Page setup Lab

Let's make the main page

The main page of your site is often called a home page or a landing page. Let's create it.

1. Create a new file called landing.html.
2. Give it a doctype declaration, a html tag, a head, and a body.

The page header

3. Inside the <body> add a <div> with an id of "header". This will become the header for all the pages on our site.
4. Inside the new div, add an image -- our company logo. Since this image is merely eye-candy give it an empty alt attribute. You'll find it in the assets folder.
5. Add a paragraph with the name of the company.
6. Add two links. The first should contain an image of a shopping cart and it links to a page called checkout.html. The second should say "Log in" and it should link to "register.html".

The page main section

7. After the header add a <div> with an id of "main".

Nothing needs to go in it right now. We'll add some things in a few minutes.

The page footer

8. Now add a <div> with an id of "footer" at the bottom of the page.
9. In it, add an unordered list. The list should have list elements (<li>s) with links to our website, Facebook, Twitter, and Instagram. Each of those should have images, web.png, facebook.png, twitter.png, and instagram.png respectively. These can be found in the assets folder.
10. Finally add a copyright notice using a real copyright symbol.
11. Run and test your page by opening landing.html in the browser of your choice. It should be a mess at this point but you should see all of your text and your images.

Creating the details page

We've just started on our landing page but the stuff we've put in it so far is great … so great in fact that we'd really like to duplicate it on the other pages. So let's create one more page and use landing.html as a template.

12. Copy this page and paste it as details.html.

13. Change the title to read "Film details – Dinner and a movie".

Adding film info to landing

Now let's add some content to the main section of the landing page.

14. Add an <h1> that says "Show times for Monday"
15. Add a <div> with an id of "day_chooser". Inside that <div> add an unordered list with "Tuesday", "Wednesday", "Thursday", "Friday", and "Saturday".
16. Add a <div> with an id of "film_briefs".
17. Inside that <div>, add another <div> with an class of "film_brief". Note that the outer one is plural and the inner one is singular. "film_brief**s**" vs "film_brief".
18. Inside the film_brief <div>, add three <div>s with an image, some paragraphs, and some showing_times. In fact, just make it look like this:

```html
<div id="film_briefs">
  <div class="film_brief">
    <div class="poster">
      <img src="assets/posters/1.jpg" alt="Fake Film 1" />
    </div>
    <div class="info">
      <h2>Fake film 1 Title</h2>
      <p class="tagline">Fake film 1 tagline</p>
      <p class="overview">Fake film 1 overview</p>
      <p class="ratings">Fake film 1 ratings</p>
    </div>
    <div class="showing_times">
      <ul>
        <li>10:30</li>
        <li>12:30</li>
        <li>2:30</li>
        <li>4:30</li>
        <li>6:30</li>
        <li>8:30</li>
      </ul>
    </div>
  </div>
</div>
```

19. Now copy the film_brief <div> and paste it 5 more times. In the end you should have one (1) <div id="film_briefs"> and six (6) <div class="film_brief">

20. If you look in the assets folder that you copied, there's a file called films.json. It's like a database file with details about all the films that are currently showing in our theaters. Take a look at that file.
21. Go through your six filmBriefs divs and replace their fake placeholder data with better data from the films.json file.

22. Run and test. Again, don't worry about image sizes and colors and fonts and so forth. We'll make things look better soon. For now, let's move on to the details page.

Adding the details to … umm … details

The details.html will hold the details for <u>one</u> (1) movie at a time. We're going to hardcode those details here[1].

23. Edit details.html
24. In the *main* section, add two divs with ids of "summary" and "detail".
25. In the "summary" div, add two more divs with ids of "text" and "poster".
26. Pick a movie. Read through the films.json file and focus on one film. Look at its details for these next steps.
27. In the "poster" div, add an <img> with the poster.
28. In the "text" div, add an <h1> with the movie title followed by a <p> with the tagline and another <p> with the film's run time. (Reminder: all of this raw data is in the films.json file).
29. Now go to the "detail" <div> and add <p>s with classes of "homepage", "overview", "release_date", and "viewer_ratings".
30. In the "viewer_ratings" <p>, do this:

```
<p class="viewer_ratings">
  Viewer ratings:
  <span class="vote_average">6.2</span>
  /10
  <span class="vote_count">(52 votes)</span>
</p>
```

Note the <span>s. These will group certain texts, allowing us to style them later.
31. The other <p>s' data should be obvious. Fill them with data from films.json.
32. Run and test one last time.

That should be enough for now. In the next chapters, we'll make these pages look really great, starting with some styling in the next chapter.

---

[1] Of course in the real world the film information would be retrieved via an Ajax call to a server. Since we don't have that yet, we'll hardcode the values.