

CST463 - WEB PROGRAMMING



Module 2

- ▶ (CSS, JavaScript) Introduction to Stylesheets : Introduction to CSS-Basic syntax and structure-Inline Styles, Embedded Style Sheets, Conflict Resolution, Linking External Style Sheets-Exploring CSS Selectors-Properties, values, Positioning Elements: Absolute Positioning, Relative Positioning - Backgrounds-List Styles-Element Dimensions- Table Layouts-Box Model and Text Flow-div and span -Basics of Responsive CSS, Media port & Media Queries.
- ▶ Introduction to JavaScript : Introduction to Scripting- Programming fundamentals of JavaScript -Obtaining User Input with prompt Dialogs-Arithmetic-Decision Making -Control Statements - Functions -Arrays - Objects -Document Object Model (DOM) -Form processing

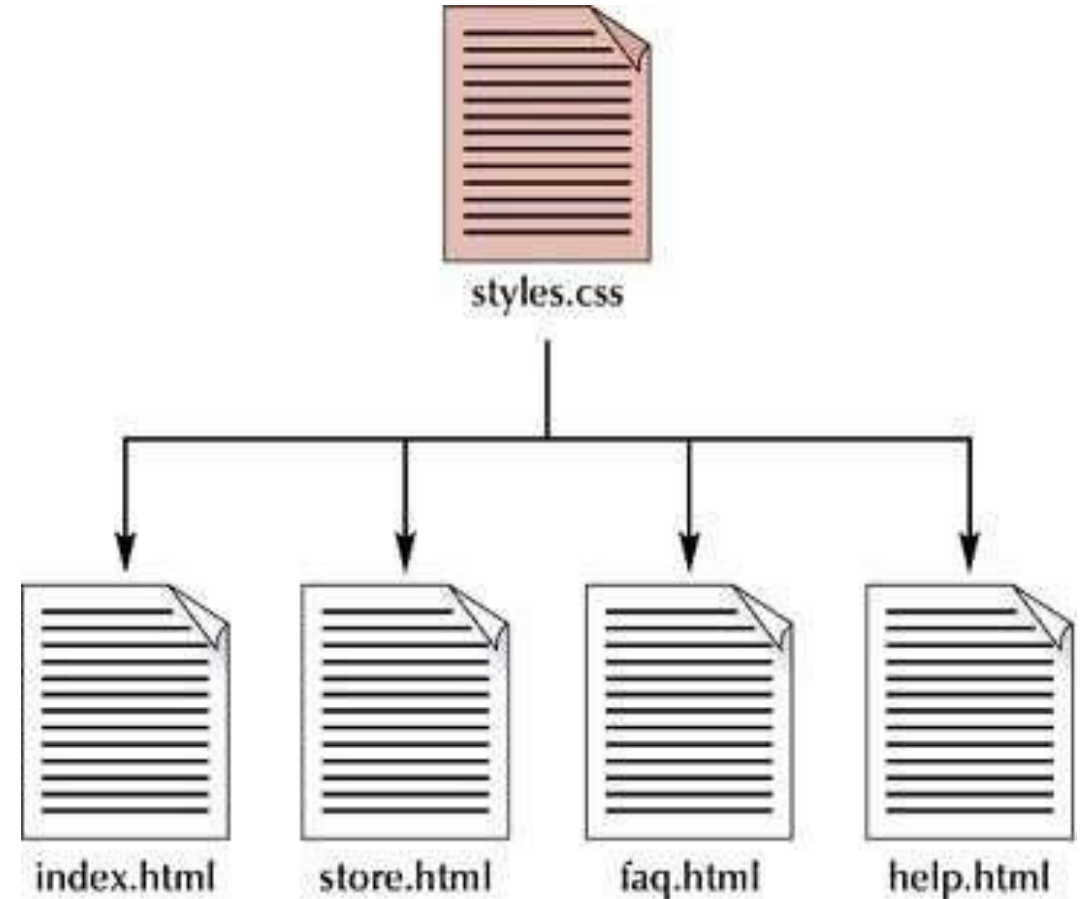
Introduction to Stylesheets

Introduction to CSS3

- ▶ Cascading Style Sheets 3 (CSS3) that allows you to specify the presentation of elements on a web page (e.g., fonts, spacing, sizes, colors, positioning) separately from the document's structure and content (section headers, body text, links, etc.).
- ▶ CSS is a set of rules for displaying markup content. CSS made the Separation of structure from presentation
 - ▶ multiple style sheets can be applied to the same Web page
 - ▶ Same style sheet can be applied to the multiple Web page
- ▶ Cascading:
 - ▶ Display rules "cascade" down
 - ▶ The most specific rule is used
- ▶ Styles Sheet:
- ▶ Rules are created as styles

Advantages of CSS3

- ▶ Saves time
- ▶ Easy to change
- ▶ Keep consistency(always behave in the same way)
- ▶ more control over layout
- ▶ Use styles with JavaScript => DHTML
- ▶ create a common format for all the Web pages



Types of CSS

- ▶ **Inline style sheet**
 - ▶ The style properties are added within the HTML tag itself
- ▶ **Embedded style sheet**
 - ▶ The set of style properties are embedded within the HTML document
- ▶ **External style sheet**
 - ▶ Common set of style properties are applied to all the web pages from an external .css file.

Each definition contains:

- A property
- A colon
- A value
- A semicolon to separate two or more values
- Can include one or more values

Selector{**property**:**value**;**property2**:**value2**;...}

h1 {font-size: 2pt; color: red}

Order of Precedence of Style Sheet

- ▶ Inline style sheets(lowest level)
- ▶ Embedded/Document Level sheets
- ▶ External style sheets(highest Level)
- ▶ Inline Style sheets have precedence over document style sheets which have precedence over external style sheet.
- ▶ When using multiple styles that conflict, which will be displayed?
- ▶ Order:
 - ▶ Inline style sheet
 - ▶ Embedded style sheet
 - ▶ External style sheet
 - ▶ Browser default

Style Specification Format

- ▶ Inline Style

Style="property0:value0;property1:value1;
.....
propertyZ:valueZ;"

- ▶ **Document Level**

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

- ▶ Each style rule in a rule list has two parts
- ▶ Selector—the tag or tags affected by the rule.
- ▶ List of property/value pairs
- ▶ Selector { property_1: value_1; property_2: value_2:... property_n: value_n;}

Style Specification Format

- ▶ External Style sheet

`<link href=URL rel="relation_type" type="link_type">`

- ▶ URL is the file.css

- ▶ Relation_type="stylesheet"
- ▶ Link_type="text/css"

- ▶ Do not include `<style>` tags

- ▶ Save the document as filename.css

Inline styles

- ▶ Add styles to each tag within the HTML file
- ▶ Use it when you need to format just a single section in a web page
- ▶ style information is directly attached to the HTML elements they affect
- ▶ Attribute style, followed by a colon and a value
- ▶ `<h1 style="color:red; font-family: sans-serif">India is my country</h1>`

```
<!DOCTYPE html>
<html>
  <head>    <title> sjc</title>
    <meta charset = "utf-8" /></head>
  <body>
    <p>Hello world from the web Master</p>
    <p style="font-size:50; font-family: Arial;font-weight:bold;color:green" >
      Welcome to HTML CSS Effects
    </p>
  </body>
</html>
```

Inline styles

```
<!DOCTYPE html>
<!-- Using inline styles -->
<html>
  <head>
    <meta charset = "utf-8">
    <title>Inline Styles</title>
  </head>
  <body>
    <p>This text does not have any style applied to it.</p>
    <p style = "font-size: 20pt;">This text has the
      <em>font-size</em> style applied to it, making it 20pt.
    </p>
    <p style = "font-size: 20pt; color: deepskyblue;">
      This text has the <em>font-size</em> and
      <em>color</em> styles applied to it, making it
      20pt and deep sky blue.</p>
  </body>
</html>
```



Inline styles

- ▶ Attribute style specifies an element's style. Each CSS property (font-size in this case) is followed by a colon and a value. the two properties, font-size and color are separated by a semicolon.
- ▶ set the given paragraph's color to deepskyblue. Hexadecimal codes may be used in place of color names.
- ▶ Figure contains the HTML standard color set.
- ▶ A complete list of HTML standard and extended colors at www.w3.org/TR/css3-color/

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

HTML standard colors and hexadecimal RGB values.

Embedded / internal/Document level

- ▶ A style is applied to the entire HTML file. Use it when you need to modify all instances of particular element (e.g., h1) in a web page

- ▶ Example

```
<style type="text/css">  
    h1 {color:red; font-size:20;font-family:monospace}
```

```
</style>
```

```
<head>
```

```
<title>Embedded Example</title>
```

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

```
</style>
```

```
</head>
```

A style declaration:

Selector {attribute1:value1; attribute2:value2; ...}

Selector = an element in a document (e.g., a header or paragraph)

Embedded / internal/Document level

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset = "utf-8">
    <title>Embedded Style Sheet</title>

    <!-- this begins the style sheet section -->
    <style type = "text/css">
      em      { font-weight: bold;
                color: black; }
      h1      { font-family: tahoma, helvetica, sans-serif; }
      p      { font-size: 12pt;
                font-family: arial, sans-serif; }
      .special { color: purple; }
    </style>
  </head>
  <body>
    <!-- this attribute applies the .special style class -->
    <h1 class = "special">Deitel & Associates, Inc.</h1>

    <p>Deitel & Associates, Inc. is an authoring and
      corporate training organization specializing in
      programming languages, Internet and web technology,
      iPhone and Android app development, and object
      technology education.</p>

    <h1>Clients</h1>
    <p class = "special"> The company's clients include many
      <em>Fortune 1000 companies</em>, government agencies,
      branches of the military and business organizations.</p>
  </body>
</html>
```



Embedded / internal/Document level

- ▶ Styles placed in the head apply to matching elements wherever they appear in the body.
- ▶ The style element's type attribute specifies the MIME (Multipurpose Internet Mail Extensions) type that describes the style element's content.
- ▶ CSS documents use the MIME type text/css

MIME type	Description
text/css	CSS documents
image/png	PNG images
text/javascript	JavaScript markup
text/plain	Plain text
image/jpeg	JPEG image
text/html	HTML markup

A few common MIME types.

```
<!DOCTYPE html>
<html >
<head>
<title>Getting Started</title>
<style type="text/css">
  h1 {font-family: sans-serif; color: orange}
  p{font-size: 1.5em;font-style:italic;background-color:grey}
</style>
</head><body><h1>Hello world using CSS</h1>
<p> Embedded Style is really interesting<p>
</body>
</html>
```


External style sheets

- ▶ Style sheets are a convenient way to create a document with a uniform theme.
- ▶ With external style sheets (i.e., separate documents that contain only CSS rules), you can provide a uniform look and feel to an entire website (or to a portion of one).
- ▶ You can also reuse the same external style sheet across multiple websites.
- ▶ Different pages on a site can all use the same style sheet. When changes to the styles are required, you need to modify only a single CSS file to make style changes across all the pages that use those styles. This concept is sometimes known as **skinning**.
- ▶ While embedded style sheets separate content from presentation, both are still contained in a single file, preventing a web designer and a content author from conveniently working in parallel.
- ▶ External style sheets solve this problem by separating the content and style into separate files

External style sheets

- ▶ An external style sheet is a .css file containing the style definition (declaration)
- ▶ Use it when you need to control the style for an entire web site
- ▶ Example
- ▶ `h1, h2, h3, h4, h5, h6 {color : red; font-family : sans-serif}`
- ▶ Save this in a new document using a .css extension
- ▶ Creating an External Style Sheet
 - ▶ Open a new blank document in Notepad
 - ▶ Type style declarations
`h1 {color:red; font-family:sans-serif;}`
 - ▶ Do not include `<style>` tags
 - ▶ Save the document as filename.css
- ▶ Linking to Style Sheets
 - ▶ Open an HTML file
 - ▶ Between `<head>` and `</head>` add
`<link href=URL rel="relation_type" type="link_type">`
 - ▶ URL is the file.css
 - ▶ `Relation_type="stylesheet"` and `Link_type="text/css"`
 - ▶ Save this file and the .css file in the same web server directory

Linking External style sheets

```
<head>  
<title>Getting Started</title>  
<link href="scraps.css"  
      rel="stylesheet"  
      type="text/css" />  
</head>
```

html file

```
h1 {font-family: sans-serif;  
    color: orange}  
b {color: blue}
```

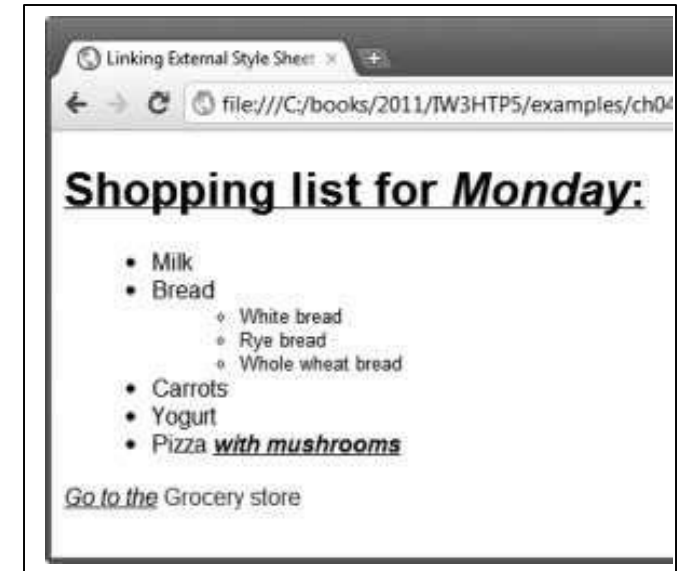
Text file of css named "stylesheet"

Linking External style sheets

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset = "utf-8">
    <title>Linking External Style Sheets</title>
    <link rel = "stylesheet" type = "text/css"
      href = "styles.css">
  </head>
  <body>
    <h1>Shopping list for <em>Monday</em>:</h1>
    <ul>
      <li>Milk</li>
      <li>Bread
        <ul>
          <li>white bread</li>
          <li>Rye bread</li>
          <li>Whole wheat bread</li>
        </ul>
      </li>
      <li>Carrots</li>
      <li>Yogurt</li>
      <li>Pizza <em>with mushrooms</em></li>
    </ul>
    <p><em>Go to the</em>
      <a class = "nodec" href = "http://www.deitel.com">
        Grocery store</a>
    </p>
  </body>
</html>
```

```
body { font-family: arial, helvetica, sans-serif; }
a.nodec { text-decoration: none; }
a:hover { text-decoration: underline; }
li em { font-weight: bold; }
h1, em { text-decoration: underline; }
ul { margin-left: 20px; }
ul ul { font-size: .8em; }
```

External style sheet.



Conflict Resolution

Conflicting styles

- ▶ Styles may be defined by a user, an author or a user agent.
- ▶ A user is a person viewing your web page, you're the author—the person who writes the document—and the user agent is the program used to render and display the document (e.g., a web browser).
- ▶ Styles cascade (and hence the term “Cascading Style Sheets”), or flow together, such that the ultimate appearance of elements on a page results from combining styles defined in several ways.
- ▶ Styles defined by the user take precedence over styles defined by the user agent.
- ▶ Styles defined by authors take precedence over styles defined by the user.
- ▶ Most styles defined for parent elements are also inherited by child (nested) elements

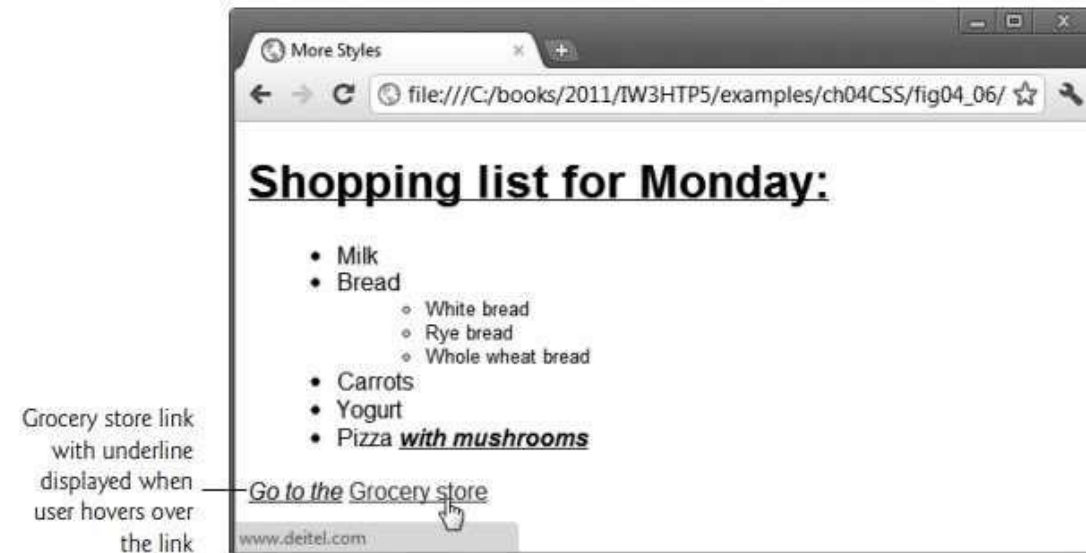
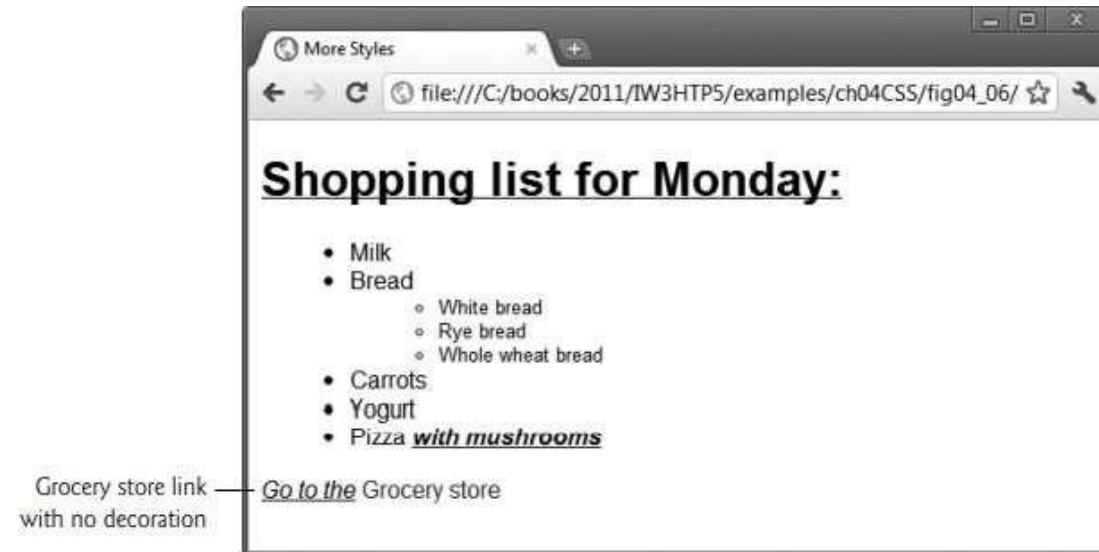
Conflict resolution

- ▶ There are certain properties that you don't want to be inherited.
- ▶ For example, the background-image property allows you to set an image as the background of an element.
- ▶ If the body element is assigned a background image, we don't want the same image to be in the background of every element in the body of our page. Instead, the background-image property of all child elements retains its default value of none.
- ▶ There are some rules for **resolving conflicts** between styles defined for elements and styles inherited from parent and ancestor elements
- ▶ Properties defined for child and descendant elements have a higher specificity than properties defined for parent and ancestor elements.
- ▶ Conflicts are resolved in favor of properties with a higher specificity, so the child's styles take precedence

```

<!DOCTYPE html>
<html> <head>
  <meta charset = "utf-8">
  <title>More Styles</title>
  <style type = "text/css">
    body { font-family: arial, helvetica, sans-serif; }
    a.nodect { text-decoration: none; }
    a:hover { text-decoration: underline; }
    li em { font-weight: bold; }
    h1, em { text-decoration: underline; }
    ul { margin-left: 20px; }
    ul ul { font-size: .8em; }
  </style>
</head>
<body>
  <h1>Shopping list for Monday:</h1>
  <ul>
    <li>Milk</li>
    <li>Bread
      <ul> <li>white bread</li>
        <li>Rye bread</li>
        <li>Whole wheat bread</li> </ul> </li>
    <li>Carrots</li>
    <li>Yogurt</li>
    <li>Pizza <em>with mushrooms</em></li>
  </ul>
  <p><em>Go to the</em>
    <a class = "nodect" href = "http://www.deitel.com">
      Grocery store</a> </p> </body> </html>

```



Style Sheets-Exploring CSS Selectors

Exploring CSS Selectors

- ▶ CSS selectors are used to "find" (or select) the HTML elements you want to style.
- ▶ There are six different selector forms
 - ▶ Simple selector
 - ▶ Class selector
 - ▶ Generic selector
 - ▶ Id selector
 - ▶ Universal selector
 - ▶ Pseudo classes

Exploring CSS Selectors- Simple selector

- ▶ Simple selector form is a list of style rules, and **property values in the rule apply to all occurrences** of the all of the named element
- ▶ The **selector is a tag name or a list of tag names, separated by commas**
- ▶ Ex) `h1, h3 { font-size: 24pt ;}`
`h2 { font-size: 20pt ;}`
- ▶ **Contextual selectors:** Selectors can also specify that the **style should apply only to elements in certain positions** in the document
- ▶ Ex) selector applies its style to the content of italic elements that are descendants of bold elements in the body of the document.
`body b i {font-size: 24pt ;}`
- ▶ Also called as **descendant selectors**.

Exploring CSS Selectors- class selector -Example

```
<!DOCTYPE html>
<html lang = "en"> <head>   <title> sjc</title>   <meta charset = "utf-8" />
<style type = "text/css">
p.regtext {font-family: Times;font-size: 14pt; width: 800px}
p.abstext { position: absolute;top: 125px;left: 50px;font-family: Times;font-size: 24pt;font-style: italic; width:
500px}
</style> </head> <body>
<p class = "abstext"> APPLES ARE GOOD FOR YOU </p>
<p class = "regtext"> Apple is the common name for any tree of the genus Malus, of the family Rosaceae. Apple
trees grow in any of the temperate areas of the world. Some apple blossoms are white, but most have stripes or
tints of rose. Some apple blossoms are bright red. Apples have a firm and fleshy structure that grows from the
blossom. The colors of apples range from green to very dark red. The wood of apple trees is fine-grained and
hard. </p>
<p class = "abstext"> ORANGES ARE GOOD IN SUMMER </p>
<p class = "regtext"> Oranges the common name for any tree of Orange. trees grow in any of the temperate
areas of the world. Some oranges blossoms are white, but most have stripes or tints of rose. </p>
</body>
</html>
```

Generic Selector

- ▶ A generic class can be defined if you want a **style to apply to more than one kind of tag**.
- ▶ A generic class must be named, and the name must begin with a period without a tag
- ▶ Example: `.really-big { ... }`
`<h1 class = "really-big"> ADMISSION</h1>...`
`<p class = "really-big"> BTECH ADMISSION 2022... </p>`
`<h2 class = "really-big"> BHRM ADMISSION 2022... </h2>`
`<h3 class = "really-big"> PG ADMISSION 2022... </h3>`

Generic Selector

```
<!DOCTYPE html> <html lang = "en">
<head> <title> sjc</title>    <meta charset = "utf-8" />
<style type = "text/css">
.regtext {font-family: Times; font-size: 14pt; width: 800px}
.admstext { position: absolute;top: 125px; left: 50px; font-family: Times; font-size: 24pt; font-style: italic; width: 500px}
</style>
</head>
<body>
  <h1 class = "admstext"> ADMISSION</h1>
  <p class = "regtext">VJCET structure that grows from the blossom. Minority catholic institution .provides Btech
and MTECh courses</p>
  <p class = "admstext"> BTECH ADMISSION 2022... </p>
  <p class = "regtext"> Hotel management institution </p>
  <h2 class = "admstext"> BHRM ADMISSION 2022... </h2>
  <h3 class = "admstext"> PG ADMISSION 2022... </h3>
  <p class = "admstext"> PG courses MTECH,MBA,MTECH available </p>
</body> </html>
```

Id Selector

- ▶ An id selector allow the **application of a style to one specific element**.
- ▶ Style specified in the id selector applies to the element with the given id.
- ▶ To create an ID for a specific tag, use the property:
<tag id="id_name">
- ▶ To apply a style to a specific ID, use:
#id_name {style attributes and values}
- ▶ The general form of an id selector is as follows :
#specific-id {property-value list}

Example:

#section14 {font-size: 20} specifies a font size of 20 points to the element
<h2 id = "section14"> Alice in wonderland </h2>

Id Selector

```
<!DOCTYPE html>
<html><head>
<style>
#para1 {
  text-align: center;
  color: green;
  font-weight: bold;
}
</style>
</head><body>
<p id="para1">Hello World!</p>
<p>This paragraph is not affected by the style.</p>
</body>
</html>
```

Hello World!

This paragraph is not affected by the style.

Universal Selector

- ▶ The **universal selector, denoted by an asterisk(*)**, which applies style to all elements in the document.
- ▶ For example:
*{color: red;}
makes all elements in the document red. Is not often useful.

Pseudo Classes

- ▶ A pseudo-class is used to define a special state of an element.
- ▶ For example, it can be used to:
 - ▶ Style an element when a user moves the mouse over it
 - ▶ Style visited and unvisited links differently
 - ▶ Style an element when it gets focus
- ▶ Pseudo classes are styles that apply when something happens, dynamically instead of simply displaying the target element
- ▶ Names of pseudo classes begin **with colons** **hover classes** apply when the mouse cursor is over the element **focus classes** apply **when an element has focus** i.e. the mouse cursor is over the element and the left mouse button is clicked
- ▶ These two pseudo classes are supported by FX2 but IE7 supports only hover

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

```
div:hover {  
    background-color: blue;  
}
```

Pseudo classes

```
<html >
<head> <title> Style for Input Type  Text </title>
<style type = "text/css">
    input:hover {color: red;border-top:dotted;}
    input:focus {color: green;border-color:yellow;}
</style>
</head>
<body>
<form action = "">
    <p> Your name: <input type = "text" /> </p>
</form>
</body>
</html>
```

Anchor Pseudo Classes

- ▶ Links can be displayed in different ways.
- ▶ We can modify the way hyperlinks appear by creating style rules modifying the <a> tag with the following “pseudo-classes”:
- ▶ link
- ▶ visited
- ▶ hover
- ▶ active

a:link {color:#0000ff}

a:visited {color: #00ff00}

a:hover {color:fuschia; font-weight:bold}

a:active {font-size:30pt}

```
/* unvisited link */  
a:link {  
    color: #FF0000;  
}
```

```
/* visited link */  
a:visited {  
    color: #00FF00;  
}
```

```
/* mouse over link */  
a:hover {  
    color: #FF00FF;  
}
```

```
/* selected link */  
a:active {  
    color: #0000FF;  
}
```

Anchor Pseudo Classes

```
<!DOCTYPE html>
<html><head>
<style>
  /* unvisited link */
  a:link { color: red;}
  /* visited link */
  a:visited { color: green;}
  /* mouse over link */
  a:hover { color: hotpink;}
  /* selected link */
  a:active { color: blue;}
</style></head>
<body>
  <h2>Styling a link depending on state</h2>
  <p><b><a href="default.asp" target="_blank">This is a link</a></b></p>
</body>
</html>
```

CSS Properties - values

- ▶ CSS properties are the styles used on specified selectors.
- ▶ They are written before values in the CSS ruleset and are separated from property values by a colon.
- ▶ Different HTML selectors and elements have different properties.
- ▶ Some properties are universal and can be used on every selector.
- ▶ Others work only on specific selectors and under particular conditions
- ▶ There are many properties and their values for HTML selectors.
- ▶ There are *520 distinct property names from 66 technical reports and 66 editor's drafts*.
- ▶ These properties are common because they are frequently used in all CSS documents and can be applied to different selectors.
- ▶ One unique thing about properties is that they have more than one value attached to them.

CSS Property

- ▶ CSS properties are the styles used on specified selectors.
- ▶ five common properties to work with
 - ▶ List properties
 - ▶ Font properties
 - ▶ Border properties
 - ▶ Text properties
 - ▶ Color Properties
- ▶ CSS VALUES
- ▶ Values are written immediately after the colon that separates them from CSS properties.
- ▶ CSS values aren't just text; they come in different forms
 - ▶ Text
 - ▶ URLs, units, measurements, integers, strings, inherit, auto, none, etc.

CSS VALUES

- ▶ Values are written immediately after the colon that separates them from CSS properties.
- ▶ CSS values aren't just text; they come in different forms
- ▶ **Keywords property values** are used when there are only a few possible values and they are **predefined** (not case sensitive)
 - ▶ Eg: small, large, medium.
- ▶ **Number values** can be **integer or sequence of digits with decimal points** and a + or – sign.
- ▶ **Length value** are specified as **number values** that are followed immediately by a two character abbreviation of a unit name.
 - ▶ There can be no space between the number and the unit name.
px for pixels, pt for points, pc for picas (12 points) , in for inches, cm for centimeters, mm for millimeters,

CSS VALUES

- ▶ **Percentage** - just a number followed immediately by a percent sign: eg: font size set to 85% means new font size will be 85% of the previous font size value.
- ▶ **URL values:** URL property values use a form that is slightly different from references to URLs in links.
 - ▶ `url(protocol://server/pathname)`
 - ▶ No space should be left between URL and the left parenthesis.
- ▶ **Colors** : Color name `rgb(n1, n2, n3)`.

Hex form: `#B0E0E6` stands for powder blue color.

CSS Property- values-TEXT properties

Properties	Description	Values
color	Sets the color of a text	<i>Hex, RGB, keyword</i>
text-transform	Sets the capitalization of the text	<i>uppercase, lowercase, capitalize, none</i>
text-align	Sets the alignment of the text on the screen	<i>right, left, center, justify</i>
letter-spacing	Sets the spacing between text characters	<i>normal, length</i>
text-decoration	Sets the decoration added to the text	<i>none, underline, line-through, overline</i>

CSS Property- values-Border properties

Properties	Description	Values
border	Sets the shorthand combination for border-width, border-style and border-color	<i>border-width, border-style, border-color</i>
border-color	Sets the color for the border	<i>Keyword, RGB, Hex, transparent, inherit</i>
border-radius	Sets the radius of the four corners of an element's border	<i>length, percentage, initial, inherit</i>
border-style	Sets the style for an element's border	<i>none, hidden, dotted, solid, dashed, double, groove, inset, outset, ridge, initial, inherit</i>
border-image	Sets an image as an element's border	<i>border-image-source, border-image-width, border-image-slice, border-image-repeat, border-image-outset, initial, inherit</i>

CSS Property- values-Font properties

Properties	Description	Values
font	Sets the shorthand for all the font specifications	<i>font-style, font-variant, font-weight, font-size/line-height, font-family, caption, icon, menu, message-box, small-caption, status-bar, inherit</i>
font-weight	Sets the weight of a font	<i>normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900, inherit</i>
font-style	Sets the style of a font	<i>Normal, italic, oblique, initial, inherit</i>

- ▶ The font-family property is used to specify a list of font name. The browser will use the first font in the list that it supports.
- ▶ **font-family: Arial, Helvetica, Courier**
- ▶ Generic fonts: They can be specified as the font family value for example :**serif, sans-serif, cursive, fantasy, and monospace** (defined in CSS).
- ▶ If a font name that has more than one word, it should be single-quoted Eg: font-family: 'Times New Roman'

CSS Property- values-Font properties

- ▶ Font Shorthand
- ▶ If more than one font property is to be specified then the values may be stated in a list as the value of the font property .
Eg: `font: bold 24pt 'Times New Roman' Palatino Helvetica`
- ▶ The order which browser follows is **last must be font name, second last font size** and then the **font style, font variant and font weight etc can be in any order** but before the font size and names.
- ▶ **Font-size** -Possible values: a length number or a name, such as **smaller, xx-large, medium , large etc.**
- ▶ **Font-variant** -The default value of the font-variant property is **normal**, can be set to **small-caps** to specify small capital characters.
- ▶ **Font-style** -The property is most commonly used to specify **italic**,
- ▶ **Font-weight** -The property is used to specify **the degree of boldness**.
- ▶ **Text Decoration**-**line-through, overline , underline, none**
- ▶ **Text Spacing:** **letter spacing , word spacing , line-height**

CSS Property- values-List properties

Properties	Description	Values
list-style	Shorthand combination for list-style-type, list-style-position, and list-style-image	<i>list-style-type, list-style-position, list-style-image, inherit</i>
list-image	Sets an image as the list-item marker	<i>none, url, initial, inherit</i>
list-type	This sets the type of list-item marker	<i>none, disc, circle, square, decimal, decimal-leading-zero, armenian, georgian, lower-alpha, upper-alpha, lower-greek, lower-latin, upper-latin, lower-roman, upper-roman, inherit</i>

CSS Property- values-List properties

- ▶ It is used to specify style of bullets or sequencing values in list items.
- ▶ The **list-style-type** of Unordered lists can be set to disc,circle,square or none.
- ▶ Bullet can be a disc (default), a square, or a circle

`<li style = "list-style-type: circle">BTECH `

- ▶ Could use an image for the bullets in an unordered list.

- ▶ **Example:**

`<li style = "list-style-image: url(book.jpg)">`

CSS Property- values-List properties

- ▶ When ordered lists are nested, it is best to use different kinds of sequence values for the different levels of nesting
- ▶ Decimal Arabic numerals 1, 2, 3, 4
- ▶ upper-alpha Uc letters A, B, C, D
- ▶ lower-alpha Lc letters a, b, c, d
- ▶ upper-roman Uc Roman I, II, III, IV
- ▶ lower-roman Lc Roman i, ii, iii, iv

```
<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>
```

CSS Property- values-color properties

- ▶ color –specify the foreground color of html elements

color:red

- ▶ background-color→set the background color of an element

background-color:grey

- ▶ Web Palette contains , one of 16 million different colors
- ▶ A set of 16 generic colors that are guaranteed to be displayable
 - ▶ Names for some:
 - ▶ blue, red, green, pink
 - ▶ Hexadecimal
 - ▶ #0000FF, #FF0000, #00FF00, #FF3399
 - ▶ RGB
 - ▶ rgb(0,0,255), rgb(255,0,0), rgb(0,255,0)
 - ▶ RGB%
 - ▶ rgb(0%,0%,100%), rgb(100%,0%,0%)

CSS Property- values-other properties

1. text-align property has the possible values, **left (the default), center, right, or justify.**

▶ we want **text to flow around another element - the float property.**

2. float property has the possible values, **left, right, and none (the default).**

▶ Float property is used to specify that text should flow around some element.

3. text-indent property used to indent the first line of a paragraph

CSS Property- values-Background

- ▶ CSS provides control over the backgrounds of block-level elements.
- ▶ CSS can set a background color or add background images to HTML5 element
- ▶ background-color→set the background color of an element

background-color:grey

- ▶ The **background-image** property is used to place an image in the background of an element
Repetition can be controlled.
- ▶ Background image can be replicated to fill the area of the element. This is known as tiling.

background-repeat property possible values:repeat (default), no-repeat, repeat-x, or repeat-y

background-position property. Possible values: top, center, bottom, left, or right.

CSS Property- values-Example

```
<html><head>  <title> VJCET </title> <meta charset = "utf-8" />
<style type = "text/css">
body { background-color:grey;background-image:url("agriculture.jfif");background-repeat:repeat-y;
      background-position:center;  }
img {float:left;}
</style> </head>
<body>
  <h1> ADMISSION</h1>.
   <p>Institution provides Btech and MTECh courses</p>
  <p > BTECH ADMISSION 2022... </p>
  <p > Hotel management institution . Minority catholic institution .provides BHRM
  courses</p> <h2 > BHRM ADMISSION 2022... </h2>
  <h3 > BHRM ADMISSION 2022... </h2><h3 > PG ADMISSION 2022... </h3>
  <p> PG courses MTECH,MBA,MTECH available </p>
</body> </html>
```

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset = "utf-8">
    <title>Background Images</title>
    <style type = "text/css">
      body { background-image: url(logo.png);
              background-position: bottom right;
              background-repeat: no-repeat;
              background-attachment: fixed;
              background-color: lightgrey; }
      p    { font-size: 18pt;
              color: Darkblue;
              text-indent: 1em;
              font-family: arial, sans-serif; }
      .dark { font-weight: bold; }
    </style>
  </head>
  <body>
    <p>
      This example uses the background-image,
      background-position and background-attachment
      styles to place the <span class = "dark">Deitel
      & Associates, Inc.</span> logo in the
      bottom-right corner of the page. Notice how the logo
      stays in the proper position when you resize the
      browser window. The background-color fills in where
      there is no image.
    </p>
  </body>
</html>

```



CSS Property- values-Background

▶ background-image Property

- ▶ The background-image property specifies the image URL for the image logo.png in the format url(fileLocation).
- ▶ You can also set the background-color property in case the image is not found

▶ background-position Property

- ▶ The background-position property places the image on the page. The keywords top, bottom, center, left and right are used individually or in combination for vertical and horizontal positioning.
- ▶ For example, to position the image as horizontally centered (positioned at 50 percent of the distance across the screen) and 30 pixels from the top, use

```
background-position: 50% 30px;
```

CSS Property- values-Background

▶ background-attachment

- ▶ fixed Property The next property setting, background-attachment: fixed , fixes the image in the position specified by background-position. Scrolling the browser window will not move the image from its position.
- ▶ The default value, scroll, moves the image as the user scrolls through the document.

▶ background-repeat Property

- ▶ The background-repeat property controls background image tiling, which places multiple copies of the image next to each other to fill the background.
- ▶ Here, we set the tiling to no-repeat to display only one copy of the background image.
- ▶ Other values include repeat (the default) to tile the image vertically and horizontally, repeat-x to tile the image only horizontally or repeat-y to tile the image only vertically.

▶ text-indent property to indent the first line of text in the element by a specified amount, in this case 1em.

▶ font-style property formats text is the font-style property, which allows you to set text to none, italic or oblique

Positioning Elements: Absolute Positioning, z-index

- ▶ The CSS position property is used to set position for an element
- ▶ CSS introduced the position property ,which gives you greater control over how document elements are displayed
- ▶ The position property specifies the type of positioning method used for an element.
- ▶ Elements are then positioned using the top, bottom, left, and right properties. However, these properties will not work unless the **position property is set first.**
- ▶ They also work differently depending on the position value.

Positioning Elements: Absolute Positioning, z-index

- ▶ Absolute Position: An element with `position: absolute;` will cause it to adjust its position with respect to its parent. If no parent is present, then it uses the document body as parent.

`position: absolute;`

- ▶ Relative Position: Setting the `top`, `right`, `bottom`, and `left` properties of an element with **`position: relative;`** property will cause it to adjust from its normal position. The other objects or elements will not fill the gap
- ▶ The element that is set to relative position can be shifted with respect to other elements in the document. The element is shifted using `top`, `right`, `bottom`, and `left` properties

Positioning Elements: Absolute Positioning, z-index

- ▶ An element with `position: absolute;` is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like `fixed`).
- ▶ However; if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling.
- ▶ Note: Absolute positioned elements are removed from the normal flow, and can overlap elements.

```
div.absolute {  
    position: absolute;  
    top: 80px;  
    right: 0;  
    width: 200px;  
    height: 100px;  
    border: 3px solid #73AD21;  
}
```

```

<!DOCTYPE html>
<html>
<head>
<style>
div.relative {
  position: relative;
  width: 400px;
  height: 200px;
  border: 3px solid #73AD21;
}
div.absolute {
  position: absolute;
  top: 80px;
  right: 0;
  width: 200px;
  height: 100px;
  border: 3px solid #73AD21;
}
</style>
</head>
<body>
<h2>position: absolute;</h2>
<p>An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed):</p>
<div class="relative">This div element has position: relative;
  <div class="absolute">This div element has position: absolute;</div>
</div>
</body>
</html>

```

position: absolute;

An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed):

This div element has position: relative;

This div element has position: absolute;

```
<!DOCTYPE html> <html> <head>
<meta charset = "utf-8">
<title>Relative Positioning</title>
<style type = "text/css">
    p { font-size: 1.3em;
        font-family: verdana, arial, sans-serif; }
    span { color: red;font-size: .6em;height: 1em; }
    .super { position: relative;top: -1ex; }
    .sub { position: relative;bottom: -1ex; }
    .shiftleft { position: relative;left: -1ex; }
    .shiftright {position: relative;right: -1ex; }
</style> </head>
<body>
    <p>The text at the end of this sentence <span class = "super">is in
superscript</span></p>
    <p>The text at the end of this sentence <span class = "sub">is in
subscript</span></p>
    <p>The text at the end of this sentence<span class = "shiftleft">is shifted
left</span></p>
    <p>The text at the end of this sentence<span class = "shiftright">is
shifted right</span></p> </body> </html>
```

Positioning Elements:Relative Positioning

The text at the end of this sentence is in superscript

The text at the end of this sentence is in subscript

The text at the end of this sentence is shifted left

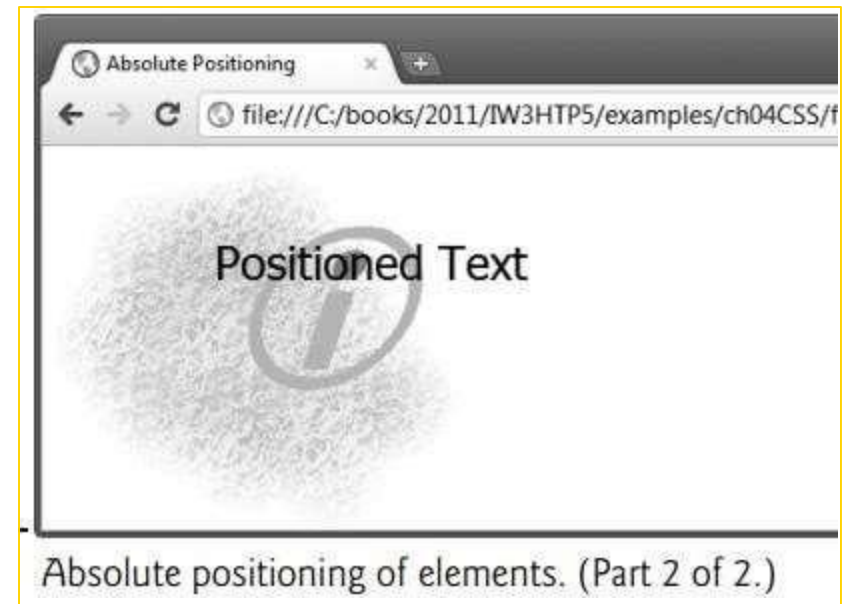
The text at the end of this sentence is shifted right

```

4 <!DOCTYPE html>
5 <html>
6   <head>
7     <meta charset = "utf-8">
8     <title>Absolute Positioning</title>
9     <style type = "text/css">
10       .background_image { position: absolute;
11                           top: 0px;
12                           left: 0px;
13                           z-index: 1; }
14       .foreground_image { position: absolute;
15                           top: 25px;
16                           left: 100px;
17                           z-index: 2; }
18       .text { position: absolute;
19               top: 25px;
20               left: 100px;
21               z-index: 3;
22               font-size: 20pt;
23               font-family: tahoma, geneva, sans-serif; }
24     </style>
25   </head>
26   <body>
27     <p><img src = "background_image.png" class = "background_image"
28         alt = "First positioned image" /></p>
29
30     <p><img src = "foreground_image.png" class = "foreground_image"
31         alt = "Second positioned image" /></p>
32
33     <p class = "text">Positioned Text</p>
34   </body>
35 </html>

```

Absolute positioning of elements. (Part 1 of 2.)



Positioning Elements: Absolute Positioning, z-index

- ▶ The z-index property allows you to layer overlapping elements.
- ▶ Elements that have higher z-index values are displayed in front of elements with lower z-index values.
- ▶ In this example, `.background_image` has the lowest z-index (1), so it displays in the background. The `.foreground_image` CSS rule (lines 14–17) gives the circle image (`foreground_image.png`, in lines 30–31) a z-index of 2, so it displays in front of `background_image.png`.
- ▶ The `p` element in line 33 is given a z-index of 3 in line 21, so its content (Positioned Text) displays in front of the other two.
- ▶ If you do not specify a z-index or if elements have the same z-index value, the elements are placed from background to foreground in the order in which they're encountered in the document.
- ▶ The default z-index value is 0.

Positioning Elements: Relative Positioning

- ▶ In relative positioning, elements are positioned relative to other elements
- ▶ An element with `position: relative;` is positioned relative to its normal position.
- ▶ Setting the `top`, `right`, `bottom`, and `left` properties of a relatively-positioned element will cause it to be adjusted away from its normal position. Other content will not be adjusted to fit into any gap left by the element.
- ▶ Setting the `position` property to `relative`, as in class `super` (lines 15–16), lays out the element on the page and offsets it by the specified `top`, `bottom`, `left` or `right` value.
- ▶ relative positioning keeps elements in the general flow of elements on the page, so positioning is relative to other elements in the flow.

Positioning Elements: Relative Positioning

- ▶ Setting the position property to relative, as in class super (lines 15–16), lays out the element on the page and offsets it by the specified top, bottom, left or right value.
- ▶ Class super (lines 15–16) lays out the text at the end of the sentence as superscript, and
- ▶ class sub (lines 17–18) lays out the text as subscript relative to the other text.
- ▶ Class shiftright (lines 19–20) shifts the text at the end of the sentence left and class shiftright (lines 21–22) shifts the text right

Span and Div

- ▶ `` and `<div>` are tags that let you select a group of elements and apply styles to them
 - ▶ `` is an inline tag
 - ▶ no breaks are added before or after `` ``
 - ▶ `<div>` is a block tag
 - ▶ a break is usually added by the browser before and after the `<div>` `</div>` tags
 - ▶ The `` tag is similar to other HTML tags,
 - ▶ they can be nested and they have `id` and `class` attributes
 - ▶ Another tag that is useful for style specifications:
`<div>` - Used to create document sections (or divisions) for which style can be specified
-

Span and Div

```
<html> <head>
<style>
div { line-height: 20px;
      margin: 30px;
      padding-bottom: 20px;
      text-align: justify;
      width: 140px;
      color: red; }
</style> </head>
<body> <p>A div element is displayed like this:
<div>This is some text in a div element.This is some text in a
      div element.This is some text in a div element.This is some
      text in a div element.This is some text in a div element.This
      is some text in a div element.</div>
Change the default CSS settings to see the effect.
</p> </body> </html>
```

A div element is displayed like this:

This is some text in a
div element.This is
some text in a div
element.This is some
text in a div
element.This is some
text in a div
element.This is some
text in a div element.

Change the default CSS settings to see the effect.

```

<!DOCTYPE html >
<html >
<head> <title>Selectors.html</title>
    <style type="text/css">
        .spanred {font-size:24pt;font-family:Arial;color:red;}
        .spanbrown {font-size:20pt;font-family:Arial;color:brown;}
    </style>
</head>
<body>

```

<p>Markup language refers to the traditional way of marking up a document. It determines the structure and meaning of textual elements . There are two types of markup languages.

```
<span class="spanred">Specific Markup Language </span>
```

It is used to generate the code that is specific to a particular application. Examples are

```
<span class="spanbrown">Generalized Markup Language</span>
```

It is generated to solve some problems associated with porting documents from one platform and operating system configuration to another</p>

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

Markup language refers to the traditional way of marking up a document. It determines the structure and meaning of textual elements .It consists of codes and tags that are added to the text to change the look or meaning of text or document. There are two types of markup languages.

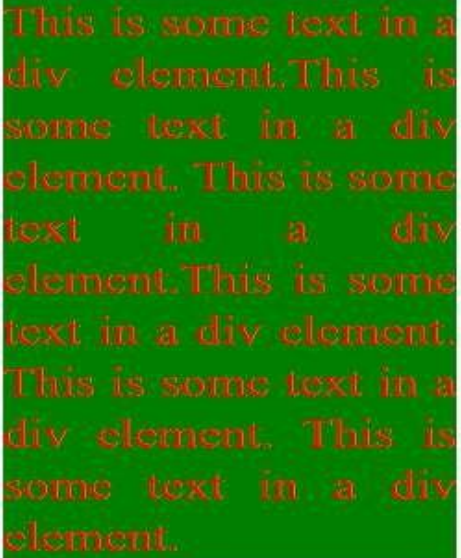
Specific Markup Language It is used to generate the code that is specific to a particular application. Examples are

Generalized Markup Language It is generated to solve some problems associated with porting documents from one platform and operating system configuration to another .

Span and Div

```
<html> <head>
<style>
div {
    line-height: 20px;
    margin: 30px;
    text-align: justify;
    width: 140px;
    color: red;
    background-color: green;
}
</style> </head> <body> <p>A div element is displayed
like this:
<div>This is some text in a div element.This is some text
in a div element.This is some text in a div element.This is
some text in a div element.This is some text in a div
element.This is some text in a div element.</div>Change
the default CSS settings to see the
effect.</p> </body> </html>
```

A div element is displayed like this:



This is some text in a
div element.This is
some text in a div
element. This is some
text in a div
element.This is some
text in a div element.
This is some text in a
div element. This is
some text in a div
element.

Change the default CSS settings to see the effect.

Box Model

Box Model



When the browser renders an element using the box model, the content is surrounded by padding, a border and a margin

Box model

- ▶ The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content.
- ▶ **Content** - The content of the box, where text and images appear
- ▶ **Padding** - Clears an area around the content. The amount of space between the content of an element and its border, known as padding. The padding is transparent
- ▶ **Border** - A border that goes around the padding and content
- ▶ **Margin** - Clears an area outside the border. The space between the border and an adjacent element known as margin. The margin is transparent
- ▶ The box model allows us to add a border around elements, and to define space between elements.

Box model

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  background-color: lightgrey;
  width: 300px;
  border: 15px solid green;
  padding: 50px;
  margin: 20px;
}
</style>
</head>
<body>
<h2>Demonstrating the Box Model</h2>
<div>Google was founded on September 4, 1998, by Larry Page and Sergey Brin while they
were PhD students at Stanford University in California. Together they own about 14% of
its publicly listed shares and control 56% of the stockholder voting power through super-
voting stock. The company went public via an initial public offering (IPO) in 2004. In
2015, Google was reorganized as a wholly owned subsidiary of Alphabet Inc. Sundar Pichai
was appointed CEO of Google on October 24, 2015, replacing Larry Page, who became the CEO
of Alphabet. On December 3, 2019, Pichai also became the CEO of Alphabet</div>
</body>
</html>
```

Demonstrating the Box Model

Google was founded on September 4, 1998, by Larry Page and Sergey Brin while they were PhD students at Stanford University in California. Together they own about 14% of its publicly listed shares and control 56% of the stockholder voting power through super-voting stock. The company went public via an initial public offering (IPO) in 2004. In 2015, Google was reorganized as a wholly owned subsidiary of Alphabet Inc. Sundar Pichai was appointed CEO of Google on October 24, 2015, replacing Larry Page, who became the CEO of Alphabet. On December 3, 2019, Pichai also became the CEO of Alphabet

Box model-controlling the margin

- ▶ To define the margins of an element, use:
 - ▶ **margin: *value***
 - ▶ where *value* = a length value ("em" is often used), a percentage (a margin proportional to the element's width, or auto
- ▶ To set margins on a side, use:
 - ▶ **margin-top**
 - ▶ **margin-right**
 - ▶ **margin-bottom**
 - ▶ **margin-left**
- ▶ E.g., `body {margin-left:12px; margin-right:3%; margin-top:10px; margin-bottom:1em}`
 - ▶ ("em" is often used), a margin proportional to the element's width, or auto

Box model-setting the padding

- ▶ To define padding, use:
 - ▶ **padding: *value***
 - ▶ where *value* = a length value or a percentage (a padding proportional to the element's width)
- ▶ To set padding on a side, use:
 - ▶ **padding-top**
 - ▶ **padding-right**
 - ▶ **padding-bottom**
 - ▶ **padding-left**

Box model-formatting the border

- ▶ Border can be set in three ways:
 - ▶ **border-width**(thin,thick,medium or length in pixels)
 - ▶ **border-style** (dotted,dashed,solid,double.default-none)
 - ▶ **border-color**
 - ▶ **Short hands for setting border**
 - ▶ **Ex: border:5px solid blue ;**
 - ▶ `td,th { border:thin double black ; }`
- ▶ To set the border, use:
 - ▶ *border : width_value style color*
- ▶ To set borders on a side, use:
 - ▶ border-top
 - ▶ border-bottom
 - ▶ border-left
 - ▶ border-right

Box model-formatting the border

- ▶ Border can be set in three ways:
 - ▶ **border-width**(thin,thick,medium or length in pixels)
 - ▶ **border-style** (dotted,dashed,solid,double.default-none)
 - ▶ **border-color**
 - ▶ **Short hands for setting border**
 - ▶ **Ex: border:5px solid blue ;**
 - ▶ `td,th { border:thin double black ; }`
- ▶ To set the border, use:
 - ▶ *border : width_value style color*
- ▶ To set borders on a side, use:
 - ▶ border-top
 - ▶ border-bottom
 - ▶ border-left
 - ▶ border-right

Box model-formatting the border

```
<style type="text/css">
table{
    padding-left:20px;
    padding-right:10pt;
    padding-top:3%;
    padding-bottom:1%;
    border-left:dotted;
    border-right:double;
    border-top:oblique;
    border-bottom:solid;
    border-width:10px;
    border-color:red
}
</style>
```

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset = "utf-8">
5     <title>Borders</title>
6     <style type = "text/css">
7       div { text-align: center;
8             width: 50%;
9             position: relative;
10            left: 25%;
11            border-width: 6px; }
12     .thick { border-width: thick; }
13     .medium { border-width: medium; }
14     .thin { border-width: thin; }
15     .solid { border-style: solid; }
16     .double { border-style: double; }
17     .groove { border-style: groove; }
18     .ridge { border-style: ridge; }
19     .dotted { border-style: dotted; }
20     .inset { border-style: inset; }
21     .outset { border-style: outset; }
22     .dashed { border-style: dashed; }
23     .red { border-color: red; }
24     .blue { border-color: blue; }
25   </style>
26 </head>
27 <body>
28   <div class = "solid">Solid border</div><hr>
29   <div class = "double">Double border</div><hr>
30   <div class = "groove">Groove border</div><hr>
31   <div class = "ridge">Ridge border</div><hr>
32   <div class = "dotted">Dotted border</div><hr>
33   <div class = "inset">Inset border</div><hr>
34   <div class = "thick dashed">Thick dashed border</div><hr>
35   <div class = "thin red solid">Thin red solid border</div><hr>
36   <div class = "medium blue outset">Medium blue outset border</div>
37 </body>
38 </html>

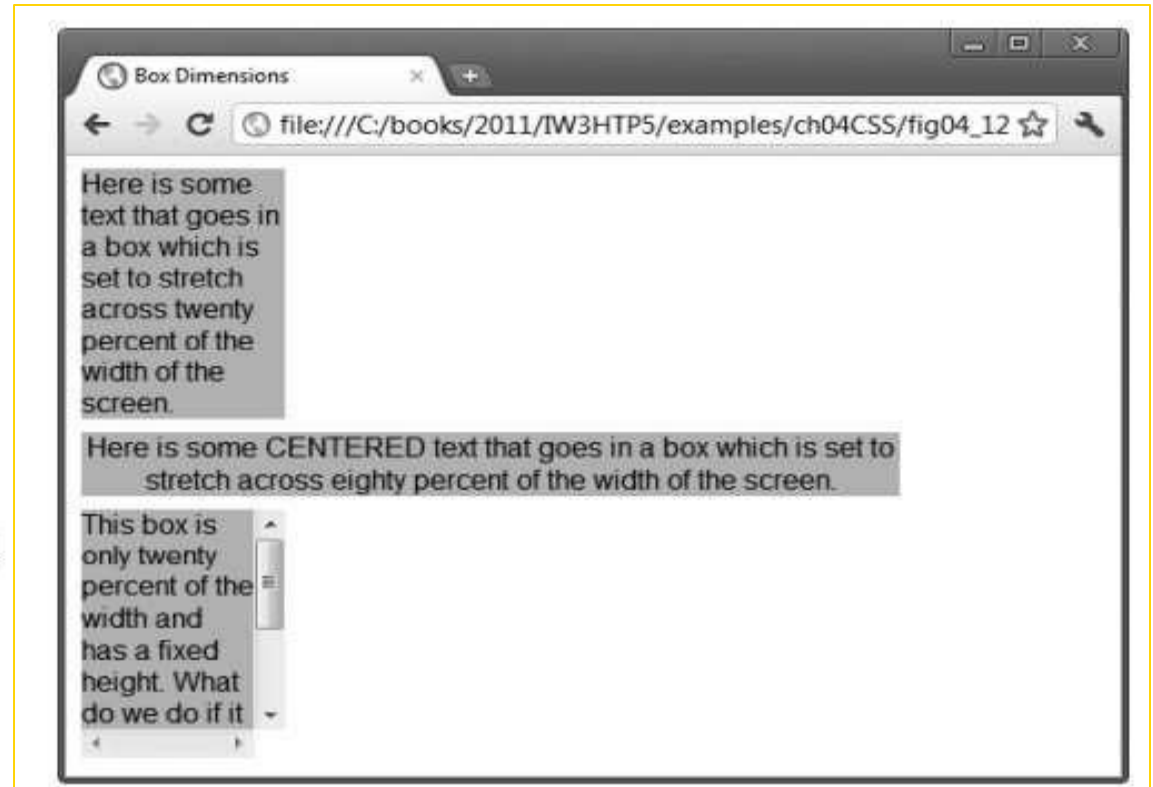
```



Element Dimensions

- ▶ In addition to positioning elements, CSS rules can specify the actual dimensions of each element.

```
1  <!DOCTYPE html>
2
3
4
5  <html>
6    <head>
7      <meta charset = "utf-8">
8      <title>Box Dimensions</title>
9      <style type = "text/css">
10        p { background-color: lightskyblue;
11             margin-bottom: .5em;
12             font-family: arial, helvetica, sans-serif; }
13      </style>
14    </head>
15    <body>
16      <p style = "width: 20%">Here is some
17        text that goes in a box which is
18        set to stretch across twenty percent
19        of the width of the screen.</p>
20
21      <p style = "width: 80%; text-align: center">
22        Here is some CENTERED text that goes in a box
23        which is set to stretch across eighty percent of
24        the width of the screen.</p>
25
26      <p style = "width: 20%; height: 150px; overflow: scroll">
27        This box is only twenty percent of
28        the width and has a fixed height.
29        What do we do if it overflows? Set the
30        overflow property to scroll!</p>
31    </body>
32  </html>
```

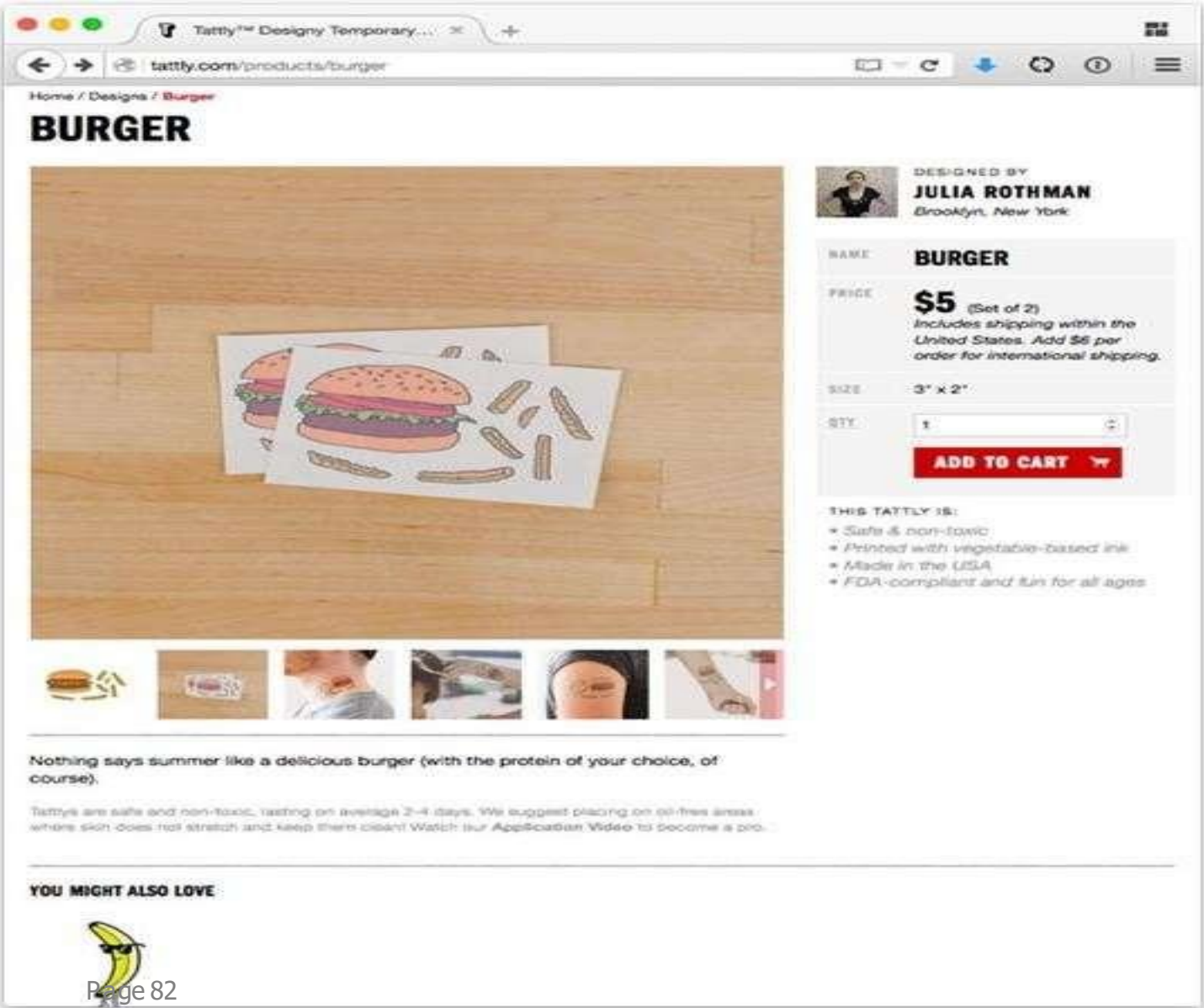


Media Types and media queries

Responsive web design

- ▶ **Responsive web design** is about creating web sites which **automatically adjust themselves** to look good on all devices, **from small phones to large desktops**
- ▶ Responsive web design provides an optimal experience, easy reading and easy navigation with a minimum of resizing on different devices such as desktops, mobiles and tabs).
- ▶ Responsive web design makes your web page look good on all devices.
- ▶ Responsive web design uses only HTML and CSS.
- ▶ Web pages should not leave out information to fit smaller devices, but rather adapt its content to fit any device:
- ▶ It is called responsive web design when you use CSS and HTML to resize, hide, shrink, enlarge, or move the content to make it look good on any screen.

Responsive web design



Media types and media queries

- ▶ CSS media types allow you to decide what a page should look like, depending on the kind of media being used to display the page.
- ▶ Media queries are a key part of responsive web design, as they **allow you to create different layouts depending on the size of the viewport**
- ▶ The @media rule, introduced in CSS2, made it possible to define different style rules for different media types.
- ▶ Examples: You could have one set of style rules for computer screens, one for printers, one for handheld devices, one for television-type devices, and so on.
- ▶ The most common media type for a web page is the screen media type, which is a standard computer screen.
- ▶ Other media types in CSS include handheld, braille, speech and print

Media types and media queries

- ▶ The handheld medium is designed for mobile Internet devices such as smartphones, while braille is for machines that can read or print web pages in braille.
- ▶ speech styles allow you to give a speech-synthesizing web browser more information about the content of a page.
- ▶ The print media type affects a web page's appearance when it's printed.
- ▶ For a complete list of CSS media types, see
<http://www.w3.org/TR/REC-CSS2/media.html#media-types>
- ▶ Media types allow you to decide how a page should be presented on any one of these media without affecting the others
- ▶ Media types allow you to decide how a page should be presented on any one of these media without affecting the others.
- ▶ Example that applies one set of styles when the document is viewed on all media (including screens) other than a printer, and another when the document is printed.


```

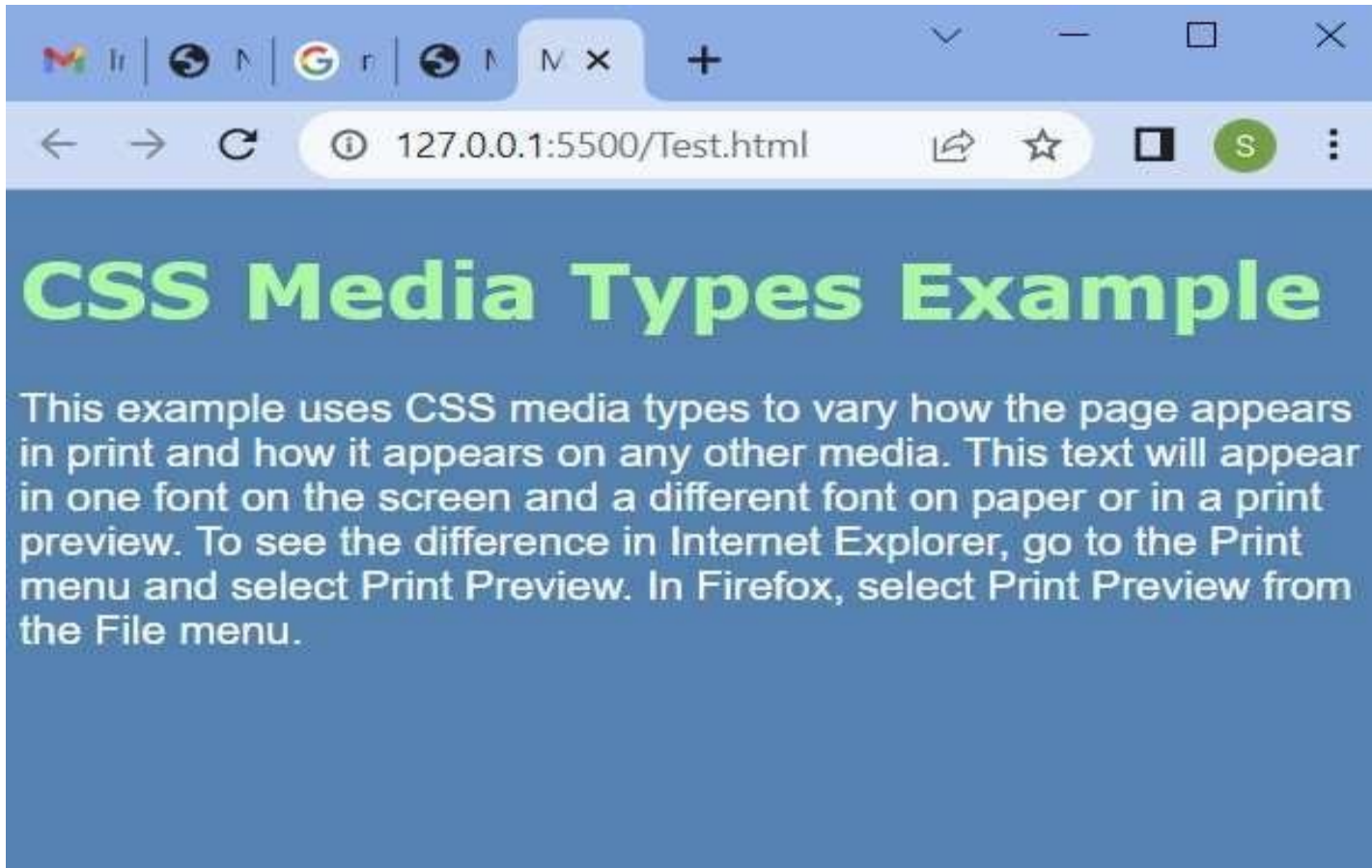
1  <!DOCTYPE html><html><head>
2      <meta charset="utf-8">
3      <title>Media Types</title>
4  </head>
5  <style type="text/css">
6      @media all{
7          body {
8              background-color: steelblue;
9          }
10
11         h1 {
12             font-family: verdana, helvetica, sans-serif;
13             color: palegreen;
14         }
15
16         p {
17             font-size: 12pt;
18             color: white;
19             font-family: arial, sans-serif;
20         }
21     }
22     /* End @media all declaration. */

```

```

23     @media print
24     {
25         body {background-color: white;}
26         h1 {color: seagreen;}
27         p {font-size: 14pt;
28             color: steelblue;
29             font-family: "times new roman", times, serif;}
30     }
31     /* End @media print declaration. */
32 </style>
33 <body><h1>CSS Media Types Example</h1>
34     <p> This example uses CSS media types to vary how the page
35         appears in print and how it appears on any other media.
36         This text will appear in one font on the screen and a
37         different font on paper or in a print preview. To see
38         the difference in Internet Explorer, go to the Print
39         menu and select Print Preview. In Firefox, select Print
40         Preview from the File menu.
41     </p>
42 </body>
43 </html>

```



10/20/22, 4:04 PM

Media Types

CSS Media Types Example

This example uses CSS media types to vary how the page appears in print and how it appears on any other media. This text will appear in one font on the screen and a different font on paper or in a print preview. To see the difference in Internet Explorer, go to the Print menu and select Print Preview. In Firefox, select Print Preview from the File menu.

Media queries

- ▶ Media query is a CSS technique introduced in CSS3.
- ▶ Media queries can be used to check many things, such as:
 - ▶ width and height of the viewport
 - ▶ width and height of the device
 - ▶ orientation (is the tablet/phone in landscape or portrait mode?)
 - ▶ resolution
- ▶ Using media queries are a popular technique for delivering a tailored style sheet to desktops, laptops, tablets, and mobile phones
- ▶ A media query consists of a media type and can contain one or more expressions, which resolve to either true or false.
- ▶ It uses the @media rule to include a block of CSS properties only if a certain condition is true.

```
@media not|only mediatype and (expressions) {  
    CSS-Code;  
}
```

Media queries

```
@media not|only mediatype and (expressions) {  
    CSS-Code;  
}
```

- ▶ meaning of the **not**, **only** and **and** keywords:
- ▶ **not**: The not keyword inverts the meaning of an entire media query.
- ▶ **only**: The only keyword prevents older browsers that do not support media queries with media features from applying the specified styles. **It has no effect on modern browsers.**
- ▶ **and**: The and keyword combines a media feature with a media type or other media features.
- ▶ They are all optional. However, if you use **not** or **only**, you must also specify a media type.
- ▶ Media Types are:

Value	Description
all	Used for all media type devices
print	Used for printers
screen	Used for computer screens, tablets, smart-phones etc.
speech	Used for screenreaders that "reads" the page out loud

Media queries

```
@media not|only mediatype and (expressions) {  
    CSS-Code;  
}
```

- ▶ If the browser window is 600px or smaller, the background color will be lightblue:

```
@media only screen and (max-width: 600px) {  
    body {  
        background-color: lightblue;  
    }  
}
```

Media queries

- ▶ Use mediaqueries to set the background-color to lavender if the viewport is 800 pixels wide or wider, to lightgreen if the viewport is between 400 and 799 pixels wide. If the viewport is smaller than 400 pixels, the background-color is lightblue

```
body {  
    background-color: lightblue;  
}  
  
@media screen and (min-width: 400px) {  
    body {  
        background-color: lightgreen;  
    }  
}  
  
@media screen and (min-width: 800px) {  
    body {  
        background-color: lavender;  
    }  
}
```

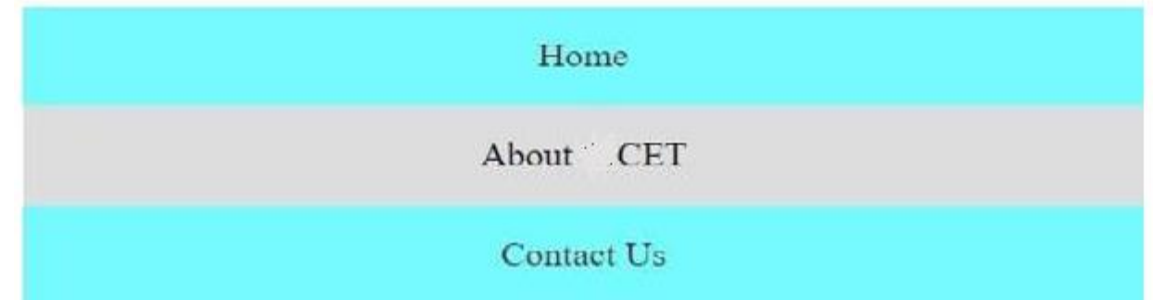
Media queries

- ▶ Media queries allow you to format your content to specific output devices. Media queries include a media type and expressions.
- ▶ Some of the common media features include:
 - ▶ • width—the width of the part of the screen on which the document is rendered, including any scrollbars
 - ▶ • height—the height of the part of the screen on which the document is rendered, including any scrollbars
 - ▶ • device-width—the width of the screen of the output device
 - ▶ • device-height—the height of the screen of the output device
 - ▶ • orientation—if the height is greater than the width, orientation is portrait, and if the width is greater than the height, orientation is landscape
 - ▶ • aspect-ratio—the ratio of width to height
 - ▶ • device-aspect-ratio—the ratio of device-width to device-height

Media queries

```
<!DOCTYPE html><html lang="en">
<head><meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
* {
  box-sizing: border-box;
}
/* style the top navigation bar */
.topnav {
  overflow: hidden;
  background-color: #00FFFF;
}
/* style the topnav links */
.topnav a {
  float: left;
  display: block;
  color: #671212;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}
```

Resize the browser window to see the effect: When the screen is less than 600px, the navigation menu will be displayed vertically instead of horizontally.



```

/* Change color on hover */
.topnav a:hover {
  background-color: #ddd; color: black;
}
/* On screens that are 600px wide or less, make the menu links
stack on top of each other instead of next to each other */
@media screen and (max-width: 600px) {
  .topnav a { float: none; width: 100%; }
}
</style></head><body>

<p>Resize the browser window to see the effect: When the screen is less
than 600px, the navigation menu will be displayed vertically instead of horizontally.</p>
<div class="topnav">
  <a href="#">Home</a>
  <a href="#">About CET</a>
  <a href="#">Contact Us</a>
</div>
</body>

```

Resize the browser window to see the effect: When the screen is less than 600px, the navigation menu will be displayed vertically instead of horizontally.

Home About CET Contact Us

Thank You