Introducing CSS

• Recall that three primary technologies are used when implementing interactive web pages.
  - HTML: which dictates the structure and content of a web page
  - JavaScript: which implements animations and user interactivity and otherwise control the behavior of the elements
  - CSS: short for Cascading Style Sheets, controls layout, formatting, and presentation

• Any nontrivial web application will require a large amount of HTML, CSS, and JavaScript.
  - As web applications grow in complexity, it’s important to decouple the HTML, CSS, and JavaScript as much as possible
  - so that changes to an HTML document never break a CSS rule or compromise the execution of JavaScript event handlers.
  - Web applications that successfully separate content, presentation, and interactivity are easier to modify and maintain.

CSS Declarations

• Web designers control the presentation of a page using one or more CSS declarations, each of which is structured like this:
  • property-name: property-value;

  A property name is one of several CSS keywords (560 according to https://www.w3.org/Style/CSS/all-properties.en.html) controlling some presentation detail.
  The set of possible property values depends on the name.
  • background-color can take on any legal JavaScript or CSS color, e.g., green, rgb(35, 172, 209), or #E98725.
  • text-align governs the horizontal alignment of text and can be set to left, right, center, or justify.
  • display controls how an element is displayed and can be bound to inline, block, inline-block, flex, and none.

CSS Declarations: Photo Wall

• The easiest way to include a CSS declaration is to include a style attribute as part of a tag, as with:
  • The above HTML fragment produces the following:

    • In the last example, each img tag is decorated with its own style attribute.
      • All values share the same height property value, so the displayed images are scaled to the specified height and are vertically aligned.
      • Unless otherwise specified, image elements are inline-block, which means they flow left to right just as text does (that’s the inline part) and their width and height can be specified (that’s the block part).
      • Each value specifies its own margin property value. The border width is 1px everywhere, but each specifies its own border style.
    • The div tag is styled so all text—or rather, all inline elements like img—under its jurisdiction are horizontally centered.
    • You don’t need to memorize all 500+ property names and the spectrum of possible property values for each. You just need to be able to read CSS declarations and understand them.
CSS Rules

• To mitigate the code replication problem, we define styles that apply not to a single element but to an entire document. The most common practice for doing so is defining a set of CSS rules, each of which takes the following form:

```html
<html>
<head>
<title>Photo Wall, Take II</title>
<meta charset="UTF8"/>
</head>
<body>
<!-- the selector component is often a cascade of selectors that telescope to identify a specific category of HTML elements. -->
</body>
</html>
```

• To help mitigate both problems, CSS declarations are typically placed in separate files and referenced by the HTML file that depends on them. These files are called *stylesheet*. The first rule applies to elements of that type that satisfies the same two constraints, but replace "highlighted" with "lampset".

• HTML and CSS can’t be completely decoupled, since each impacts the other. We rely on the two mechanisms above to make a web application’s architecture as modular as possible.

• The selector portion of the rule can take on many forms, the most basic of which take on the following structure:

  - A hash followed by the DOM id of an element, which states the rule should be applied to the element with that id.
  - A period followed by a name, which defines a rule that HTML tags adopt by defining a class attribute whose value includes the name. So, `span class="answer incorrect final">3</span>` would cause the 31 of interest would be presented in *red*. angry *red*.
  - An HTML tag name, which specifies that all elements of that type respect the rule. The rule to the right would mandate all images be 72px square.

The Cascade of CSS

• The selector component is often a waterfall of selectors that telescope to identify a specific category of HTML elements.

  ```html
  span.highlighted
  #keyboard
  .correct
  ```

• The first rule applies to `span` elements that:
  - are below the element with id “keyboard” in the DOM tree
  - include "highlighted" in its list of class attribute values

• The second rule also applies to `span` satisfying the same two constraints, but replace "highlighted" with "lampset".

• The *cascade* is the algorithm browsers use to decide what rules apply to what elements, particularly in the face of conflicts.

CSS and JavaScript

• JavaScript can be used to modify the presentation of existing elements.

• The JavaScript standard recognizes that per-element style manipulation is so common that it exposes the `classList` attribute:

```javascript
window.onload = function() {
  var elem = document.getElementById("photoWall");
  elem.classList.add("portrait");
  elem.classList.add("frame");
  elem.classList.add("border");
  elem.style.border = "2px solid #000000";
};
```

• The `classList` attribute stores its class names in an array-like object. So, the element representing:

  `<span class="answer incorrect final">3</span>`

would include a `classList` attribute with three classes. The only three `classList` methods of interest are illustrated here:

- `elem.classList.add("portrait")` adds the class.
- `elem.classList.remove("portrait")` removes the class.
- `elem.classList.contains("portrait")` checks whether the class is present.

The End

• Here is an implementation of the photo wall application that makes proper use of CSS rules and `classList`:

```html
<html>
<head>
<title>Photo Wall, Take II</title>
<meta charset="UTF8"/>
</head>
<body>
<!-- the selector component is often a cascade of selectors that telescope to identify a specific category of HTML elements. -->
</body>
</html>
```

• To mitigate the code replication problem, we define styles that apply not to a single element but to an entire document. The most common practice for doing so is defining a set of CSS rules, each of which takes the following form:

```html
<html>
<head>
<title>Photo Wall, Take II</title>
<meta charset="UTF8"/>
</head>
<body>
<!-- the selector component is often a cascade of selectors that telescope to identify a specific category of HTML elements. -->
</body>
</html>
```

• To help mitigate both problems, CSS declarations are typically placed in separate files and referenced by the HTML file that depends on them. These files are called *stylesheet*. The first rule applies to elements of that type that satisfies the same two constraints, but replace "highlighted" with "lampset".

• HTML and CSS can’t be completely decoupled, since each impacts the other. We rely on the two mechanisms above to make a web application’s architecture as modular as possible.

• The selector portion of the rule can take on many forms, the most basic of which take on the following structure:

  - A hash followed by the DOM id of an element, which states the rule should be applied to the element with that id.
  - A period followed by a name, which defines a rule that HTML tags adopt by defining a class attribute whose value includes the name. So, `span class="answer incorrect final">3</span>` would cause the 31 of interest would be presented in *red*. angry *red*.
  - An HTML tag name, which specifies that all elements of that type respect the rule. The rule to the right would mandate all images be 72px square.

The Cascade of CSS

• The selector component is often a waterfall of selectors that telescope to identify a specific category of HTML elements.

  ```html
  span.highlighted
  #keyboard
  .correct
  ```

• The first rule applies to `span` elements that:
  - are below the element with id “keyboard” in the DOM tree
  - include "highlighted" in its list of class attribute values

• The second rule also applies to `span` satisfying the same two constraints, but replace "highlighted" with "lampset".

• The *cascade* is the algorithm browsers use to decide what rules apply to what elements, particularly in the face of conflicts.

CSS and JavaScript

• JavaScript can be used to modify the presentation of existing elements.

• The JavaScript standard recognizes that per-element style manipulation is so common that it exposes the `classList` attribute:

```javascript
window.onload = function() {
  var elem = document.getElementById("photoWall");
  elem.classList.add("portrait");
  elem.classList.add("frame");
  elem.classList.add("border");
  elem.style.border = "2px solid #000000";
};
```

• The `classList` attribute stores its class names in an array-like object. So, the element representing:

  `<span class="answer incorrect final">3</span>`

would include a `classList` attribute with three classes. The only three `classList` methods of interest are illustrated here:

- `elem.classList.add("portrait")` adds the class.
- `elem.classList.remove("portrait")` removes the class.
- `elem.classList.contains("portrait")` checks whether the class is present.

The End