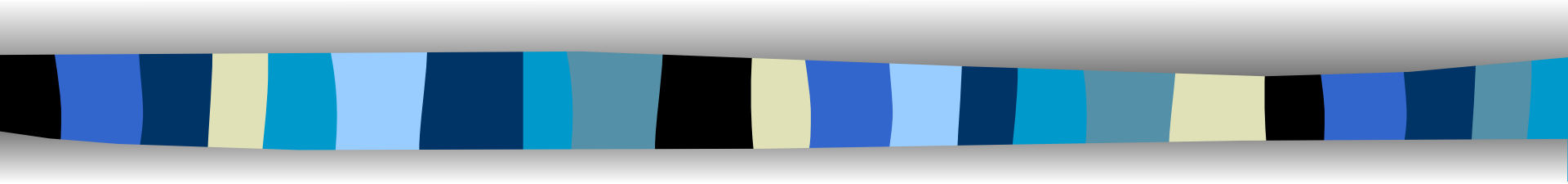
Cascading Style Sheets (CSS)



CSCI 3000
Web Programming

Dr. Luis A. Cueva-Parra

Cascading Style Sheets (CSS)

- CSS describes how HTML elements are to be displayed on screen, paper, or in other media.
- ***CSS Syntax:*** A CSS rule-set consists of a selector and a declaration block.

  selector declaration declaration

  ```
  h1 {color:blue; font-size:12px;}
  ```

  property value property value
- The selector points to the HTML element you want to style.
- The declaration block contains one or more declarations separated by semicolons.

CSS Selectors

- The *element* selector selects elements based on the element name. Here all <p> elements are selected.

```
p {text-align:center; color:red;}
```

- The `id` selector uses the `id` attribute of an HTML element to select a specific unique element.

```
#para1 {text-align:center; color:red;}
<p id="para1">Hello World!</p>
```

- The *`class`* selector selects elements with a specific class attribute.

```
.center {text-align:center; color:red;}
<p class="center">Hi there!</p>
```

CSS Selectors

- We can specify that only specific HTML elements should be affected by a class.

```
p.center {text-align:center; color:red;}
<p class="center">This paragraph</p>
```

- HTML elements can also refer to more than one class. Here `p` is styled using classes `center` and `large`

```
<p class=”center large”>Hi There!</p>
```

- We group selectors if we have elements with the same style definitions.

```
h1,h2,p {text-align:center; color:red;}
```

CSS Comments

- Comments are used to explain the code, and may help when you maintain your code.
- Comments are ignored by browsers.
- A CSS comment starts with /* and ends with */
- Comments can also span multiple lines.

```
p { color: red;
 /* This is a single-line comment */
    text-align: center; }
```

CSS Multiple Style Sheets

- If some properties have been defined for the same selector (element) in different style sheets, the value from the last read style sheet will be used.

```
<head>
<link rel="stylesheet"
type="text/css" href="mystyle.css">
<style>
  h1 { color: orange;}
</style>
</head>
```

CSS Cascading Order

- What style will be used when there is more than one style specified for an element?
- Generally speaking we can say that all the styles will "cascade" into a new "virtual" style sheet by the following order (first has highest priority)

  1) Inline style (inside an HTML element)

  2) External and internal style sheets (in the head section)

  3) Browser default

CSS Colors

- Colors are specified by:
  - a valid color name - like "`red`"
  - an RGB value - like "`rgb(255, 0, 0)`"
  - a HEX value - like "`#ff0000`"
- HTML and CSS supports 140 standard color names.

```
CornflowerBlue  #6495ED
HoneyDew        #F0FFF0
```

CSS Background Properties

- `background-color` specifies the background color of an element.

```
body {background-color: Peru;}
h1 {background-color: green;}
div {background-color: lightblue;}
p {background-color: yellow;}
```

CSS Background Properties

- `background-image` specifies an image to use as the background of an element.
- By default, the image is repeated (horizontally and vertically) so it covers the entire element.

```
body {background-image: url("paper.gif");}
```

- To repeat *only* horizontally use

```
body {background-image: url(“myF.gif”);
      background-repeat: repeat-x;}
```

CSS Background Properties

- To show the background image *only once*

```
body {background-image: url(“myF.gif”);
      background-repeat: no-repeat;}
```

- To *position* the background image and *fix* it so it *doesn't scroll* with the rest of the page.

```
body {background-image: url(“myF.gif”);
      background-repeat: no-repeat;
      background-position: right top;
      background-attachment: fixed;}
```

CSS Background Properties

- ***Shorthand property***: Specifies all the background properties in one single property.

```
body { background: #ffffff url(“myF.gif”)
no-repeat right top; }
```

- The order of the property values is:

1)background-color
2)background-image
3)background-repeat
4)background-attachment
5)background-position

CSS Border Properties

- Border properties specify the style, width, and color of an element's border.
- `border-style` specifies what kind of border to display.

```
p.dotted {border-style: dotted;}
```

- The allowed `border-style` values are (1/2):

`dotted`	Defines a dotted border
`dashed`	Defines a dashed border
`solid`	Defines a solid border
`double`	Defines a double border

CSS Border Properties

- The allowed `border-style` values are (2/2):

`groove`	Defines a 3D grooved border. The effect depends on the border-color value
`ridge`	Defines a 3D ridged border. The effect depends on the border-color value
`inset`	Defines a 3D inset border. The effect depends on the border-color value
`outset`	Defines a 3D outset border. The effect depends on the border-color value
`none`	Defines no border
`hidden`	Defines a hidden border

CSS Border Properties

- The `border-style` property can have from one to four values (for the *top border*, *right border*, *bottom border*, and the *left border*).

```
p.mix {border-style: dotted dashed
solid double;}
```

CSS Border Properties

- The `border-width` specifies the width of the four borders. Sets the width in `px`, `pt`, `cm`, or `em`, or by using one of the three pre-defined values: `thin`, `medium`, or `thick`.

```
p.one { border-style: solid; border-width: 5px;}
p.two { border-style: solid; border-width: medium; }
p.three { border-style: solid; border-width: 2px 10px 4px 20px; }
```

CSS Border Properties

- The `border-color` specifies the color of the four borders.
- The `border-color` property can have from one to four values (for the top border, right border, bottom border, and the left border).
- If border-color is not set, it inherits the color of the element.

```
p.colored { border-style: solid;
border-color: red green blue
yellow;}
```

CSS Border Properties

- We can specify border properties individually for each side.

```
p { border-top-style: dotted;
    border-right-style: solid;
    border-bottom-style: double;
    border-left-style: groove; }
```

CSS Border Properties

- The `border-style, border-width,` and `border-color` have four values for specifying a style, width or color for each side of a border.

```
border-style: solid double dashed dotted;
```

Top Right Bottom Left

CSS Border Properties

```
border-style: solid double dashed;
```

T R B

```
border-style: solid double;
```

T+B R+L

```
border-style: solid;
```

T+R+B+L

CSS Border Properties

- ***Shorthand property***: Specifies all the border properties in one single property.

```
p { border: 6px dotted blue }
```

- The order of the property values is:
  1)border-width
  2)border-style (required)
  3)Border-color

```
p { border-left: 5px solid red }
```

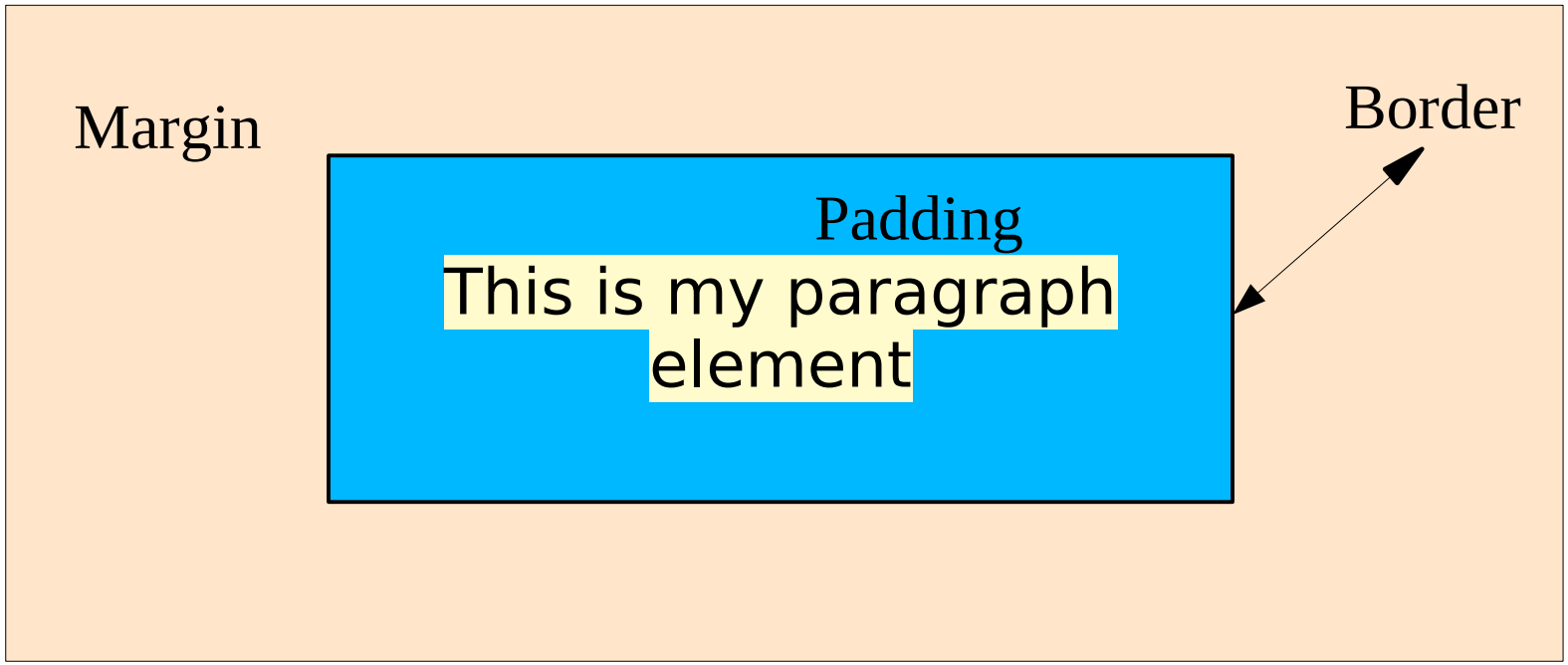
CSS Rounded Borders

- The `border-radius` specifies the radius of a rounded border.

```
p {
    border: 2px solid green;
    border-radius: 5px;
}
```

CSS Margins

- The CSS `margin` property defines the space around elements but outside of the bordes.



CSS Margins

- We can specify the margins individually for each side.

```
p { margin-top: 100px;
    margin-right: 150px;
    margin-bottom: 80px;
    margin-left: 100px; }
```

CSS Margins

- ***Shorthand property***: Specifies all the margin properties in one single property.

```
p { margin: 25px 50px 75px 100px; }
```

- The order of the property values is:

1)margin-top
2)margin-right
3)margin-bottom
4)margin-left

CSS Margins

```
margin: 100px 75px 50px;
```

T R B

```
margin: 100px 75px;
```

T+B R+L

```
margin: 100px;
```

T+R+B+L

CSS Margins – Special Values

- The value `auto` horizontally centers the element within its container.

```
p {margin: auto;}
```

- The value `inherit` inherits the margin from the parent element.
- When elements share their bottom and top margins, the rendering will collapse these two margin in one shorter than adding the two margins.

CSS Margins – Special Values

- Inheritance example (HTML content):

```
<h2>Use of the inherit value</h2>

<p>Let the left margin be inherited from the parent element:</p>


<div>

<p class="ex1">This paragraph has an inherited left margin (from the div element).</p>

</div>
```

CSS Margins – Special Values

- Inheritance example (CSS declaration)

```
div {
    border: 1px solid red;
    margin-left: 100px;
}

p.ex1 {
    margin-left: inherit;
}
```

CSS Text

- The property `color` sets the color of the text.

```
h1 {color: #4682B4;}
```

- The property `text-align` sets the horizontal alignment of a text. The values are: `center, left, right, justify.`

```
h1 {text-align: center;}
div {text-align: justify;}
```

CSS Text

- The property `text-decoration` sets or remove decorations from text. The values are: `none, overline, line-through, underline.`

```
a {text-decoration: none;}
h1 {text-decoration: overline;}
h3 {text-decoration: underline;}
```

- The property `text-transform` sets uppercase or lowercase letter in a text. The values are: `uppercase, lowercase, capitalize.`

```
p.uppercase {text-transform: uppercase;}
```

CSS Text

- The property `text-indent` sets the indentation of the first line of a text. Its values are in units of distance: `pt, px, em,` etc.

```
p {text-indent: 35px;}
```

- The property `letter-spacing` sets the spacing between the characters in a text.

```
h1 {letter-spacing: 4px;}
h2 {letter-spacing: -4px;}
```

- The property `line-height` sets the space between lines.

```
p.big {line-height: 1.8;}
```

CSS Text

- The property `direction` is used to change the text direction.

```
p {direction: rtl;}
```

- The property `word-spacing` sets the space between the words in a text.

```
h1 {word-spacing: 12px;}
h2 {word-spacing: -5px;}
```

- The property `text-shadow` adds shadow to text.

```
h1 {text-shadow: 3px 2px blue;}
```

CSS Fonts

- The property `font-family` sets the font family of a text. Some font families are: `arial, georgia, times new roman, verdana, courier new,` etc.

```
p {font-family: Times, serif, “Times New Roman”;}
```

- The property `font-style` sets mostly italic text. It has 3 values: `normal, italic, oblique.`

```
p.italic {font-style: italic;}
p.oblique {font-style: oblique;}
```

CSS Fonts

- The property `font-size` sets the size of a text. The `font-size` value can be absolute or relative. Default font size is 16px (16px=1em).

```
body {font-size: 100%;}
p {font-size: 0.875em;}
h1 {font-size: 2.5em;}
```

- The unit `vw` (viewport width) can be used to set text size. It sets the size relative to the browser window. `1vw` = 1% of viewport width.

```
<h1 style="font-size:10vw">Hello!</h1>
<p style="font-size:4vw">My text </p>
```

CSS Icons

- The simplest way of adding icons to an HTML page is using an CSS icon library.
- Add the class of the icon to any inline element such as `<i>` or `<span>.`
- **Font Awesome Icons:**

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
```

```
<i class="fa fa-car" style="font-size:48px;"></i>
<i class="fa fa-car" style="font-size:60px;color:red;"></i>
```

CSS Icons

- **Bootstrap Icons:**

```
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com
/bootstrap/3.3.7/css/bootstrap.min.css"
>
```

```
<i class="glyphicon glyphicon-envelope"></i>
<i class="glyphicon glyphicon-thumbs-up"></i>
```

CSS Icons

- **Google Icons:**

```
<link rel="stylesheet"
href="https://fonts.googleapis.com/icon
?family=Material+Icons">
```

```
<i class="material-icons" style="font-
size:36px;">traffic</i>
<i class="material-icons" style="font-
size:48px;color:red;">traffic</i>
```

CSS Links

- Links can be styled with any CSS property (e.g. `color, font-family, background,` etc.).

```
a {color: purple;}
```

- Links also can be styled based on their **state**. There are 4 link states:

  a:link - a normal, unvisited link

  a:visited - a link the user has visited

  a:hover - a link when the user mouses over it

  a:active - a link the moment it is clicked

```
a:hover {color: orange;}
a:visited {color: green;}
```

CSS Links

- Order rules when setting several link states:

  a:hover MUST come after a:link and a:visited

  a:active MUST come after a:hover

- The property `text-decoration` removes underlines from links.

```
a:link {text-decoration: none;}
a:visited {text-decoration: none;}
a:hover {text-decoration: underline;}
```

- The property `background-color` rsets the background color for links.

```
a:link {background-color: cyan;}
```

CSS Link Buttons (1/2)

```
a:link, a:visited {
    background-color: #f44336;
    color: white;
    padding: 14px 25px;
    text-align: center;
    text-decoration: none;
    display: inline-block;
}
a:hover, a:active {
    background-color: blue;
}
```

CSS Link Buttons (2/2)

```
<a href="default.asp"
target="_blank">This is a link</a>
```

CSS Lists

- CSS list properties will set item markers, an image as list item marker and add background colors to lists and list items.
- The property `list-style-type` sets the type of list item marker.

```
ul.a {list-style-type: circle;}
ul.b {list-style-type: square;}
ol.c {list-style-type: upper-roman;}
ol.d {list-style-type: lower-alpha;}
```

CSS Lists

- The property `list-style-image` sets an image as the list item marker.

```
ul {
   list-style-image:url('myImg.gif');}
```

- The property `list-style-position` sets the position of the list item marker.

```
ul.a {list-style-position: outside;}
ul.b {list-style-position: inside;}
```

- Bullets/markers by default are outside of the list item.

CSS Lists

- Lists have margin and padding.
- The remove bullets/markers use the property `list-style-type:none` sand to remove margin and padding set them to 0.

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

- The shorthand property `list-style` sets all list properties in one declaration.

```
ul {list-style: square inside
url("myFig.gif";}
```

CSS Lists

- The order of the properties in the shorthand property is:

  List-style-type

  List-style-position

  List-style-image

- Lists can be styled with colors: .

```
ul { background: #3399ff;
     padding: 20px;}
ul li { background: #cce5ff;
        padding: 5px;}
```

CSS Tables

- The property `border` sets the border thickness, type and color of table.

```
table, th, td {
                border: 1px solid green;}
```

- The property `border-collapse` sets the table border to collapse into one single border.

```
table {border-collapse: collapse;}
table, th, td {
                border: 2px solid #787878;}
```

CSS Tables

- The properties `width` and `height` set the width, and height of a table.

```
table { width: 100%;}
th { height: 45px;}
```

- The property `text-align` sets the horizontal alignment (left, right or center) of the cells in a table.

```
th { text-align: right;}
```

- The property `vertical-align` sets the vertical alignment (top, bottom or middle) of the cells in a table.

```
td { vertical-align: bottom;}
```

CSS Tables

- The padding of the cells (th, td) can be set with the property padding.

```
table { width: 100%;}
th, td { padding: 35px;}
```

- The property border-bottom sets horizontal dividers to the cells (th, td).

```
th, td { border-bottom: 2px solid
                        #ddd;}
```

- Use the :hover selector on tr to highlight table rows on mouse over.

```
tr:hover { background-color:#f5f5f5;}
```

CSS Tables

- To create striped tables, use the `nth-child()` selector and add a `background-color` pto all even (or odd) table rows.

```
tr:nth-child(even) {
            background-color: #745313;}
```

- To make a responsive table, place the table element inside a container element (such as) with `overflow-x:auto`

```
<div style="overflow-x:auto;">
<table>
 . . .
</table>
</div>
```

CSS Layout – Display Property

- The property `display` sets if/how an element will be displayed. The values `block` and `inline` are the default values for most elements.
- Block-level elements start on a new line and occupies the full width available.
- Block-level elements include:

  `<div>`, `<h1>` - `<h6>`, `<p>`, `<form>`, `<header>`, `<footer>` and `<section>`
- Inline elements don't start on a new line and only takes the necessary width.
- Inline elements include: `<span>`, `<a>` and `<img>`

CSS Layout – Display Property

- The value `none` for the property `display` is used to dynamically remove elements from a website.
- `display:none` is the default for the `<script>` element (JavaScript).
- We can override the default display value, from `block` to `inline` or vice-versa.

```
a { display: block;}
```

- `display:none` hides an element as it is not there.
- `visibility:hidden` hides an element but the space is kept. The layout is maintained.

CSS Layout – Float and Clear

- The CSS `float` property sets how the element floats on the layout. Its possible values are:

  left: floats to the left of its container

  right: floats to the right of its container

  none: it does not float (default)

  inherit: inherits the float value of its parent.

- The CSS `clear` property specifies what elements can float beside the cleared element and on which side.

CSS Layout-Float and Clear

```
.div1 {
    float: left;
    width: 100px;
    height: 50px;
    margin: 10px;
    border: 3px solid #73AD21;
}

.div2 {
    border: 1px solid red;
}
```

CSS Layout-Float and Clear

```
.div3 {
    float: left;
    width: 100px;
    height: 50px;
    margin: 10px;
    border: 3px solid #73AD21;
}
.div4 {
    border: 1px solid red;
    clear: left;
}
```

CSS Layout-Float and Clear

```
<h2>Without clear</h2>

<div class="div1">div1</div>

<div class="div2">div2 - Notice that div2 is
after div1 in the HTML code. However, since
div1 floats to the left, the text in div2
flows around div1.</div>

<br><br>

<h2>With clear</h2>

<div class="div3">div3</div>

<div class="div4">div4 - Here, clear: left;
moves div4 down below the floating div3. The
value "left" clears elements floated to the
left. You can also clear "right" and
"both".</div>
```