Lab 2

Members:

- Sam Avis
- Philip Chang
- Kristofer Kwan
- Yanshen Lin
- Yuhao Wang

What we did:

For this lab we created an html page and an xml page to display 10 different song titles, including information for:

- Song Name
- Album Name
- Album Image
- Album Site
- Song Date of Release
- Song Genre
- Band Site

Part 1:

For the html webpage, we had 10 different `section` tags to differentiate between songs. This overall kept the structure of the site clean. Visually, each section has a heading (an h1 tag), indicating the song number, followed by the image. under the image, the site lists out the metadata of the song, (as listed in the section above). This metadata was stored in an unordered list, as there's no explicit hierarchy regarding the song information that would require a nested approach. To validate our site, we utilized the w3 validation checker.

Part 2:

For the xml webpage, we had 10 different `song` tags which were nested within the `songs` element. Each individual `song` element contained an `album_art song name`, `artist`, `artist_site`, `album`, `album_info`, `release_date`, and `genre`. For `artist_site` and `album_info`, we additionally added a description section (to describe the url) and a `url` element that stored the URL relevant to the `album_info`, `artist_site`, and `album_art` tags. To validate our site, we utilized the an online xml validation checker.

Part 3:

For the CSS styling of our HTML page, we added a class to each attribues of a given song (which in turn has the class `song-data`). These attributes included:

- `song-name`
- `artist-name`
- `artist-site`

- `album`
- `album-info`
- `release-date`
- `genre` -- The `genre` class additionally has a `genre-data` section additionally listing every genre associated with a given song.

Required Changes

In the CSS style sheet (linked in the header of the html file), `lingoland-lab2-part3.css`, Theres a section, `.song-name`, which specifies the song name to be in small-caps. If the song is a favorite song, the song name list element will additionally have the `favorite` class, where the text will be highlighted in green. The `.artist-name` section changes the font-weight of the text to bold and increases the font size by 20% from the default, wheras the `.release-date` and `.genre` classes all show a proportionately smaller font-size (`fontsize: smaller;`). The CSS file explicitely floats the image to the right of the song content, as well as specifies each standardizes each album cover art to have a height and width of `200px`. For the genre list, we added a dash before each list item via the tag `.genre-data > li:before`.

Additional Changes:

- Added in nested column logic to html page, which contains a grandfather class (determines the table size), the parent class (determines specific row size and height), and the child1 and child2 classes (where child1 is used for song attributes, and child2 is used to frame the album cover images)
- Added margin left and right of 5% to give some space between the edge of the screen and the "table"
- set the "table headers" to bold of 600 and the color of the song headers to white
- Fixed the height of each row to be 250px via `song-data` class (unique to each row)
- Added padding to the right and top of album cover images
- Added a bottom border per each list instance to further separate items (using sections tag)
- Added alternating backgrounds per each item in the list

Part 4

For the CSS styling of the XML page, we added class attributes for the following xml elements and attributes:

- `song`
- `name`
- `artist`
- `album`

Required Changes:

For our song attribute, we forced all descendents under the song attribute to be `display block`, in order to allow for items to be listed. In addition, we made the `name` attribute's font size proportionately smaller than default, the `artist` font color to be blue, and the album font to be italicized.