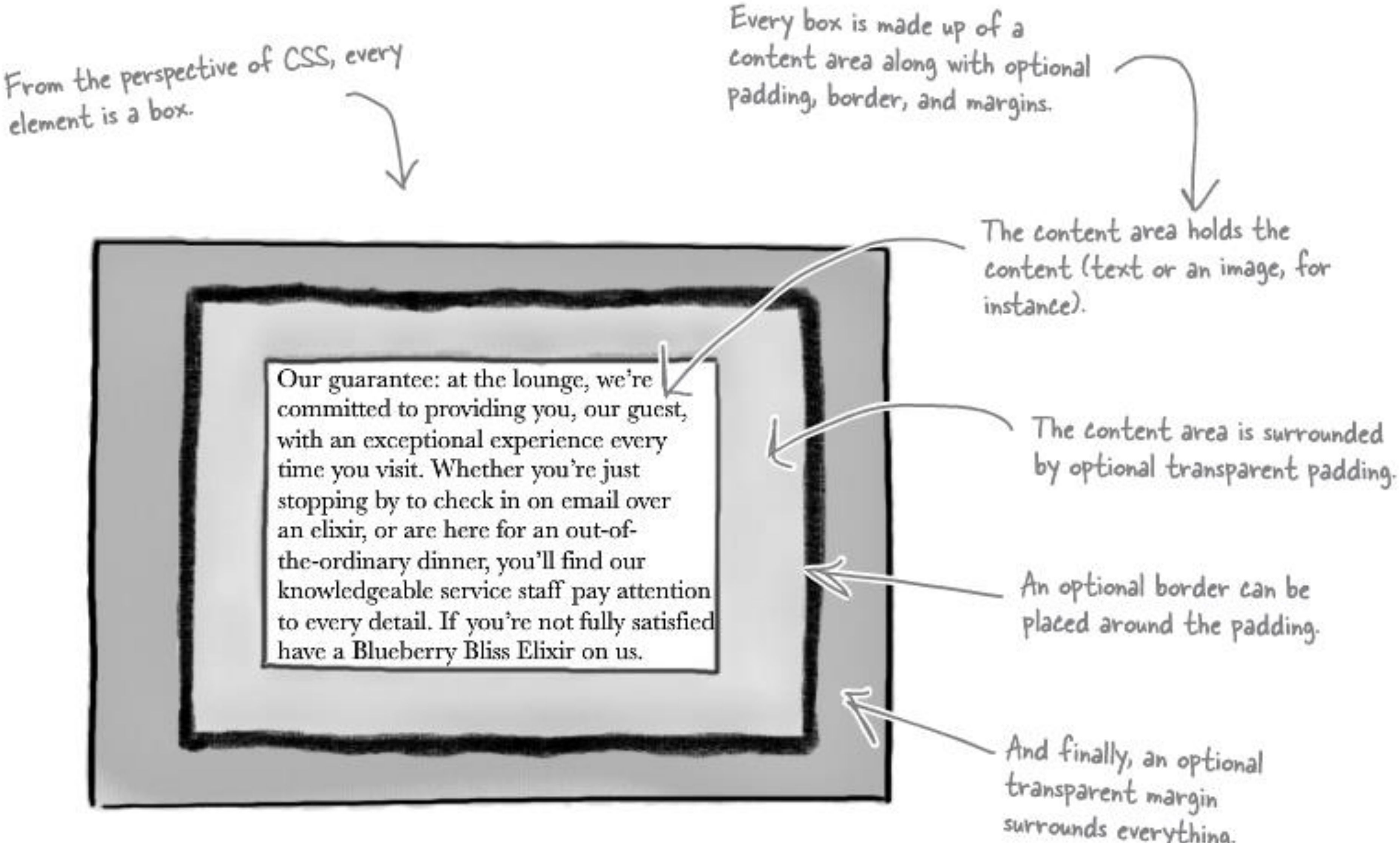
Lecture 9: Box Model, Styling Elements, and Media Queries

Dr. Jason Schanker

Line Spacing

- The `line-height` property is used to specify vertical spacing between lines (known in publishing industry as leading, pronounced ledding); 1.6em means 1.6 times the height of the text.

The Box Model



From the perspective of CSS, every element is a box.

Every box is made up of a content area along with optional padding, border, and margins.

Our guarantee: at the lounge, we're committed to providing you, our guest, with an exceptional experience every time you visit. Whether you're just stopping by to check in on email over an elixir, or are here for an out-of-the-ordinary dinner, you'll find our knowledgeable service staff pay attention to every detail. If you're not fully satisfied have a Blueberry Bliss Elixir on us.

The content area holds the content (text or an image, for instance).

The content area is surrounded by optional transparent padding.

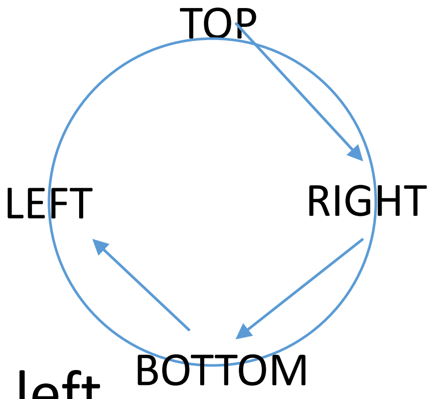
An optional border can be placed around the padding.

And finally, an optional transparent margin surrounds everything.

Box Model Dissected

- Content area: Tight box surrounding content (e.g., text, image, etc.)
- Padding: Blank spacing surrounding content area (inside border) that's "internal" to the element in the sense that if the element has a background, it will show through the padding
- Border: Surrounds the padding (and content area); divides padding and margin; encloses content and background of element
- Margin: Blank space surrounding border; "External" to the element in the sense that it may merge with margins of other elements and element's background will not extend behind margin
- "Think of padding as part of the element, while the margin surrounds your element and buffers it from the things around it."

Padding/Margin Properties



TOP

RIGHT

BOTTOM

LEFT

- If four numbers are used, then they specify the top, right, bottom, and left paddings/margins in that order (like a clock starting from 12).
  - Example: `margin: 5px 10px 15px 20px;` gives a top margin of 5 pixels, a right one of 10, a bottom one of 15 and a left one of 20
- If two numbers are used, then they specify the vertical (top/bottom) spacing and the horizontal (right/left) spacing.
  - Example: `padding: 5px 10px;` specifies 5 pixels of top/bottom padding and 10 pixels of right/left padding
- If one number is used, then all spacing (top, right, bottom, and left) is the same:
  - Example: `margin: 20px;` specifies 20 pixels of margin space (top, right, bottom, and left).
- The default padding and margins are 0 pixels and these properties are <u>NOT</u> inherited. You can specify top, right, bottom and left padding and margins individually by adding
  - Example: `padding-top: 5px;` sets the top`-top, -right, -bottom, and -left.` padding to 5 pixels; if unspecified, all other padding will still be 0.

Background Properties

- The `background-color` is used to specify the background color of the content area, padding, and if it's dashed, dotted, etc., between gaps in the border as well.
    - Example: `background-color: #ff0000;`
- The `background-image` property can be used to specify the background when you want to use an image. Use a value of `url (`*path to image*`) ;` Note: The path does *not* need to be in quotes.
    - Example: `background-image:url(images/background.gif);`
- The `background-position` property positions a background image. You can use combinations of `top/bottom`, and `right/left`.
    - Example: `background-position: top left;`
- The `background-repeat` property can be used with `repeat` (which tiles the image and is the default), `repeat-x` and `repeat-y` (which tiles only horizontally and vertically, respectively), `no-repeat` (only once), or inherit (inherits the parent's `background-repeat` value).
    - Example: `background-repeat: no-repeat;`

Border properties

- The `border-style` property can have values of `solid` (default), `double, dashed, dotted, groove, inset, outset, or ridge`.
- The `border-width` property can have values of `thin, medium,` or `thick`, or be specified with a number of pixels (e.g., `2px`).
- The `border-color` property can be specified with the same values as text and background colors.
- You can also specify top, right, left, and bottom border properties for border-color, border-width and border-style by inserting `-top`, `-right, -bottom, and -left` after border:
  - ❑Example: `border-right-color: rgb(255, 0, 0);`
- The `border-radius` property can be used to create borders with rounded corners. The values can be expressed in pixels (`px`) or relative to text size with the `em` unit. You can also insert `-top-left, -top-right, -bottom-right, or -bottom-left` to specify corners individually.
  - ❑Example: `border-top-right-radius: 2em;`

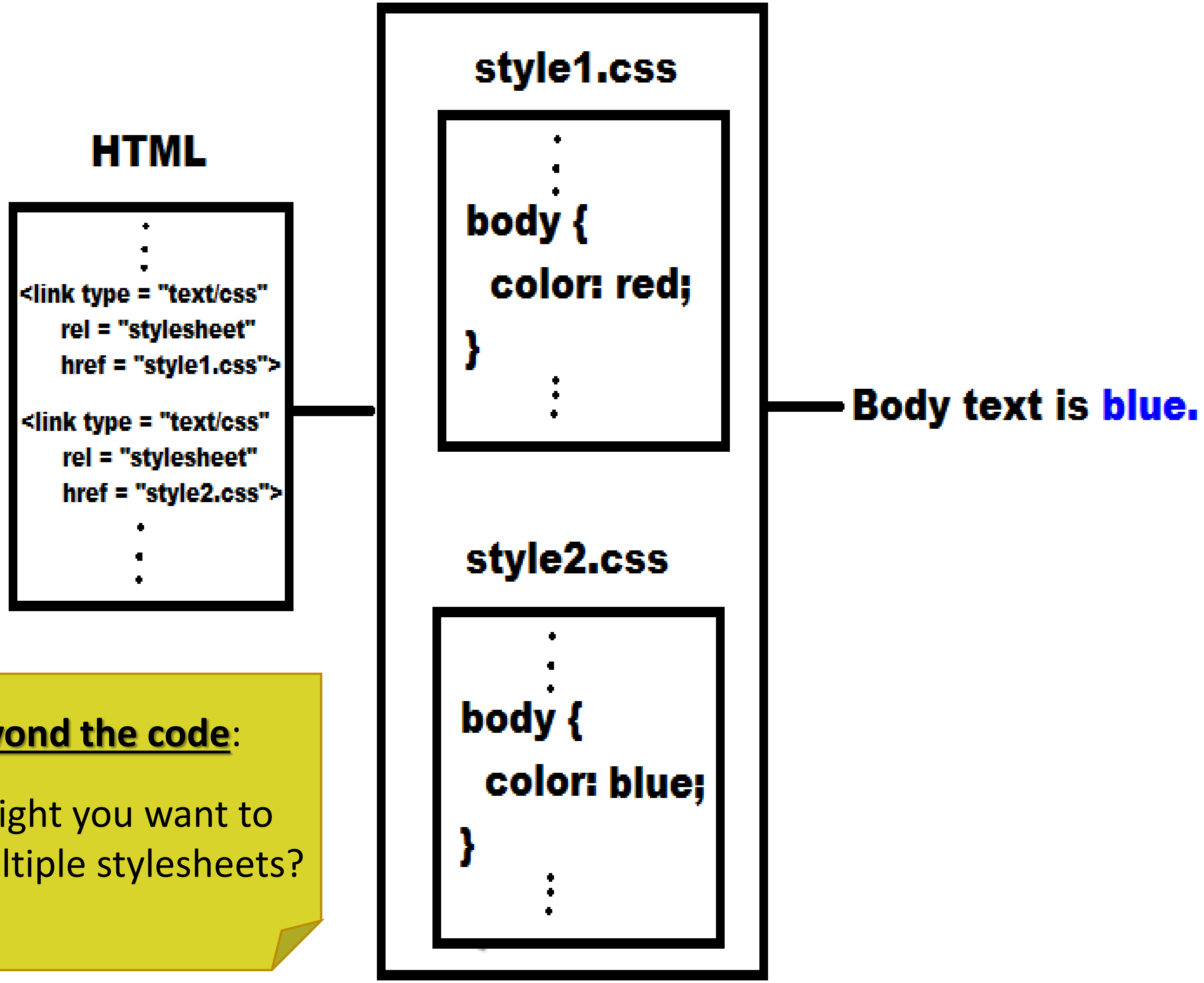
Using ids

- You can use the `id` attribute with any element just as you can with the `class` attribute.
- “You use a class when you might want to use a style with more than one element. And if what you need to style is unique and there’s only one on your page, then use an id. The id attribute is strictly for naming unique elements.”
- You can use both the class an id attribute with the same element.
  - ❑ Example: You can have a unique logo on the page that’s one of the multiple drawings (as opposed to photos) that you display on your page. In that case, you may have a tag like this:
    `<img id = "logo" class = "drawing" src = "images/logo.gif">`
- To select the element with a specific id in a CSS rule for styling, use a # before the id (e.g., `img#logo` or simply `#logo`).

```
#logo {
        border: 1px dashed;
}
```

Multiple stylesheets

- You can link to multiple stylesheets, in which case the browser treats the rules from the stylesheets as if they came from a single one, with the later linked sheets' rules coming *after* the earlier linked sheets' rules. This is important for determining "winners" in styling conflicts.



HTML

```
        ⋮
<link type = "text/css"
     rel = "stylesheet"
     href = "style1.css">
<link type = "text/css"
     rel = "stylesheet"
     href = "style2.css">
        ⋮
```

style1.css

```
     ⋮
body {
  color: red;
}
     ⋮
```

style2.css

```
     ⋮
body {
  color: blue;
}
     ⋮
```

Body text is blue.

<u>Beyond the code</u>:

Why might you want to use multiple stylesheets?

Media Queries

- You may want to use different stylesheets for different devices. For example, you may want a 3-column layout for a desktop user but a single-column layout to target a mobile device user.
- Using HTML and/or CSS media queries, you can do exactly that.
  - ❑ HTML: Use the `media` attribute of the `link` element:
    - Example:
      ```
      <link href="lounge-mobile.css" rel="stylesheet"
            media="screen and (max-device-width: 480px)">
      ```
    - Example:
      ```
      <link href="lounge-print.css" rel="stylesheet"
            media="print">
      ```
  - ❑ CSS: Use the `@media` directive:
    - Example:
      ```
      @media screen and (max-device-width: 480px) {
         CSS Rules for browser windows <= 480 pixels run on screens go here
       }
      ```
    - Example:
      ```
      @media print {
        CSS Rules for printing
       }
      ```
    - All rules not surrounded by media directives are applied to all devices.
  - ❑ HTML link element media attribute is better to use for large number of device-specific rules.

Book Exercises

I. Create a lounge.css file. Set the default font size of the page’s text to small and a font face of Verdana; if Verdana’s not available, use Helvetica; then if Helvetica’s not available, use Arial; otherwise have the browser use its default sans-serif font. Also, set the line spacing to be 1.6 times the font size. Then set the colors of the h1 and h2 elements to have an aquamarine color (hex code #007e7e) “to match the glass in the logo”. Set the primary headings to 150% of the body text and the secondary headings to 130% of the body text.

II. Give the guarantee paragraph a unique id of `guarantee` in the HTML. Italicize its text and give it a font face of Georgia; if Georgia’s not available, use "Times New Roman"; then if "Times New Roman" is not available, use Times; otherwise have the browser use its default serif font. Also, set the line spacing to be 1.9 times the font size.

III. Set the padding to 25 pixels and the margin to 30 pixels. Then use a thin white dashed border to give the paragraph a “ragged” look.

IV. Set the pargraph background color to aquamarine (hex code #007e7e), and its `background-image` property to `url(images/background.gif)`. Make sure the image does *not* repeat and that its in the top left corner.

Slightly Modified Book Exercises (cont.)

V. Increase the padding on the left only to 80 pixels so that the text does not appear on the star. Then increase the right margin to 250 pixels.

VI. Cut and paste the guarantee rule into two new CSS files: lounge-compact.css and lounge-medium.css. Make what you believe are appropriate changes to the guarantee rule for small browser windows in the lounge-compact.css file. Then add two link elements with media attribute *above* the existing link element in the HTML, in which one specifies use of the lounge-compact.css stylesheet for a browser window with width of at most 480 pixels (screen) and the other one which specifies use of the lounge-medium.css stylesheet for a browser window whose width is at least 481 pixels (screen). Finally, add one specifying lounge-print.css for print media. Test these all by resizing the browser window and viewing the page in Print Preview mode.

Notes

- This is primarily a summary of Chapter 9 of *Head First HTML and CSS*, 2nd Edition by Elisabeth Robson and Eric Freeman, 2012. It contains images, exercises, and code from the book.