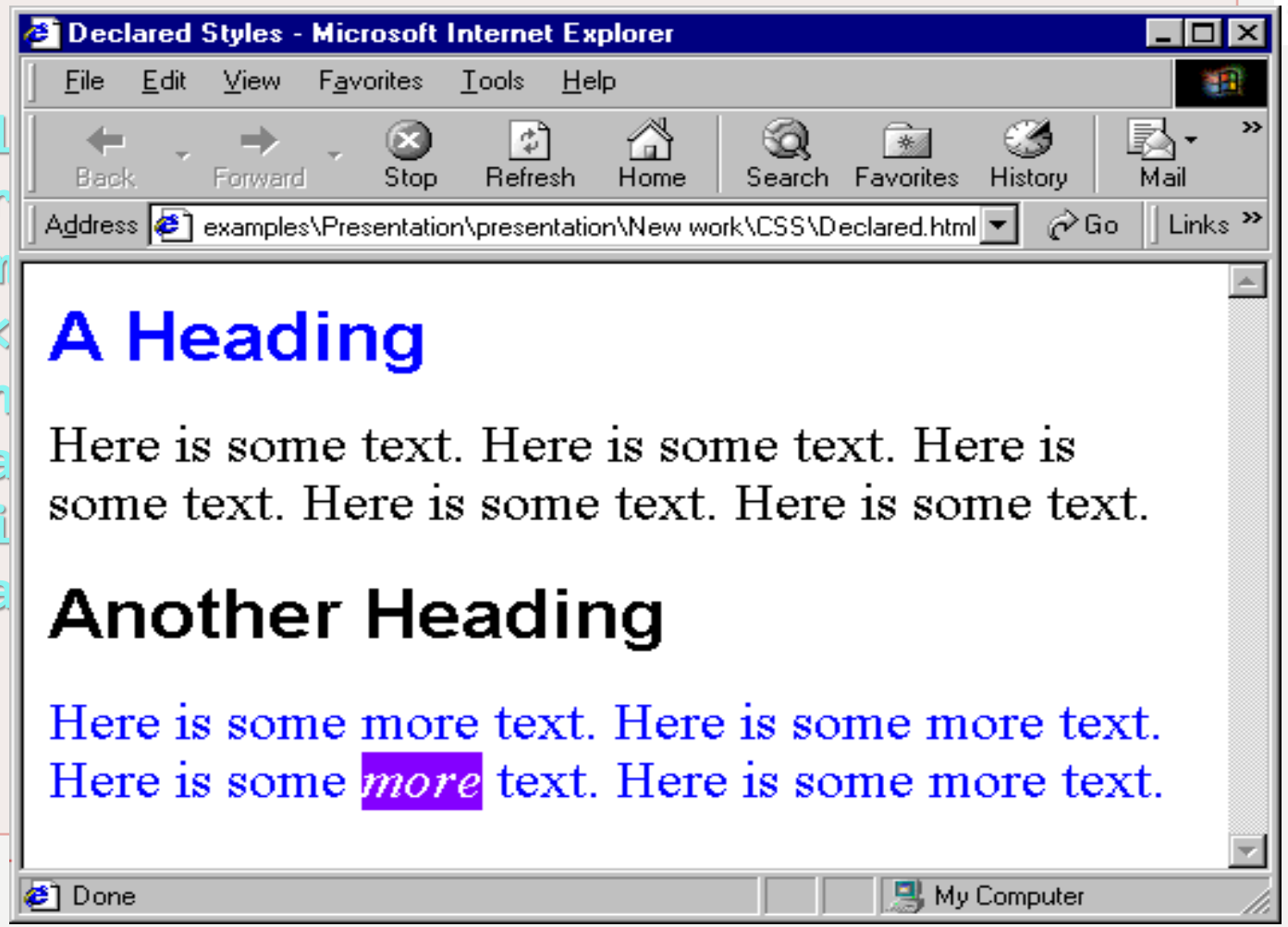
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Style Sheets</title>
  <style type="text/css">
    em {background-color:#8000FF; color:white}
    h1 {font-family:Arial, sans-serif}
    p  {font-size:18pt}
    .blue {color:blue}
  </style>
<head>
```

EMBEDDED STYLES: EXAMPLE (2)

```
…
<body>
  <h1 class="blue">A Heading</h1>
  <p>Here is some text. Here is some text. Here
  is some text. Here is some text. Here is some
  text.</p>
  <h1>Another Heading</h1>
  <p class="blue">Here is some more text.
  Here is some more text.</p>
  <p class="blue">Here is some <em>more</em>
  text. Here is some more text.</p>
</body>
</html>
```

EMBEDDED STYLES: EXAMPLE (3)

```
…
<body>
   <h1 cl
   <p>Her
   is som
   text.<
   <h1>An
   <p cla
   Here i
   <p cla
   text.
</body>
</html>
```



EXTERNAL CSS STYLES

- External linking
  - Separate pages can all use a shared style sheet
  - Only modify a single file to change the styles across your entire Web site (see http://www.csszengarden.com/)
- `link` tag (with a `rel` attribute)
  - Specifies a relationship between current document and another document

```
<link rel="stylesheet" type="text/css"
  href="styles.css">
```

  - `link` elements should be in the `<head>`

EXTERNAL CSS STYLES (2)

`@import`

- Another way to link external CSS files
- Example:

```
<style type="text/css">
  @import url("styles.css");
  /* same as */
  @import "styles.css";
</style>
```

- Ancient browsers do not recognize @import
- Use @import in an external CSS file to workaround the IE 32 CSS file limit

EXTERNAL STYLES: EXAMPLE

styles.css

```
/* CSS Document */

a    { text-decoration: none }

a:hover { text-decoration: underline;
          color: red;
          background-color: #CCFFCC }

li em   { color: red;
          font-weight: bold }

ul   { margin-left: 2cm }

ul ul   { text-decoration: underline;
          margin-left: .5cm }
```

EXTERNAL STYLES: EXAMPLE (2)

external-styles.html

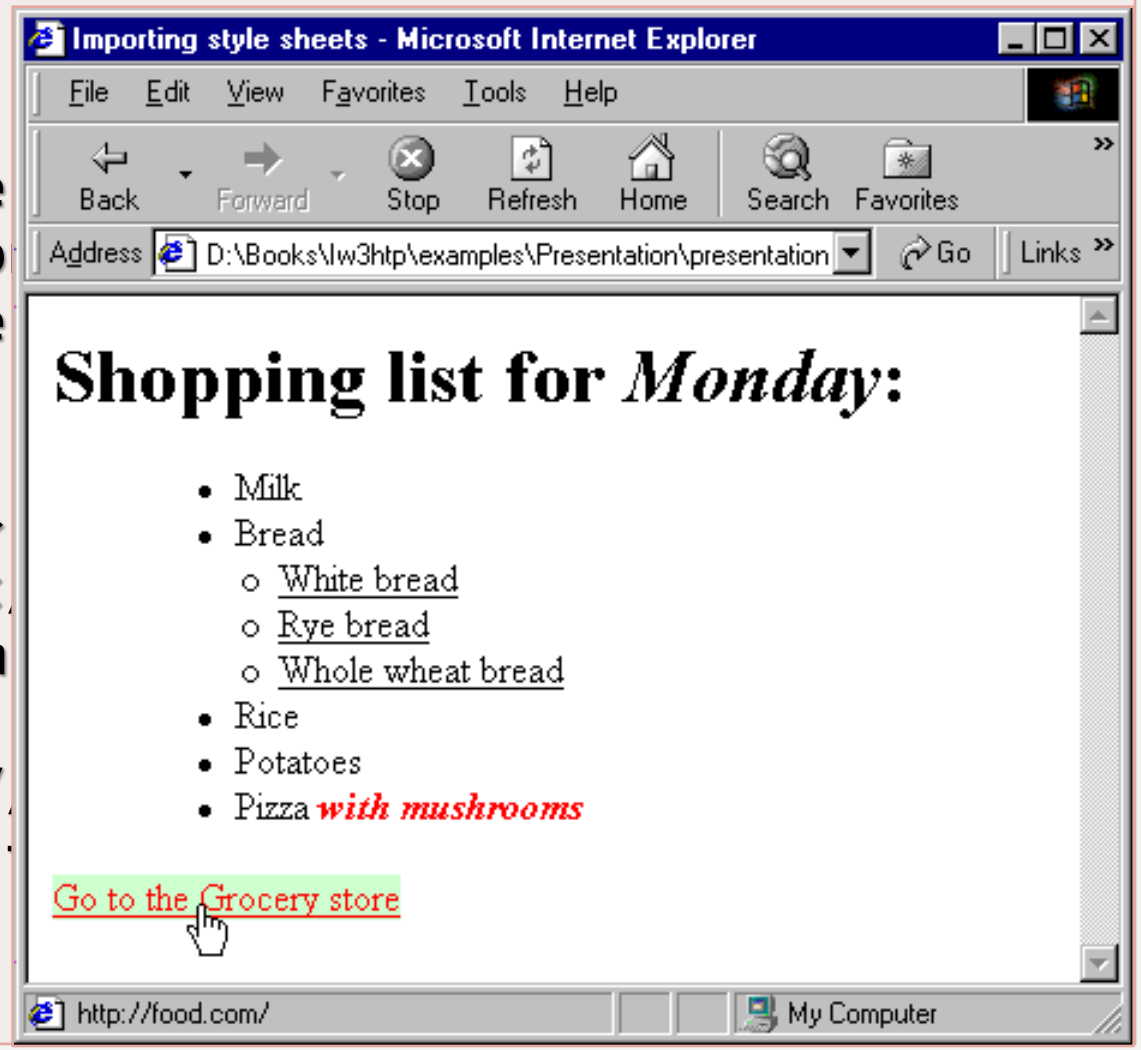
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
  Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Importing style sheets</title>
  <link type="text/css" rel="stylesheet"
    href="styles.css"  />
</head>
<body>
  <h1>Shopping list for <em>Monday</em>:</h1>
  <li>Milk</li>
  …
```

EXTERNAL STYLES: EXAMPLE (3)

```
  …
  <li>Bread
    <ul>
      <li>White bread</li>
      <li>Rye bread</li>
      <li>Whole wheat bread</li>
    </ul>
  </li>
  <li>Rice</li>
  <li>Potatoes</li>
  <li>Pizza <em>with mushrooms</em></li>
</ul>
<a href="http://food.com" title="grocery
  store">Go to the Grocery store</a>
</body>
</html>
```

EXTERNAL STYLES: EXAMPLE (4)

```
…
<li>Bread
  <ul>
    <li>White
    <li>Rye b
    <li>Whole
  </ul>
</li>
<li>Rice</li>
<li>Potatoes<
<li>Pizza <em
</ul>
<a href="http:/
  store">Go to
</body>
</html>
```



TEXT-RELATED CSS PROPERTIES

- `color` – specifies the color of the text
- `font-size` – size of font: `xx-small`, `x-small`, `small`, `medium`, `large`, `x-large`, `xx-large`, `smaller`, `larger` or numeric value
- `font-family` – comma separated font names
  - Example: `verdana`, `sans-serif`, etc.
  - The browser loads the first one that is available
  - There should always be at least one generic font
- `font-weight` can be `normal`, `bold`, `bolder`, `lighter` or a number in range [`100` … `900`]

CSS RULES FOR FONTS (2)

- `font-style` – styles the font
  - Values: `normal`, `italic`, `oblique`
- `text-decoration` – decorates the text
  - Values: none, `underline`, `line-trough`, `overline`, `blink`
- `text-align` – defines the alignment of text or other content
  - Values: `left`, `right`, `center`, `justify`

SHORTHAND FONT PROPERTY

- font
  - Shorthand rule for setting multiple font properties at the same time

```
font:italic normal bold 12px/16px verdana
```

is equal to writing this:

```
font-style: italic;
font-variant: normal;
font-weight: bold;
font-size: 12px;
line-height: 16px;
font-family: verdana;
```

BACKGROUNDS

- background-image
  - URL of image to be used as background, e.g.:

```
background-image:url("back.gif");
```

- background-color
  - Using color and image and the same time
- background-repeat
  - repeat-x, repeat-y, repeat, no-repeat
- background-attachment
  - fixed / scroll

BACKGROUNDS (2)

- `background-position`: specifies vertical and horizontal position of the background image
  - Vertical position: `top`, `center`, `bottom`
  - Horizontal position: `left`, `center`, `right`
  - Both can be specified in percentage or other numerical values
  - Examples:

```
background-position: top left;
```

```
background-position: -5px 50%;
```

BACKGROUND SHORTHAND PROPERTY

- background: shorthand rule for setting background properties at the same time:

```
background: #FFF0C0 url("back.gif") no-repeat
fixed top;
```

is equal to writing:

```
background-color: #FFF0C0;
background-image: url("back.gif");
background-repeat: no-repeat;
background-attachment: fixed;
background-position: top;
```

  - Some browsers will not apply BOTH color and image for background if using shorthand rule

BACKGROUND-IMAGE OR <IMG>?

- Background images allow you to save many image tags from the HTML
  - Leads to less code
  - More content-oriented approach
- All images that are not part of the page content (and are used only for "beautification") should be moved to the CSS

BORDERS

- `border-width`: `thin`, `medium`, `thick` or numerical value (e.g. `10px`)
- `border-color`: color alias or RGB value
- `border-style`: `none`, `hidden`, `dotted`, `dashed`, `solid`, `double`, `groove`, `ridge`, `inset`, `outset`
- Each property can be defined separately for left, top, bottom and right
  - `border-top-style`, `border-left-color`, …

BORDER SHORTHAND PROPERTY

- border: shorthand rule for setting border properties at once:

```
border: 1px solid red
```

is equal to writing:

```
border-width:1px;
border-color:red;
border-style:solid;
```

- Specify different borders for the sides via shorthand rules: border-top, border-left, border-right, border-bottom
- When to avoid border:0

WIDTH AND HEIGHT

- `width` – defines numerical value for the width of element, e.g. `200px`
- `height` – defines numerical value for the height of element, e.g. `100px`
  - By default the height of an element is defined by its content
  - Inline elements do not apply height, unless you change their `display` style.

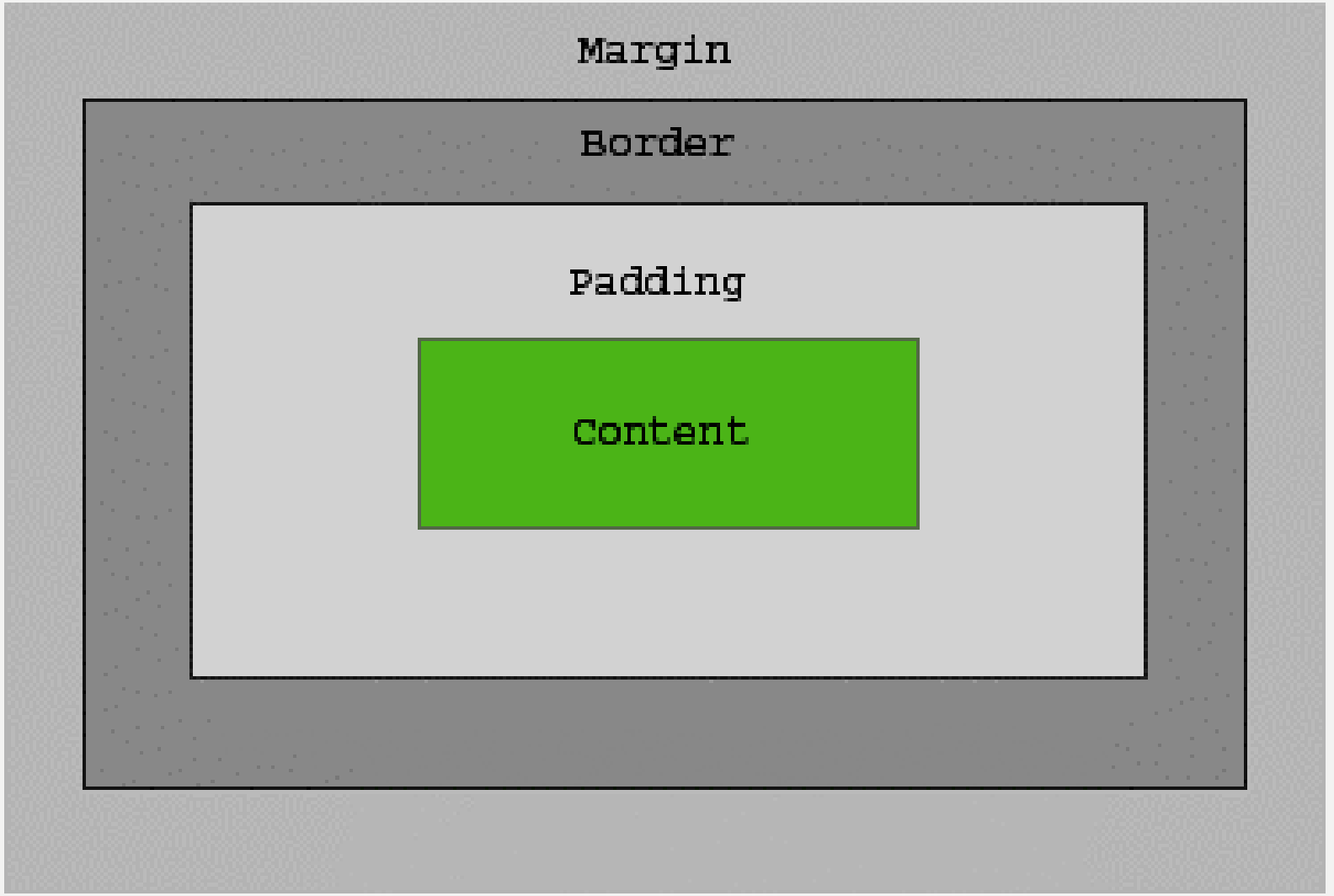
MARGIN AND PADDING

- `margin` and `padding` define the spacing around the element
  - Numerical value, e.g. `10px` or `-5px`
  - Can be defined for each of the four sides separately - `margin-top`, `padding-left`, …
  - `margin` is the spacing outside of the border
  - `padding` is the spacing between the border and the content
  - What are collapsing margins?

MARGIN AND PADDING: SHORT RULES

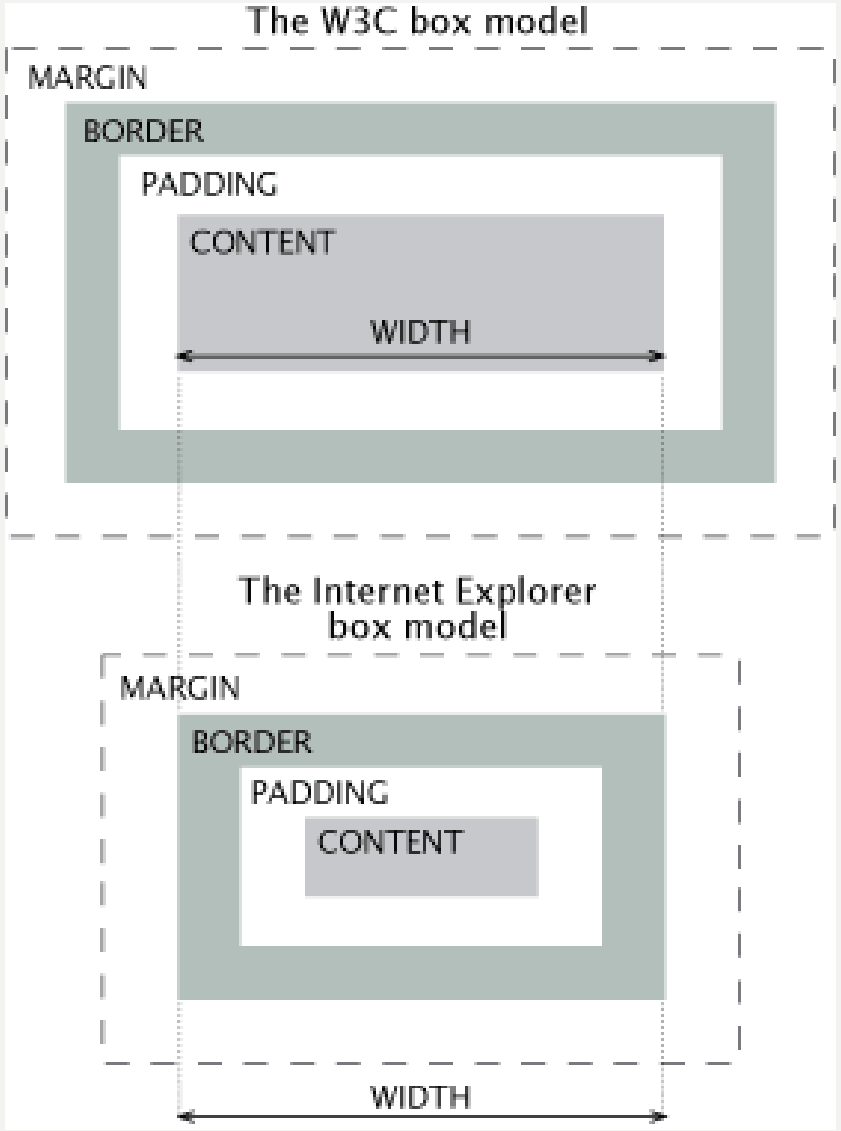
- `margin: 5px;`
  - Sets all four sides to have margin of 5 px;
- `margin: 10px 20px;`
  - top and bottom to `10px`, left and right to `20px`;
- `margin: 5px 3px 8px;`
  - top 5px, left/right 3px, bottom 8px
- `margin: 1px 3px 5px 7px;`
  - top, right, bottom, left (clockwise from top)
- Same for `padding`

THE BOX MODEL



IE QUIRKS MODE

- When using quirks mode (pages with no DOCTYPE or with a HTML 4 Transitional DOCTYPE), Internet Explorer violates the box model standard



POSITIONING

- `position`: defines the positioning of the element in the page content flow
- The value is one of:
  - `static` (default)
  - `relative` – relative position according to where the element would appear with static position
  - `absolute` – position according to the innermost positioned parent element
  - `fixed` – same as absolute, but ignores page scrolling

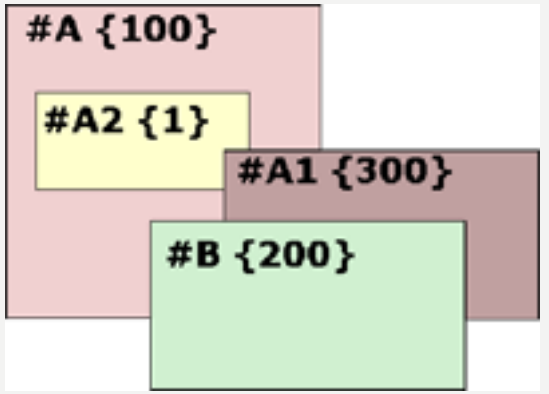
POSITIONING (2)

- Margin VS relative positioning
- Fixed and absolutely positioned elements do not influence the page normal flow and usually stay on top of other elements
  - Their position and size is ignored when calculating the size of parent element or position of surrounding elements
  - Overlaid according to their z-index
  - Inline fixed or absolutely positioned elements can apply height like block-level elements

POSITIONING (3)

- `top`, `left`, `bottom`, `right`: specifies offset of absolute/fixed/relative positioned element as numerical values
- `z-index` : specifies the stack level of positioned elements
  - Understanding stacking context

Each positioned element creates a stacking context.
Elements in different stacking contexts are overlapped according to the stacking order of their containers. For example, there is no way for #A1 and #A2 (children of #A) to be placed over #B without increasing the z-index of #A.



INLINE ELEMENT POSITIONING

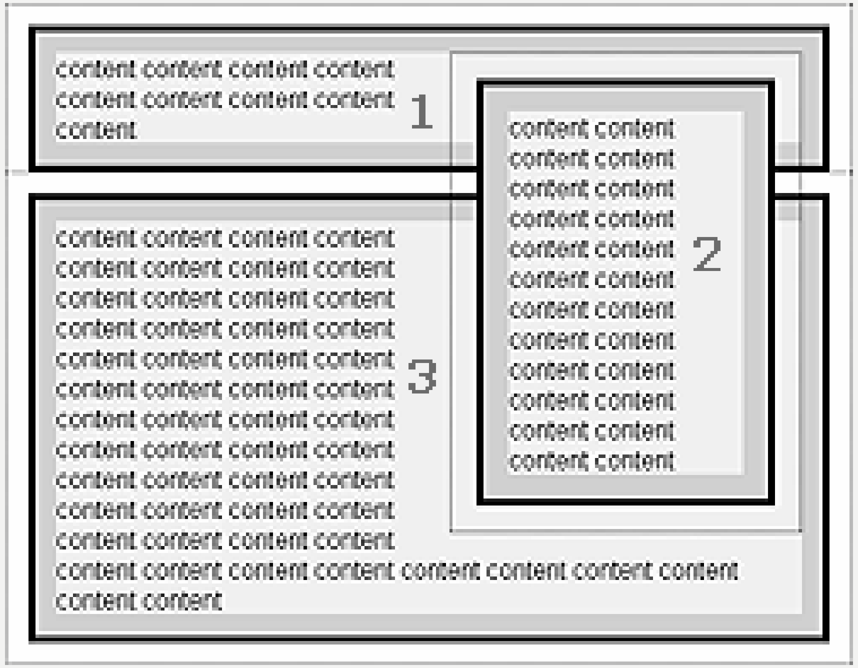
- `vertical-align`: sets the vertical-alignment of an inline element, according to the line height
  - Values: `baseline`, `sub`, `super`, `top`, `text-top`, `middle`, `bottom`, `text-bottom` or numeric
- Also used for content of table cells (which apply `middle` alignment by default)

FLOAT

- `float`: the element "floats" to one side
  - `left`: places the element on the left and following content on the right
  - `right`: places the element on the right and following content on the left
  - floated elements should come before the content that will wrap around them in the code
  - margins of floated elements do not collapse
  - floated inline elements can apply height

FLOAT (2)

- How floated elements are positioned



CLEAR

- `clear`
  - Sets the sides of the element where other floating elements are NOT allowed
  - Used to "drop" elements below floated ones or expand a container, which contains only floated children
  - Possible values: `left`, `right`, `both`
- Clearing floats
  - additional element (`<div>`) with a clear style

CLEAR (2)

- Clearing floats (continued)
  - `:after { content: ""; display: block; clear: both; height: 0; }`
  - Triggering hasLayout in IE expands a container of floated elements
    - display: inline-block;
    - zoom: 1;

OPACITY

- `opacity`: specifies the opacity of the element
  - Floating point number from 0 to 1
  - For old Mozilla browsers use `-moz-opacity`
  - For IE use `filter:alpha(opacity=value)` where value is from 0 to 100; also, "binary and script behaviors" must be enabled and `hasLayout` must be triggered, e.g. with `zoom:1`

VISIBILITY

- `visibility`
  - Determines whether the element is visible
  - `hidden`: element is not rendered, but still occupies place on the page (similar to `opacity:0`)
  - `visible`: element is rendered normally

DISPLAY

- `display`: controls the display of the element and the way it is rendered and if breaks should be placed before and after the element
  - `inline`: no breaks are placed before and after (`<span>` is an inline element)
  - `block`: breaks are placed before AND after the element (`<div>` is a block element)

DISPLAY (2)

- `display`: controls the display of the element and the way it is rendered and if breaks should be placed before and after the element
  - none: element is hidden and its dimensions are not used to calculate the surrounding elements rendering (differs from `visibility: hidden`!)
  - There are some more possible values, but not all browsers support them
    - Specific displays like `table-cell` and `table-row`

OVERFLOW

- `overflow`: defines the behavior of element when content needs more space than you have specified by the size properties or for other reasons. Values:
  - `visible` (default) – content spills out of the element
  - `auto` - show scrollbars if needed
  - `scroll` – always show scrollbars
  - `hidden` – any content that cannot fit is clipped

OTHER CSS PROPERTIES

- `cursor`: specifies the look of the mouse cursor when placed over the element
  - Values: `crosshair`, `help`, `pointer`, `progress`, `move`, `hair`, `col-resize`, `row-resize`, `text`, `wait`, `copy`, `drop`, and others
- `white-space` – controls the line breaking of text. Value is one of:
  - `nowrap` – keeps the text on one line
  - `normal` (default) – browser decides whether to brake the lines if needed

BENEFITS OF USING CSS

- More powerful formatting than using presentation tags
- Your pages load faster, because browsers cache the `.css` files
- Increased accessibility, because rules can be defined according given media
- Pages are easier to maintain and update

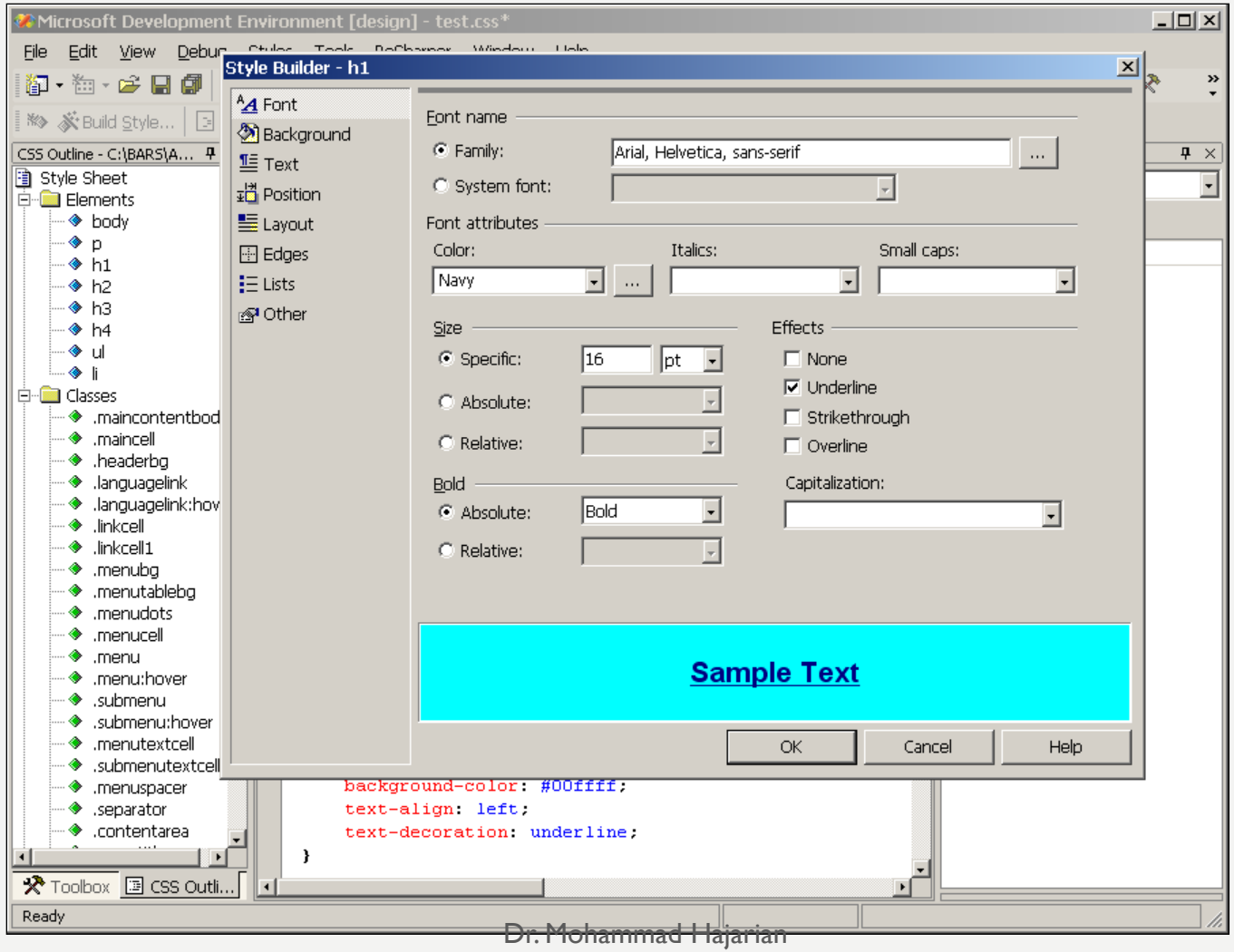
MAINTENANCE EXAMPLE

موسسه آموزش عالی غیر دولتی غیر انتفاعی بصیر نیک

CSS file

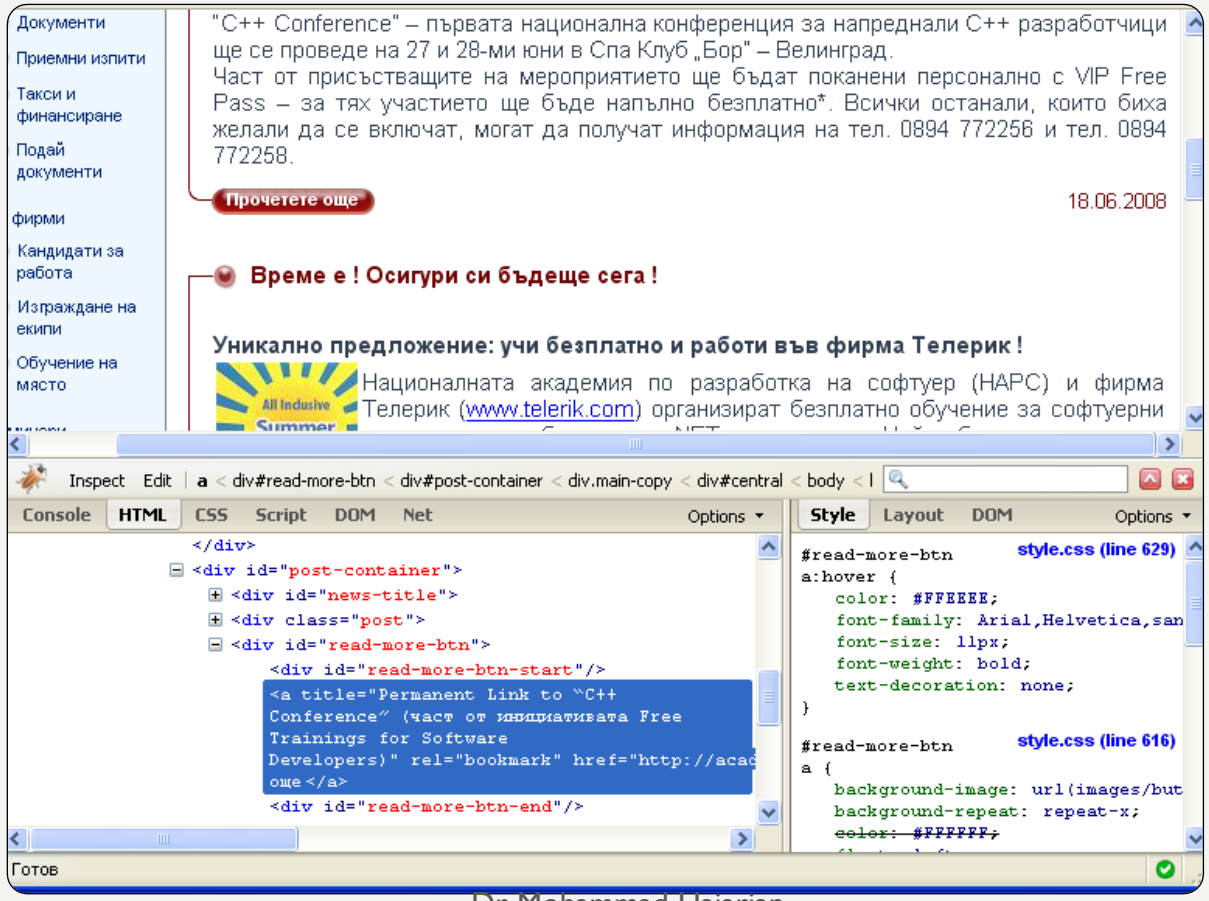
CSS DEVELOPMENT TOOLS

- Visual Studio – CSS Editor



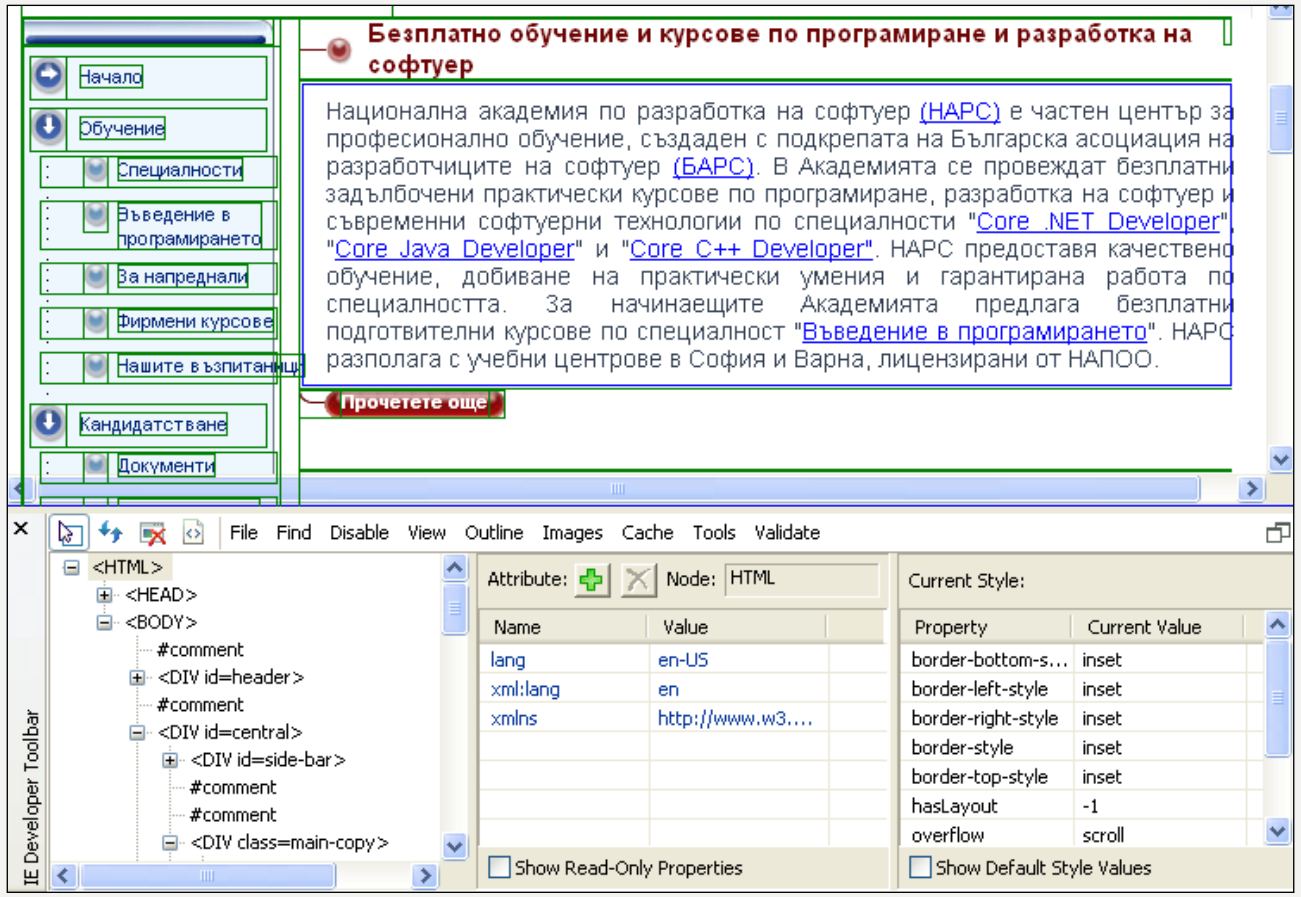
CSS DEVELOPMENT TOOLS (3)

- Firebug – add-on to Firefox used to examine and adjust CSS and HTML



CSS DEVELOPMENT TOOLS (4)

- IE Developer Toolbar – add-on to IE used to examine CSS and HTML (press [F12])



Bootstrap

What is Bootstrap?

- Bootstrap is a free front-end framework (HTML and CSS) for faster and easier web development

- Bootstrap is famous for being developed with components that have the ability to follow the property of responsive designs
  - Responsive Design is about using CSS and HTML to resize, hide, shrink, enlarge, or move the content to make it look good on any screen
  - Responsive Design allow your page works for computer, tablets and mobile phones.

What is Bootstrap?

- In other words, bootstrap is a collection of CSS classes and JavaScript functions the you get ready to use and will not worry about write code like this:

```
.glyphicon {
    position: relative;
    top: 1px;
    display: inline-block;
    font-family: 'Glyphicons Halflings';
    font-style: normal;
    font-weight: 400;
    line-height: 1;
    -webkit-font-smoothing: antialiased;
    -moz-osx-font-smoothing: grayscale
}
```

```
(jQuery), +function(a) {
  "use strict";
  function b(b) {
      return this.each(function() {
          var c = a(this),
              e = c.data("bs.alert");
          e || c.data("bs.alert", e = new d(this)), "string" == typeof b && e[b].call(c)
      })
  }
  var c = '[data-dismiss="alert"]',
      d = function(b) {
          a(b).on("click", c, this.close)
      };
```

First Bootstrap Page

- Include the **HTML5 doctype** at the beginning of the page, along with the **lang** attribute and the correct **character set**

```
<!DOCTYPE html>
<html lang="pt">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
  </head>
</html>
```

First Bootstrap Page

- Meta **viewport tag** ensure proper rendering and touch zooming
- Width of the page to follow the screen-width of the device and initial zoom level to default zoom level of device
- User will have a better experience, thus will not need zoom the page to view page information

First Bootstrap Page

- You will need to include three files:
  - **bootstrap.min.css**
  - **jquery.min.js**
  - **bootstrap.min.js**

- You must download it and include in you page
- For this example, we include from the Internet

```
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
```

First Bootstrap Page

- Bootstrap has a concept of containing element to wrap site contents

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

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

Bootstrap Grids

- Bootstrap's grid system allows up to 12 columns across the page
- You can divide the container in rows and each row in columns with space multiple of the 12

Bootstrap Grids

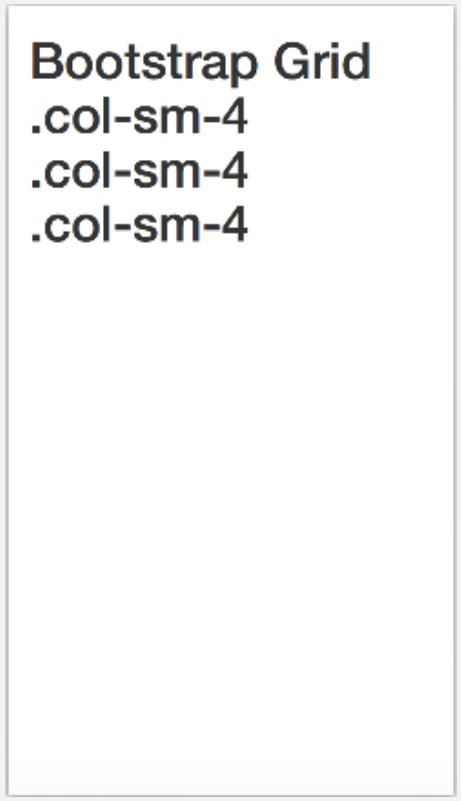
- In the bellow example, we divide the space inside the container in 1 row and this row in 3 columns with the same side. 4+4+4 = 12
- You can use any combination that the sum be equal to 12.

```
<div class="container">
  <h1>Bootstrap Grid</h1>
  <div class="row">
      <div class="col-sm-4">.col-sm-4</div>
      <div class="col-sm-4">.col-sm-4</div>
      <div class="col-sm-4">.col-sm-4</div>
    </div>
</div>
```

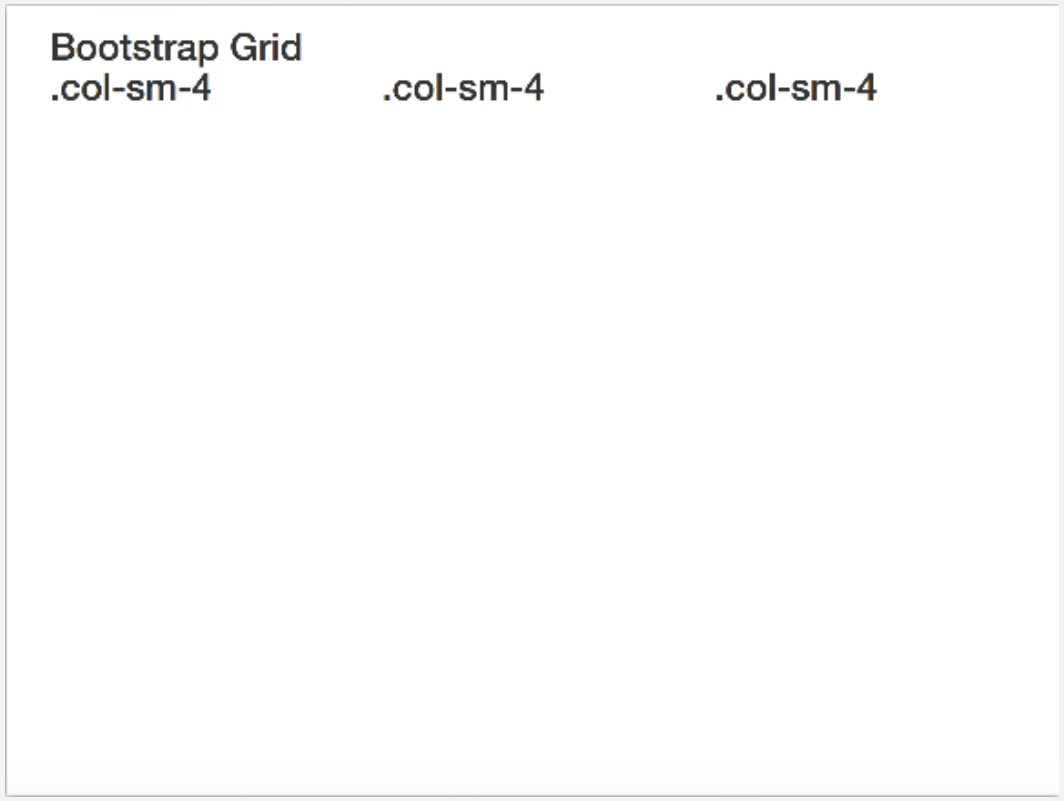
Bootstrap Grids

- Responsive Bootstrap's grid system with 3 columns

small screens



big screens



Bootstrap Table

- 4 main classes:
  - .table
  - .table-striped
  - .table-bordered
  - .table-hover
- 5 contextual classes:
  - .active
  - .success
  - .info
  - .warning
  - .danger

Bootstrap Tables

```
<div class="container">
  <h1>Bootstrap Table</h1>
  <table class="table table-striped table-bordered table-hover">
    <thead>
      <tr>
        <th>Firstname</th>
        <th>Lastname</th>
        <th>Email</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>John</td>
        <td>Doe</td>
        <td>john@example.com</td>
      </tr>
      <tr>
        <td>Joseph</td>
        <td>Saints</td>
        <td>joseph@example.com</td>
      </tr>
      <tr class="warning">
        <td>Mary Help</td>
        <td>Saints</td>
        <td>mary@example.com</td>
      </tr>
    </tbody>
  </table>
</div>
```

Bootstrap Tables

Bootstrap Table

Firstname	Lastname	Email
John	Doe	john@example.com
Joseph	Saints	joseph@example.com
Mary Help	Saints	mary@example.com

Bootstrap Images

- 3 main classes:
  - . *img-rounded*
  - . img-circle
  - . *img-thumbnail*

```
<div class="row">
  <div class="col-md-4">
      <img class="img-rounded" src="garden-house-hotel-11.jpg" alt="Garden" style="width:100%">
  </div>
  <div class="col-md-4">
      <img class="img-circle" src="XMAS-HOUSE-5-256x256.jpg" alt="Snow" style="width:100%">
  </div>
  <div class="col-md-4">
      <img class="img-thumbnail" src="Accommodation_Image.jpg" alt="Beach" style="width:100%">
  </div>
</div>
```

Bootstrap Images

Bootstrap Alerts

- Bootstrap provides an easy way to create predefined alert messages
- Alerts are created with the .alert class, followed by one of the four contextual classes
- .alert-success
- .alert-info
- .alert-warning
- .alert-danger

Bootstrap Alerts

```
<div class="container">
  <h1>Bootstrap Alerts</h1>

  <div class="alert alert-success">
    <strong>Success!</strong> Indicates a successful or positive action.
  </div>

  <div class="alert alert-info">
    <strong>Info!</strong> Indicates a neutral informative change or action.
  </div>

  <div class="alert alert-warning">
    <strong>Warning!</strong> Indicates a warning that might need attention.
  </div>

  <div class="alert alert-danger">
    <strong>Danger!</strong> Indicates a dangerous or potentially negative action.
  </div>
</div>
```

Bootstrap Alerts

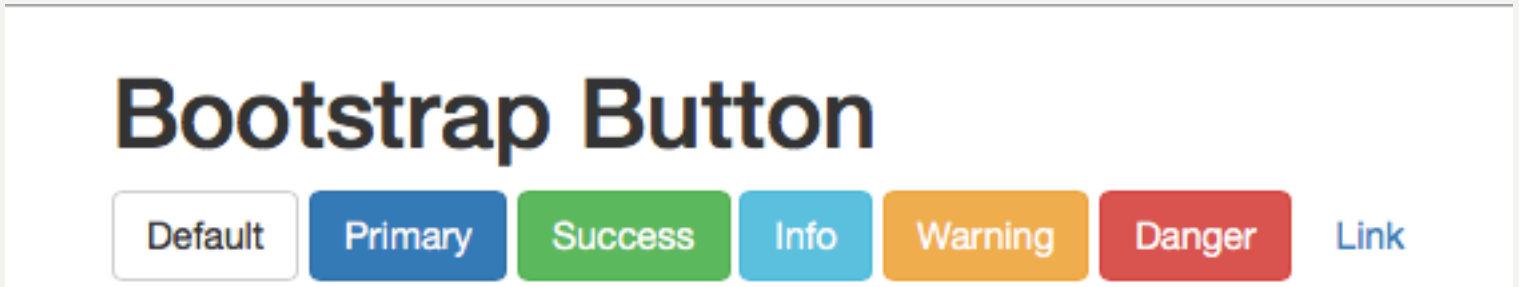
Bootstrap Buttons

- Bootstrap provides seven styles of buttons:
- To achieve the button styles above, Bootstrap has the following contextual classes:

  - .btn-default
  - .btn-primary
  - .btn-success
  - .btn-info
  - .btn-warning
  - .btn-danger
  - .btn-link

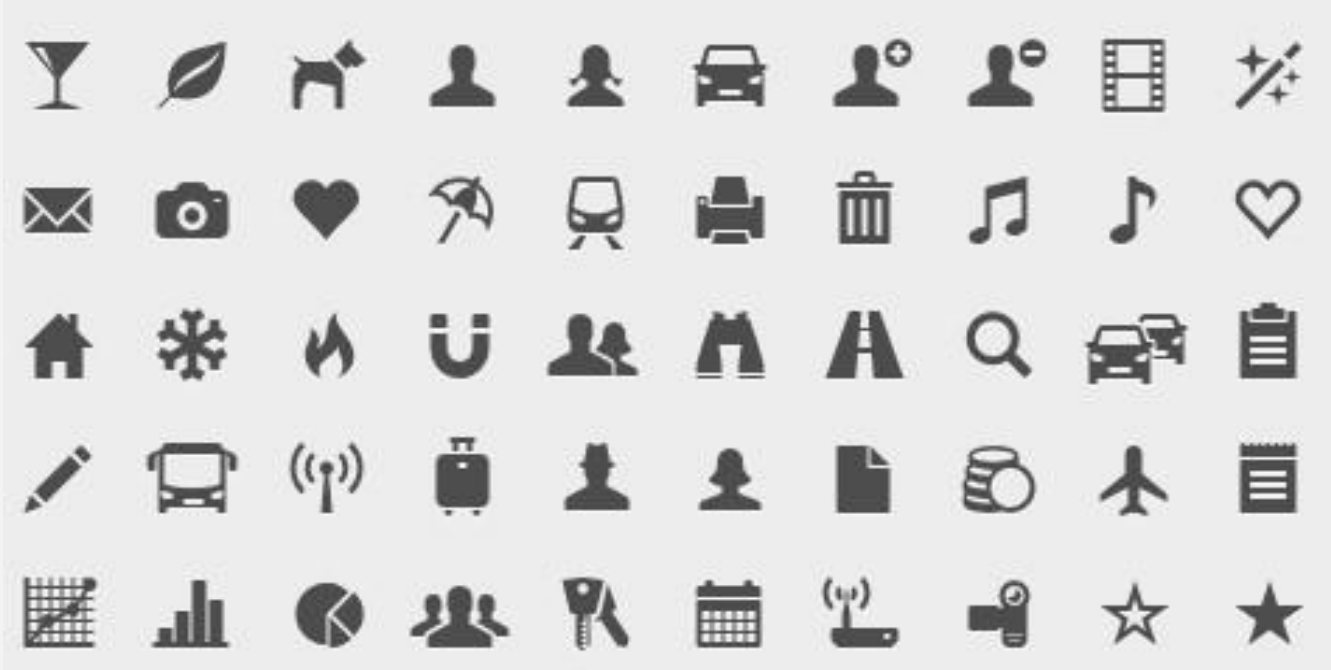
Bootstrap Buttons

```
<div class="container">
  <h1>Bootstrap Button</h1>

    <button type="button" class="btn btn-default">Default</button>
    <button type="button" class="btn btn-primary">Primary</button>
    <button type="button" class="btn btn-success">Success</button>
    <button type="button" class="btn btn-info">Info</button>
    <button type="button" class="btn btn-warning">Warning</button>
    <button type="button" class="btn btn-danger">Danger</button>
    <button type="button" class="btn btn-link">Link</button>

</div>
```



Bootstrap Glyphicons

- Bootstrap provides 260 glyphicons from the Glyphicons Halflings set
- ([http://getbootstrap.com/components/)](http://getbootstrap.com/components/)



Bootstrap Glyphicons

```
<div class="container">
  <h1>Bootstrap glyphicon</h1>

    <p> Simple Span:
        <span class="glyphicon glyphicon-check"></span>
    </p>

    <p>Default button with icon:
        <button type="button" class="btn btn-default"> Play
            <span class="glyphicon glyphicon-play"></span>
        </button>
    </p>

    <p>Search icon on a styled button:
        <button type="button" class="btn btn-info">
            <span class="glyphicon glyphicon-search"></span> Search
        </button>
    </p>

    <p>Link with icon:
        <a href="#" class="btn btn-success btn-lg">
            <span class="glyphicon glyphicon-print"></span> Print
        </a>
    </p>

</div>
```

Bootstrap Glyphicons

Bootstrap Labels

- Labels are used to provide information about something
- Bootstrap create labels with colorful backgrounds to highlight the text inside de label
- Use the .label class, followed by one of the six contextual classes .label-default, .label-primary, .label-success, .label-info, .label-warning or .label-danger

Bootstrap Labels

```
<div class="container">
    <h1>Bootstrap Label</h1>

    <span class="label label-default">Default Label</span>
    <span class="label label-primary">Primary Label</span>
    <span class="label label-success">Success Label</span>
    <span class="label label-info">Info Label</span>
    <span class="label label-warning">Warning Label</span>
    <span class="label label-danger">Danger Label</span>
```

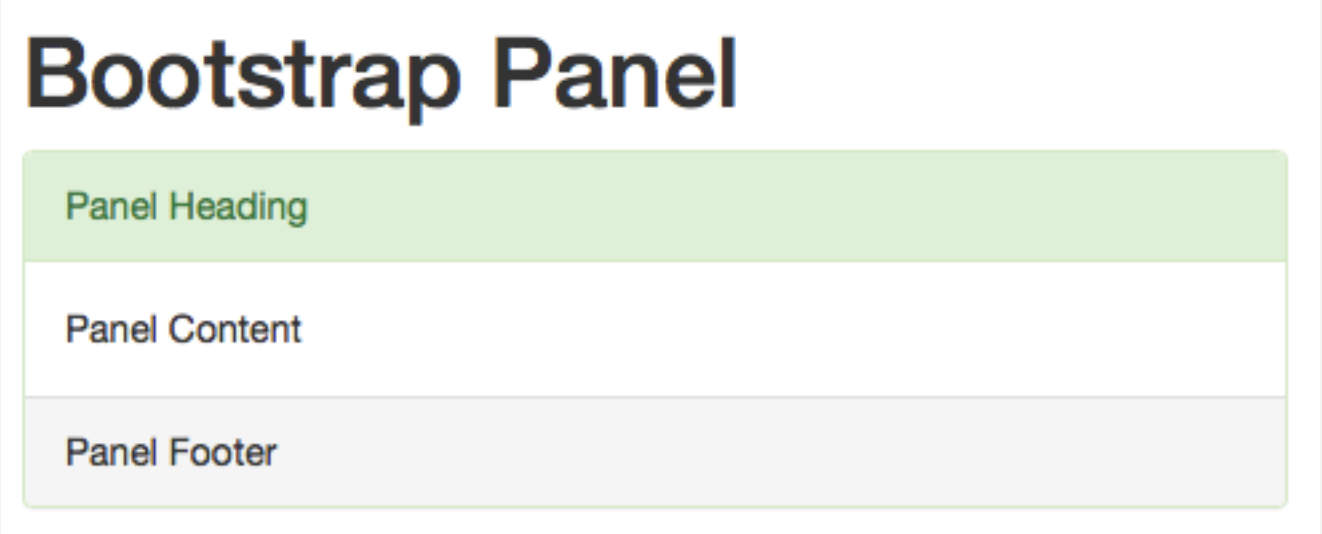
Bootstrap Panels

- A panel in bootstrap is a bordered box with some padding around its content that can be use to highlight or separated some information

- Like other bootstrap elements panel has contextual classes also (.panel-default, .panel-primary, .panel-success, .panel-info, .panel-warning, or .panel-danger)

Bootstrap Panels

```
<div class="container">
    <h1>Bootstrap Panel</h1>

    <div class="panel panel-success">
        <div class="panel-heading">Panel Heading</div>
        <div class="panel-body">Panel Content</div>
        <div class="panel-footer">Panel Footer</div>
    </div>
```



Bootstrap Themes and Templates

- You can find several templates e themes free to download
  - A **theme** consists of customized CSS
  - A **template** consists of one or more predesigned HTML pages, which often make use of a theme
- Both are collection of bootstrap elements (grids, buttons, panels), put together for someone, until have a fully functional website or web application.
- You can find some free templates here:

  https://startbootstrap.com/template-categories/all/

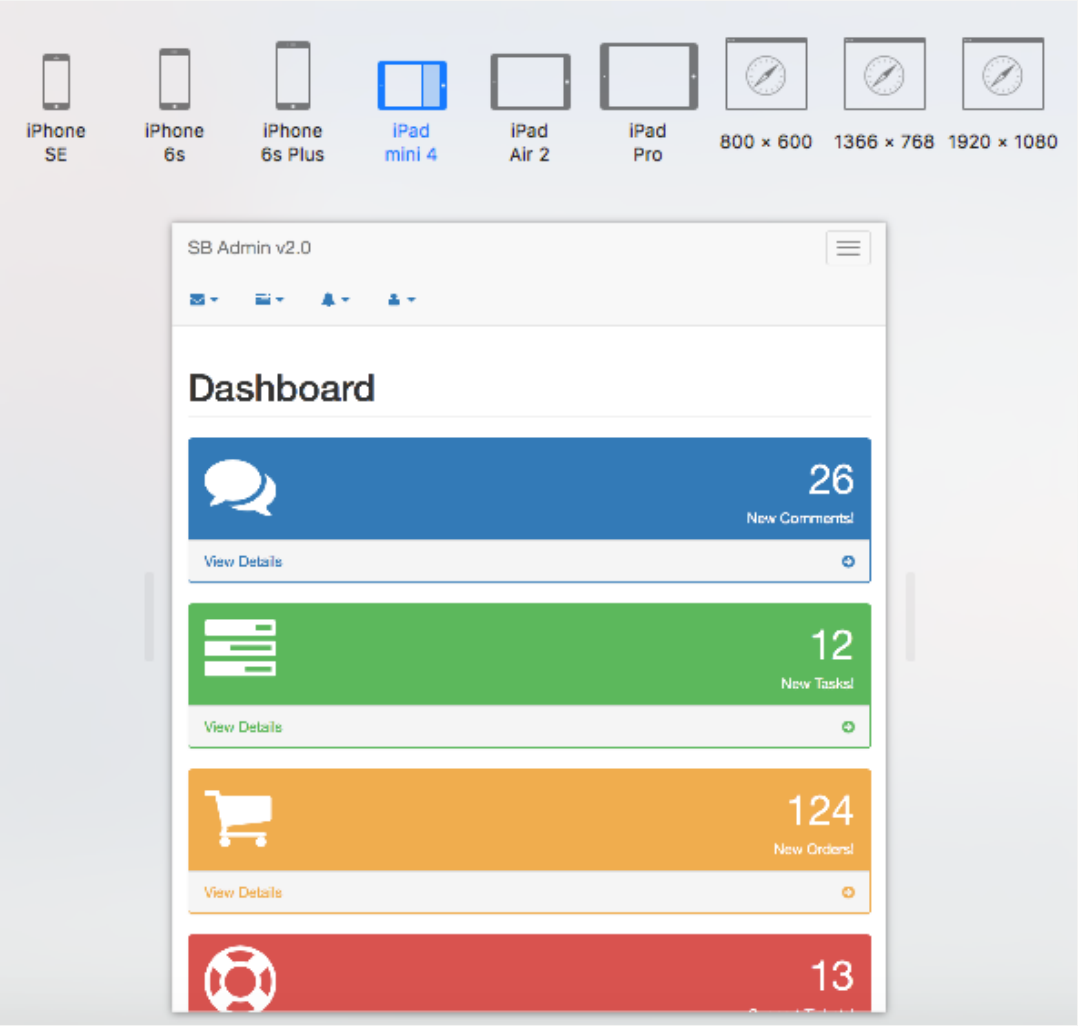
Bootstrap Themes and Templates

- For web application **SB Admin 2** template is appropriate. It has forms, tables, charts and other useful components

Bootstrap Themes and Templates

- Download **SB Admin 2** template for your computer and open the /pages/index.html in your web browser
- Start to customize it, instead write you application design from scratch

Bootstrap Themes and Templates



موسسه آموزش عالی غیر دولتی غیر انتفاعی بصیر بیک

- **References**
  - w3schools. Bootstrap 3 Tutorial. https://www.w3schools.com/bootstrap/

- **Source Code**
  - https://github.com/jadsonjs/bootstrap

PROJECT 1 (2 POINTS)

- یک سوپرمارکت درنظر دارد محصولات خود را به صورت اینترنتی برای فروش قراردهد. این سوپر مارکت محصولات خود را به چهار دسته عمده تقسیم میکند که شامل آرایشی و بهداشتی، خواربار، میوه و نوشیدنی ها میباشد. در هریک از این بخش ها کاربران میتوانند محصولات موجود را مشاهده نمایند و برای اطلاعات بیشتر در مورد قیمت و توضیحات محصول، وارد صفحه جزئیات محصول شوند. کاربران در صفحه جزئیات محصول میتوانند سفارش خود را ثبت نمایند. این سوپرمارکت در نظر دارد ثبت سفارش کاربران با استفاده از شماره تماس آنها صورت گیرد. همچنین هنگام ثبت سفارش آدرس منزل کاربر برای تحویل محصول دریافت میشود . پس از ثبت سفارش، مدیر فروشگاه قادر خواهد بود در صفحه مخصوص خود سفارش های ارسال شده را به همراه آدرس و شماره تماس کاربران مشاهده نماید

وظایف:

- با رعایت اصول طرحی و قابلیت استفاده، طراحی low-fidelity صفحات سایت این فروشگاه شامل wireframe و storyboard را به صورت دستی یا با استفاده از نرم افزار بالزامیک انجام دهید.
- طراحی high-fidelity صفحات سایت این فروشگاه را بر اساس طراحی که در مرحله قبل انجام دادید با استفاده از HTML, CSS و Bootstrap پیاده سازی نمایید.

INSTRUCTIONS

پس از اتمام پروژه فایلهای خود را به آدرس MAINSYSTEM@GMAIL.COM ایمیل نمایید.

- مهلت ارسال تا یک روز قبل از جلسه نهم میباشد.
- ارسال با تاخیر تنها یک هفته دیر تر و با کسر ۵۰٪ نمره امکان پذیر است.
- از قراردادن مسائلی که تدریس نشده در پاسخ ها خود داری کنید (نمره منفی خواهد داشت).

Q/A

- End of Session 6

موسسه آموزش عالی غیر دولتی غیر انتفاعی بصیر بیک

THANK YOU!