🔔   Menu

◀ **Previous**   **Next** ▶

Unit 2 / Lesson 2 / Assignment 2

Street Fighter: HTML and CSS Walkthrough

🕐 Estimated Time: **1-2 hours**

In this assignment you will set up your HTML and CSS. You'll need to make the background for the body black and need a container div inside of the body with a white background that can display your Ryu-related content. You are going to need to use `.hide()` and `.show()`, so you will also need some classes and divs that will be hidden and shown. Ryu will be positioned within the container and the Hadouken will be positioned in relation to Ryu.

As you go through the steps below, you'll want to have index.html and main.css open in Sublime Text (and for this, you may find it useful to use View --> Layout --> Columns:2 so you can see both files side by side). You'll also want to open index.html in Chrome, so you can preview your page as you make changes.

Background Color

Setting the body background to black is easy enough. Inside of main.css, make that your first declaration, with

```css
body {
    background-color: black;
}
```

Main Container

Create the outermost container div - the white block across the middle of the page. In index.html, between the body tags, add a new div with a class of "main". Then, in main.css, add the following declaration:

```css
.main {
    background-color: white;
    margin-top: 110px;
    padding-left: 100px;
    min-width: 1200px;
    height: 500px;
    position: relative;
    z-index: 1;
}
```

These first three settings should make immediate sense: you want a white background for this div and you want it to be pushed down from the top of the page so the black background shows through. We set the left padding so that whatever is contained by this DIV will be pushed away from the left of the page. If you were trying to implement these margin and padding properties on your own, you'd use a trial and error process — ideally with Dev tools — to fine tune your margin-top.

The reasoning behind the min-width of 1200px is probably less obvious. Ultimately, this div is going to need to accommodate our Ryu images (which are all 659px wide) and our Hadouken, which will start out near the right of

this image and travel to the right of the screen. Setting the min-width to 1200px should accommodate these requirements (even if it makes the experience suboptimal for mobile users). If the user's browser window is narrower than 1200px, they'll have to scroll.

The minimum height of this div is set to 500px. This means that even if we don't have content in it, it will still display.

Ryu Container

Add another container div inside of div.main to hold the Ryu images. You will alternate between showing and hiding these three images. All three will need to be in the HTML (you'll take care of this in the next step), so it makes sense to contain these related elements in a common div. Inside of div.main, create a new div with a class of "ryu". In main CSS, create a style rule for this class, and set the float property to left. Ultimately, you will want div.ryu and div.hadouken (which we'll create in a moment) to float next to one another within the main div. Setting the float property to left on both classes will achieve this.

Ryu Image Divs

Now create divs with classes for the Ryu still, Ryu ready, and Ryu "Hadoukening" images (hold off on doing anything with the Ryu looking cool image; you'll implement that in the final assignment of this lesson). Inside of div.ryu, create three new divs as follows:

```
<div class="ryu">
  <div class="ryu-still"></div>
  <div class="ryu-ready"></div>
  <div class="ryu-throwing"></div>
</div>
```

Then, in main.css, add the following style declarations:

```css
.ryu {
    width: 659px;
    height: 494px;
}

.ryu-ready, .ryu-still, .ryu-throwing {
  width: 659px;
  height: 494px;
}

.ryu-ready, .ryu-throwing {
  display: none;
}

.ryu-ready {
  background-image: url('../images/ryu-ready-position.gif');
}

.ryu-still {
  background-image: url('../images/ryu-standing-still.png');
}

.ryu-throwing {
  background-image: url('../images/ryu-throwing-hadouken.png');
}
```

Note how you are avoiding repetition by assigning the width and height of 659 and 494 for all three classes at the same time. You then set .ryu-ready and .ryu-throwing to display:none because initially you want the div.ryu-still to display — you'll only display the other divs in response to events. You then set the background image for the three classes separately.

There's one additional background-image related property you need to set, not just for the Ryu classes, but also for the Hadouken class in the next step.

To guarantee that the images are not repeated if the div they are in is bigger than their dimensions, you must set the background-repeat property. In main.css, add the following rule:

```
.hadouken, .ryu-ready, .ryu-still, .ryu-throwing {
  background-repeat: no-repeat;
}
```

Look at a Preview

Before moving on to position the Hadouken div, preview what you have so far in Chrome. Open index.html in your browser. You should see the black background, a white container in the middle of the page, and inside of that, the image of Ryu standing still, pushed away from the left of the page. You're eventually going to be hiding and showing the three Ryu images in response to events, so preview that behavior now. Open Developer Tools and go to the Elements panel. Select the div.ryu-still and in the Styles panel set this div's display property to none. Then select div.ryu-ready and in the Styles panel set its display property to block. You should see the Ryu ready animated gif replace Ryu standing still. Finally, hide div.ryu-ready and show div.ryu-throwing to preview that image.

Hadouken Div

Now insert the Hadouken div into the page and write your styles. Inside div.main, below div.ryu, add a new div with a class of "hadouken". You need to set the background image, width, height, and display properties for this div much like you did for the Ryu image divs. In main.css, add the following rule:

```css
.hadouken {
  background-image: url('../images/hadouken.gif');
  width: 156px;
  height: 90px;
  display: none;
  float: left;
  position: absolute;
  z-index: 10;
}
```

Now figure out the settings needed to properly position this div. To better visualize how your elements are positioned, temporarily set the following style rule in our main.css:

```css
* {
  outline: red 1px solid;
}
```

This will put a red outline around all of your DOM elements. Now open index.html in Chrome, then in Developer tools, show div.ryu-throwing, hide div.ryu-still, and show div.hadouken. You should see Ryu in his throwing position, with the Hadouken positioned immediately to the right of div.ryu and at the top.

You need to position the Hadouken so it's next to Ryu's hands. To achieve this, the left and top properties seem ideal. You may recall that when you don't explicitly set the position property on an element, it defaults to static. However, the left and right properties only work with relatively or absolutely positioned elements. You'll set `div.hadouken`'s position property to `absolute`, and initially set the left and right properties to zero, then you'll
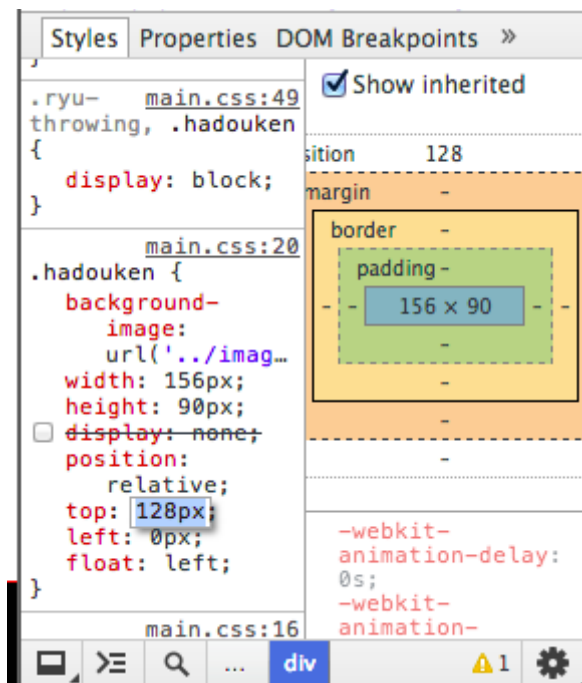
use Developer tools to refine those settings. Inside of main.css change your
`.hadouken` style so it looks like this:

```
.hadouken {
  background-image: url('../images/hadouken.gif');
  width: 156px;
  height: 90px;
  /*display: none;*/
  position: absolute;
  z-index: 10;
  top: 0px;
  left: 0px;
  float: left;
}
```

Note that you're temporarily commenting out `display: none` here, so it
displays when the page loads. You'll want to uncomment that line after you
figure out the top and left values.

Save these changes, and then reload index.html. Back in the elements panel,
you'll want to set `div.ryu-still` to `display: none`, and `div.ryu-throwing`
to `display: block`. Then, you should select `div.hadouken`, and in the styles
panel click on the value for the top property. After you click, your Styles panel
should look like this:

With the top value highlighted, you can use the up and down arrows on your keyboard to increase and decrease this value. To position the Hadouken, you'll probably need to iterate between adjusting the top and left properties. Setting top to 167px and left to 520px looks pretty good (but if you'd like the starting position of the Hadouken to be slightly different, feel free to go with different values). Now, go back to main.css and put in the top and left values for `.hadouken` that you just tested. Also uncomment the `display: none` line in `.hadouken`, and get rid of your `* { outline: red 1px solid;}` setting.

In the next assignment, you'll write the code for your events using jQuery and JavaScript. Before you move on, first commit your changes using your GitHub client. And if you want to check your code, you can see what index.html and main.css should look like at this point in this Gist.

Note

If you push your changes to GitHub and setup GH Pages to view your project in action be sure that you specify index.html in the URL bar.

This will be explained more at the end of 2.2.5 or feel free to ask your mentor about this.

☆ ☆ ☆ ☆ ☆   ·   Report a typo or other issue

✔ **Mark as completed**

‹ **Previous**   **Next** ›

›