University of East Anglia
COMPUTING SCIENCES

Lab 2: CSS Basics

Before attempting this lab, you will need to have completed lab **1B: Hands-On Coding with HTML**.

Learning objectives

By completing this lab, you should be able to:

- Explain how HTML elements and attributes such as **class** and **id** can be used as CSS selectors
- Create style rules to globally define background colour and text properties for HTML elements
- Create style rules to locally define text properties for elements of a particular class or id
- Apply CSS combinators such as descendant and child selectors
- Apply style rules defined in an external CSS file to a HTML document by linking them together
- Test CSS styles using various screen sizes

Background

Like HTML, CSS is not really a programming language. Though unlike HTML, it is not a markup language either — it is a **style sheet language**. This means that it lets you apply styles *selectively* to *elements* in HTML documents. For example, to make all text in *paragraph* elements that appear on a HTML page appear red, we can write the following code in a file named `style.css` and save it in a new directory named "`styles`".

```
p {
color: red;
}
```

We then need to link this CSS file to our HTML document (otherwise, the CSS will not be applied!). To apply CSS rules written in a separate style sheet file, we will write this line:

```
<link href="styles/style.css" rel="stylesheet">
```

and place it between the `<head>` and `</head>` tags in the HTML document. The `href="styles/style.css"` means the file is located in a directory called `styles` (which is in the same directory as the HTML document), and within the `styles` directory the file is called `style.css`.

For this module we will only use external stylesheets.

In this lab you will write some CSS style rules for the Mimo Yoga Studio webpages created in Lab 1B. The sitemap for the Mimo Yoga Studio website is shown below in Figure 1, which describes the structure of the website: a "Home" page with three main content pages: "Classes", "Schedule" and "Contact".

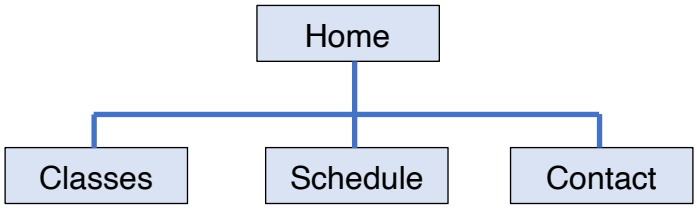
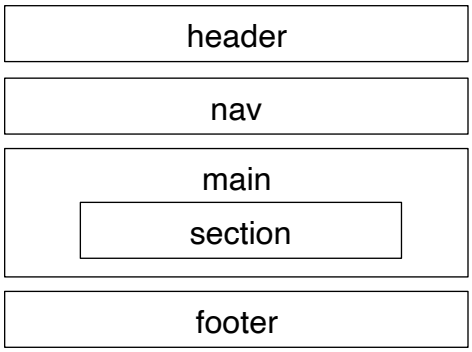


Figure 1. Mimo Yoga Studio sitemap

Make sure that your HTML files have the following structure:



The main goals for this lab are to:

1. Create an external style sheet named **yoga.css** for Mimi Yoga Studio website, containing some style rules.
2. Modify the Home page to utilise the external style sheet.
3. Modify the Classes page to have a consistent look and feel with the Home page.

Exercises

1. Create a new directory/folder in your student U: directory called **yogacss**. Copy *all* the files from previous session (which should be `index.html` and `classes.html` in the **yoga** directory) into this new **yogacss** directory.

2. Use a text editor (Visual Studio Code editor) to open the `index.html` file in this **yogacss** directory. We will use link element to import our external stylesheet. Copy the following code and paste it inside the `<head>` element. Then save the file. We will create this external stylesheet next.

```
<link rel="stylesheet" href="yoga.css">
```

3. Now use the same text editor to create a **new file** called `yoga.css` and save it in the same directory/folder. This style sheet should contain the following rules – you may wish to refer to the CSS page on MDN for reference: https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/CSS_basics

   a) A universal selector (i.e. an asterisk `*`) and set `margin` and `padding` to `0`. [**Note**: universal selector is used to select everything in the document. Here we are resetting the default browser rendering]

   b) The webpage/document background is white: we will be using the `html` selector for this job. [HINT: use `background-color` property. The value can be just `white` or hex value `#fff`]

   c) The global styles of the document are (use the `body` element selector):
      - background colour `#EFF4F7`
      - text colour `#3F2860`
      - font family `Verdana, Arial, Helvetica, sans-serif` font
      - font size 16px
      - line height 1.5

      (**Hint:** you need to use the American spelling “color”)

      Now open the `index.html` using Chrome. Your webpage should look similar to Figure 2 below:

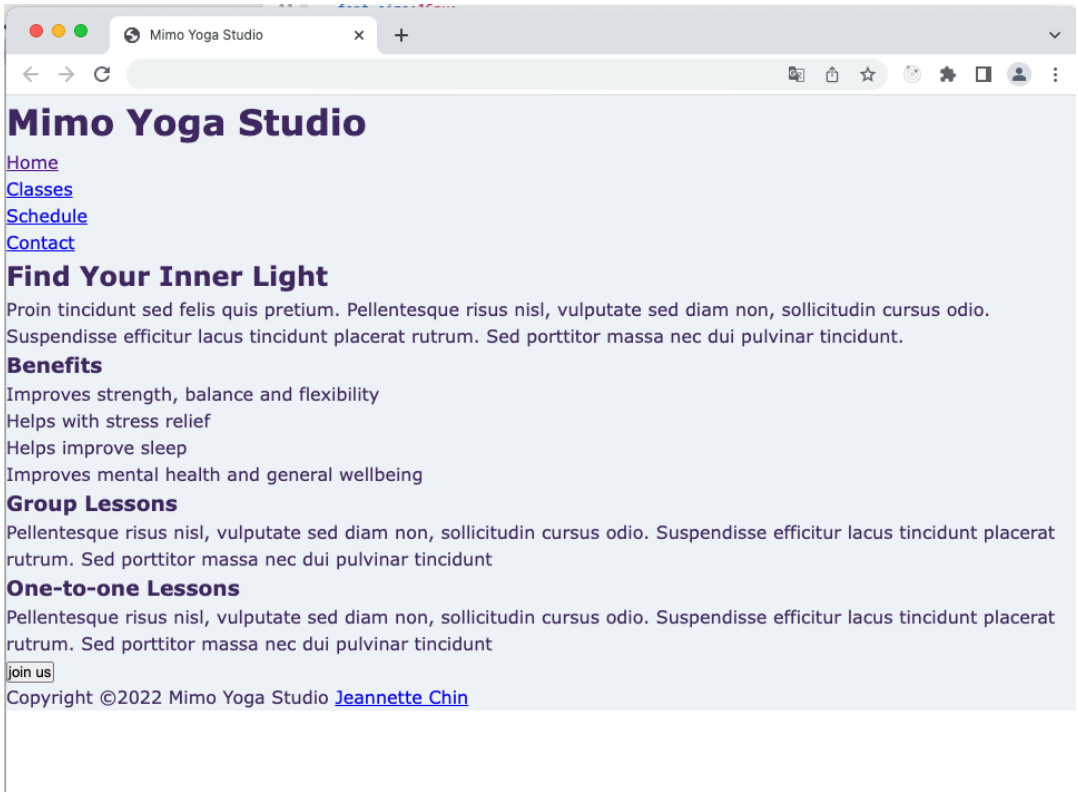


Figure 2. Output of index.html

Next, we will write some rules to style our header and some other general/global styles.

d) Styles for the `header` element: background colour `#CFBECE` with text centred, and `height` is 100px. (**Hint:** use `text-align` property and you need to use the American spelling "center")

e) Styles for the direct children of this `header` element: `h1` element: line height is 200% and padding top is 16px. [Hint: use child combinator, see MDN]

   Now refresh the webpage. Your result should look similar to Figure 3 below:

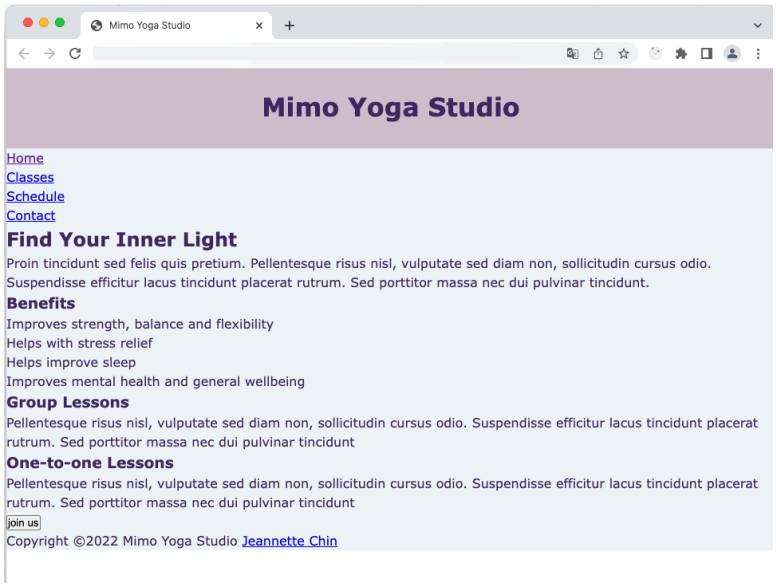


Figure 3. Output of index.html

f) The unordered list has no bullet points style. [**Hint**: use `list-style-type` property on `ul` selector] see MDN

g) Style for `p` selector: margin values: top and bottom is `10px`, left and right is `auto`. If you are not familiar with this configuration, see MDN

h) The anchor tag styles (i.e. `a` selector) :
   a) has no decorative line on the text (Hint: use `text-decoration` property on `a` selector) MDN
   b) the text colour for the links and their visited state is `#3F2860` (HINT: use CSS pseudo-class `a:link` and `a:visited` selectors)
   c) the colour of the hover state is `#729479`

Now refresh your browser. Your webpage should look similar to Figure 4 below:

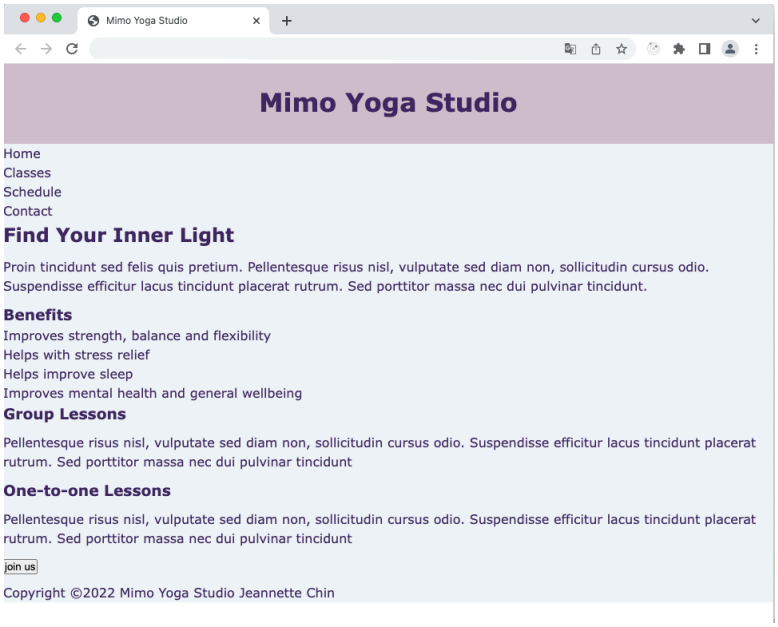


Figure 4. Output of index.html

Now test the mouse hover effect. Does it work? What colour did you see? Next, we will style the navigation bar to appear horizontally.

i) Styles for `nav` selector:
   a) centred text, `height` is `100px`, `padding` is `8px`. [Hint: use `text-align` property MDN]

Refresh the page. Your result should look similar to Figure 5 below.

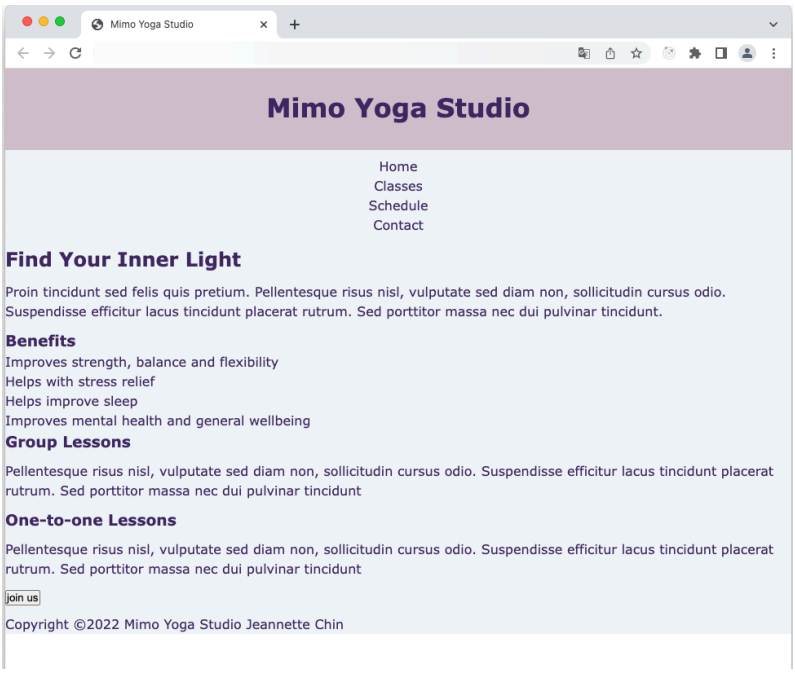


Figure 5. Output of index.html

The navigation links are now centred. But they appear in a vertical list. Why?

Next, we will configure the navigation links to appear horizontally.

j) A new CSS **class** called "`nav`" with the following rules:
   a) The `display` property has a value of `inline-block`
   b) Padding is 24px

Now add this `nav` class to `nav` list items `<li>` on your `index.html` page. Below is a fragment of my code:

```
<nav>
    <ul>
        <li class="nav"><a href="index.html">Home</a></li>
        <li class="nav"><a href="classes.html">Classes</a></li>
        <li class="nav"><a href="schedule.html">Schedule</a></li>
        <li class="nav"><a href="contact.html">Contact</a></li>
    </ul>
</nav>
```

Figure 6. index.html code fragment

Once you are done, refresh the page. Your result should look similar to Figure 7 below:

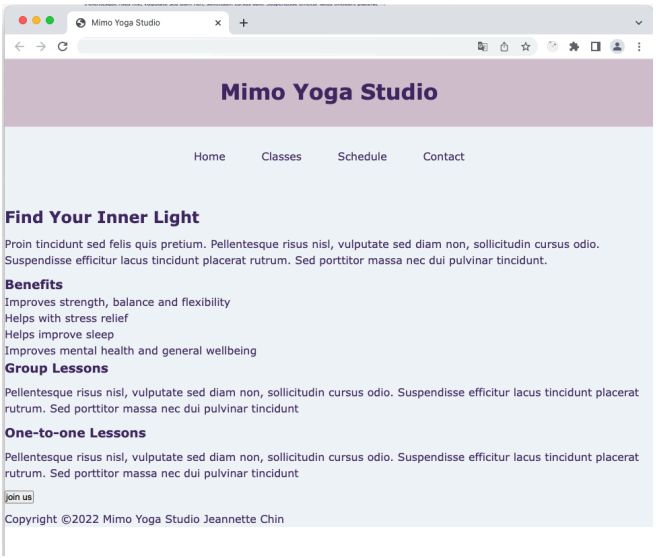


Figure 7. Output of index.html

k) Style for `main` selector:
   a) `margin` values are top and bottom is `auto`, left and right is `20%` MDN
   b) `padding` is `16px` MDN
   c) `border` values are `2px solid` in white MDN
   d) We would like to set rounded corner as well, use `border-radius` property and set the value 20px. MDN

Once you are done, refresh the page. Your result should look similar to Figure 8 below.

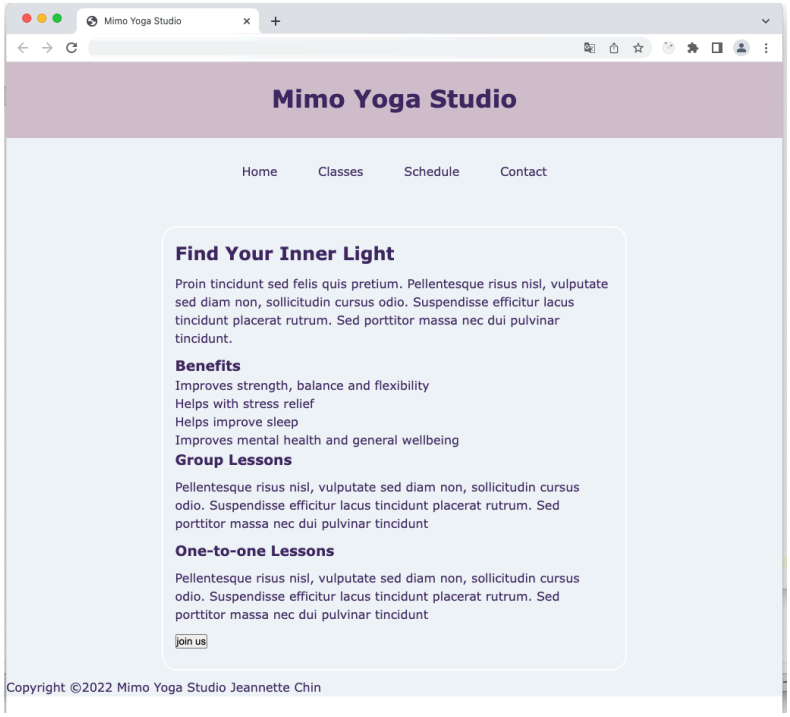


Figure 8. Output of index.html

So far we have configured our `main` element to appear in the middle of the page. Next, we will create two boxes inside this `main` element. But first, we create the outer container of our boxes.

l) Style for `section` selector:
   a) `padding` values are top and bottom 20px, left and right is 0 MDN
   b) `margin` values: top is 10px, right, bottom and left is auto. MDN
   c) Background colour is `#fff9e8`. MDN

m) Style for the direct children of `section` selector: `p` element has the content aligned in the middle. [HINT: use `text-align` property]

Now refresh the page. Your result should look similar to Figure 9 below.

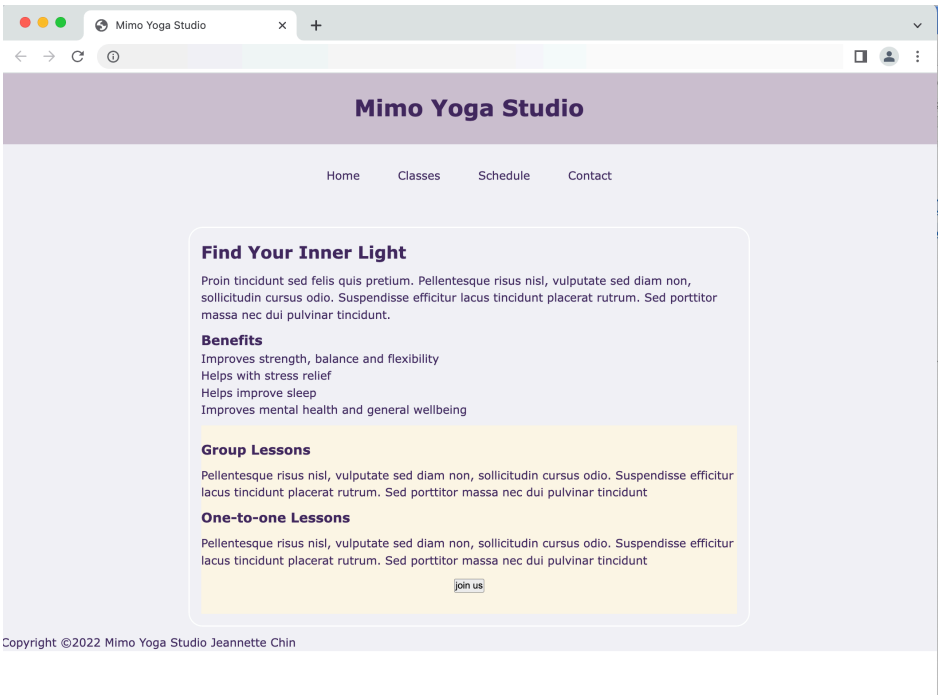


Figure 9. Output of index.html

The outer container of our boxes (i.e. section element) is aligned nicely to its parent. Which element is its parent?

Next, we will style our two boxes such that they are placed next to each other. We will create a new CSS class called "`service`".

n) Style for new CSS class named `service`:
   a) the `display` is inline-block
   b) `width` is 47%
   c) background colour is white
   d) padding: top and bottom is 24px, left and right is 11px
   e) margin: top is auto, right is auto, bottom is 20px and left is 5px,
   f) rounded corner with 20px [Hint: use `border-radius` property]
   g) `box-sizing` reference to `border-box`. Note: we want the browser to take into account of the padding set in the border. MDN

Next, we will need to add this CSS class to our `article` elements so that they appear like a box. Below is a fragment of my code:

```
<section>
    <article class="service">
        <h3>Group Lessons</h3>
        <p>Pellentesque risus nisl, vulputate sed
        Sed porttitor massa nec dui pulvinar tinc
    </article>
    <article class="service">
        <h3>One-to-one Lessons</h3>
        <p>Pellentesque risus nisl, vulputate sed
        Sed porttitor massa nec dui pulvinar tinc
    </article>
```

Figure 10. index.html code fragment

Now refresh the page. Your result should look similar to Figure 11 below.

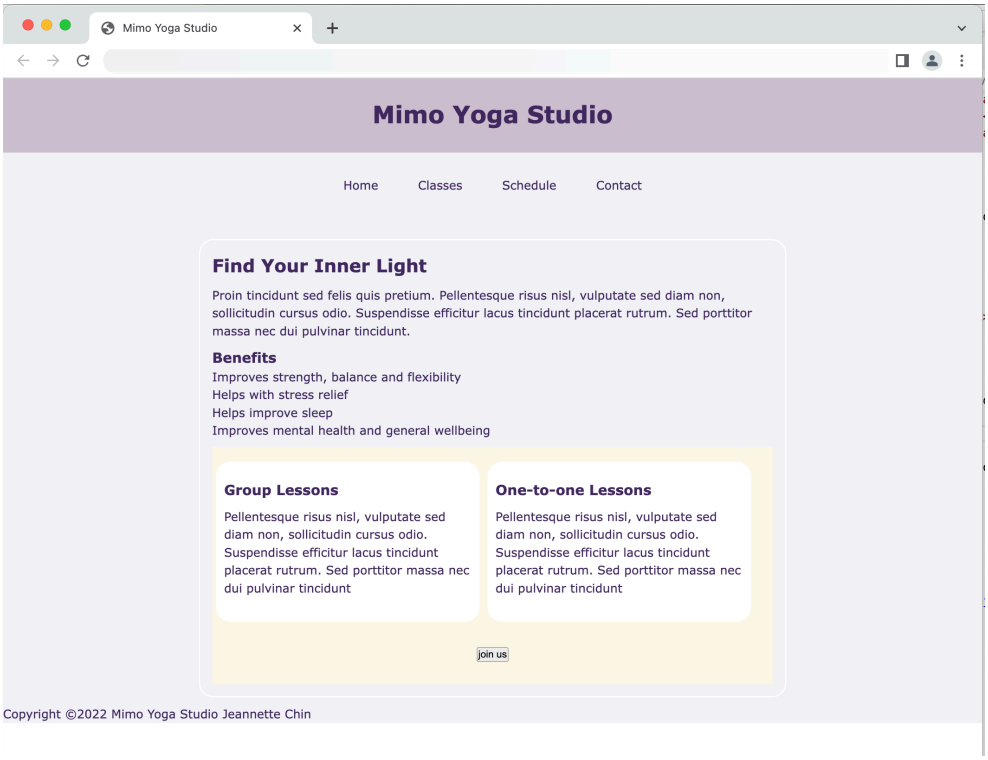


Figure 11. The output of index.html

You may notice the button needs attention. We will style this button and make it looks more prominent on the page. To do this we will create a new CSS class called "`btn`"

o) Style for `btn` class:
   a) `Display` is inline block
   b) Background colour is `#B8195A`
   c) The text colour is white
   d) Transform the text to upper case
   e) `Padding`: top and bottom is 6px, left and right is 10px

f) `Font` weight is 700
g) `Font` size is 16px
h) `Border`: 2px solid , colour is `#B8195A`
i) We will configure the rounded corner to be 10px
j) `Cursor` is `pointer`

After the above rules have been configured, add this `btn` class to the `button` element. Below is my code:

```
<p>
    <button class="btn"> join us </button>
</p>
```

Figure 12. index.html code fragment

Refresh the page. Your result should look similar to Figure 13 below

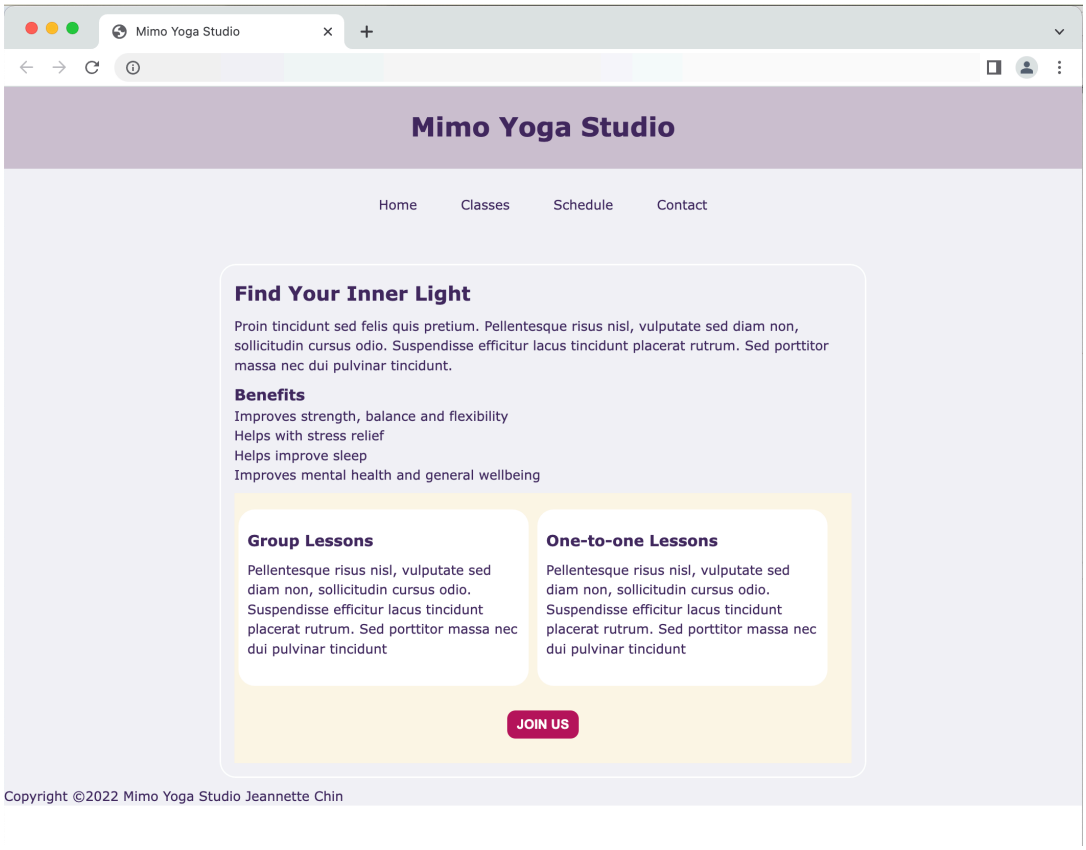


Figure 13. The output of index.html

Next, we will style our footer element.

p) Styles for the `footer` selector:
   a) background colour is `#573A56`
   b) font size `0.7em`
   c) text is centred
   d) text colour is #fff9e8
   e) padding is 16px
   f) margin top should be 32px

Refresh the page. Your result should look similar to Figure 14 below

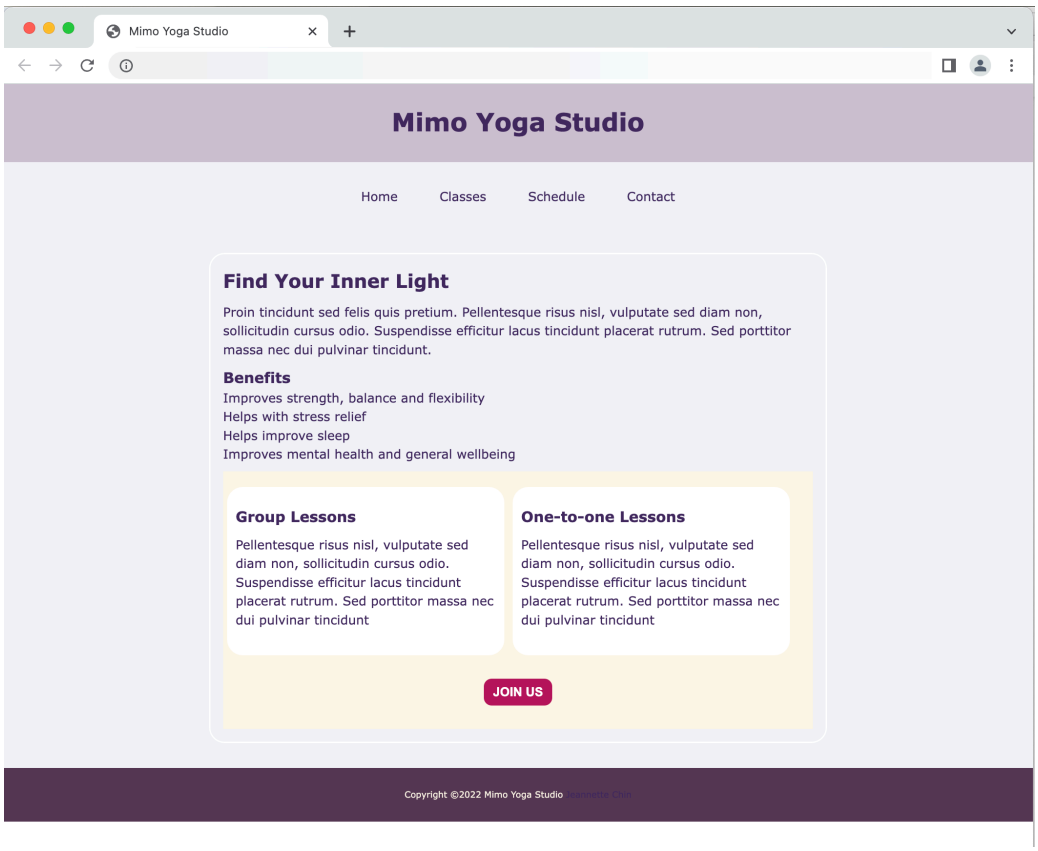


Figure 14. The output of index.html

Notice the appearance is getting along nicely. However, the email link in the footer section is hard to read. We will fix this by using descendant combinator (MDN) and pseudo-class (MDN)

q) Style for `footer` anchor link: text colour is `#fff9e8` [Hint: use `a:link` selector]
r) Style for `footer` anchor hover: text colour is `#e8fff9`. [Hint: use `a:hover` selector]

Refresh the page. Your result should look similar to Figure 15 below.

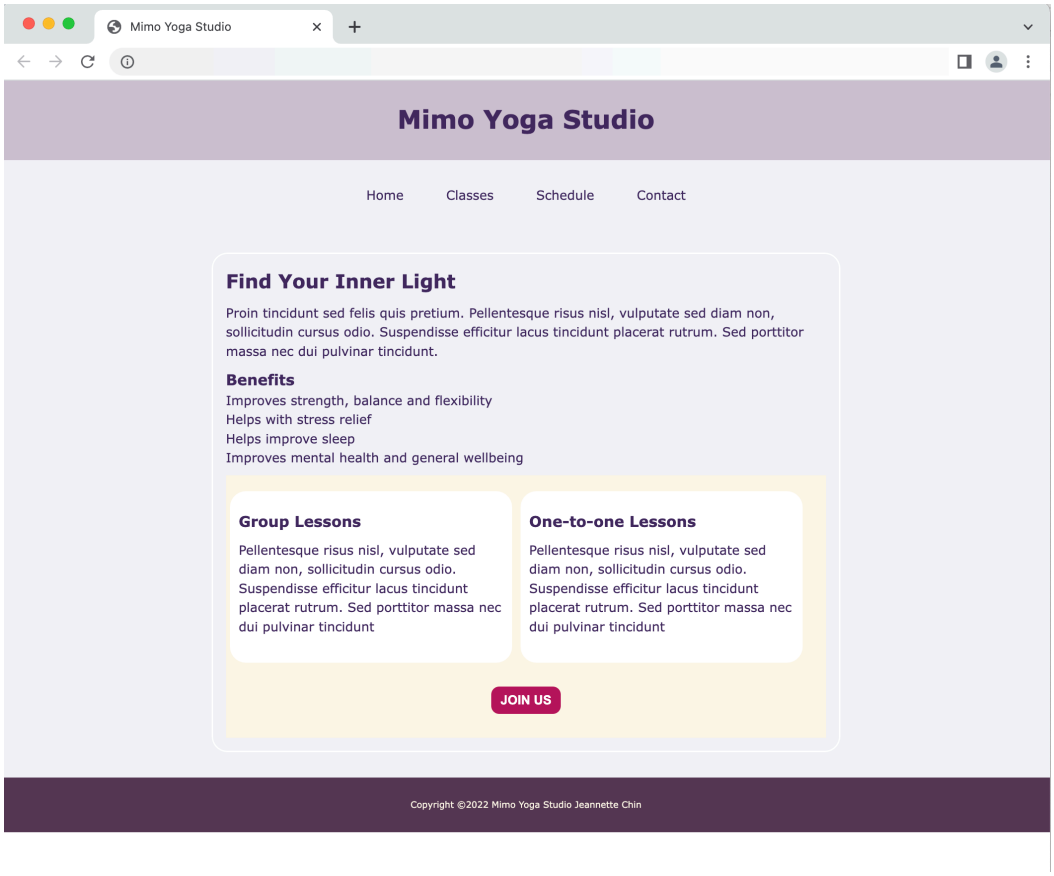


Figure 15. The output of index.html

Test the mouse hover effect on the email link. Is your result as expected?

The content inside the `main` element could do with some styling. Particular for the Benefits section. We will now create some rules to style this section. Back to your CSS file, create a new CSS class called `benefits-list`.

s) Style for CSS **class** named `benefits-list` :
   a) the `display` style is `square` [Hint: use `list-style-type` property]
   b) `margin left` is 18px. [Hint: use `margin-left` properties]

Next, add this CSS class to the unordered list `ul`. Below is my code.

```
<h3>Benefits</h3>
<ul class="benefits-list">
<li>Improves strength, balance and flexibility</li>
<li>Helps with stress relief</li>
<li>Helps improve sleep</li>
<li>Improves mental health and general wellbeing</li>
</ul>
```

Figure 16. index.html code fragment

Refresh the page. Your result should look similar to Figure 17 below.

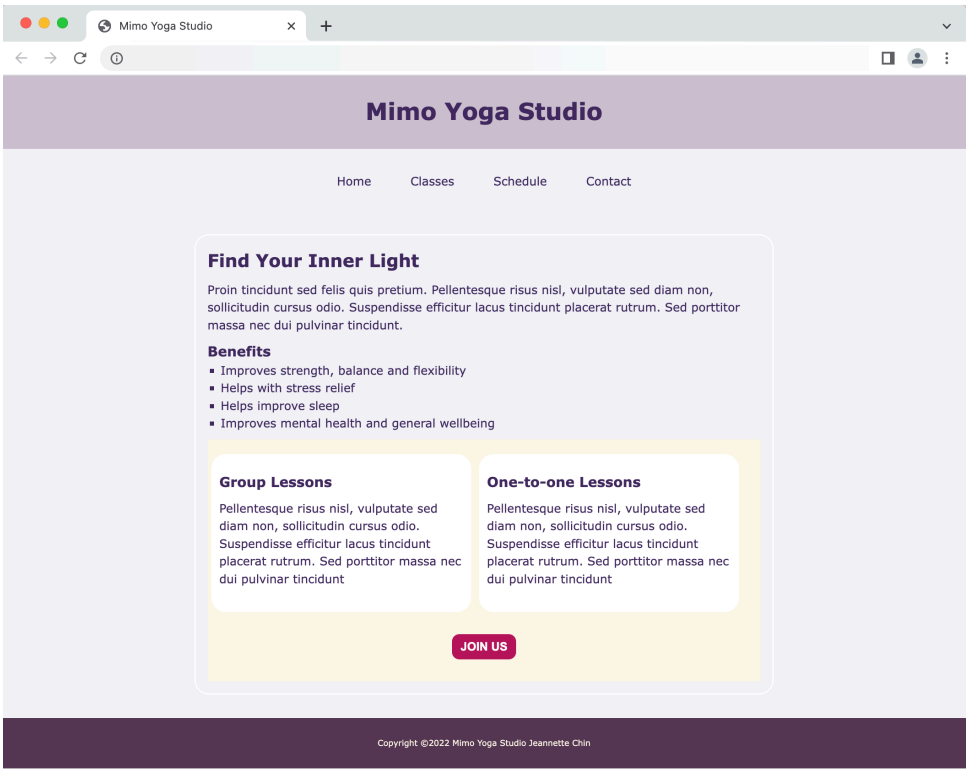


Figure 17. The output of index.html

Check the syntax with the CSS validator (https://jigsaw.w3.org/css-validator/). Correct and retest if necessary.

Next we are going to work on images using CSS.

4. On your text editor, open the `classes.html` file in the **yogacss** directory. Edit this file to add the "`yoga.css`" style sheet. [**Hint**: use link element]. Save the `classes.html` file and test it using Chrome browser. Your *target* output should look similar to the screenshot shown below in Figure 18. [**Hint**: refer to Figure 21]

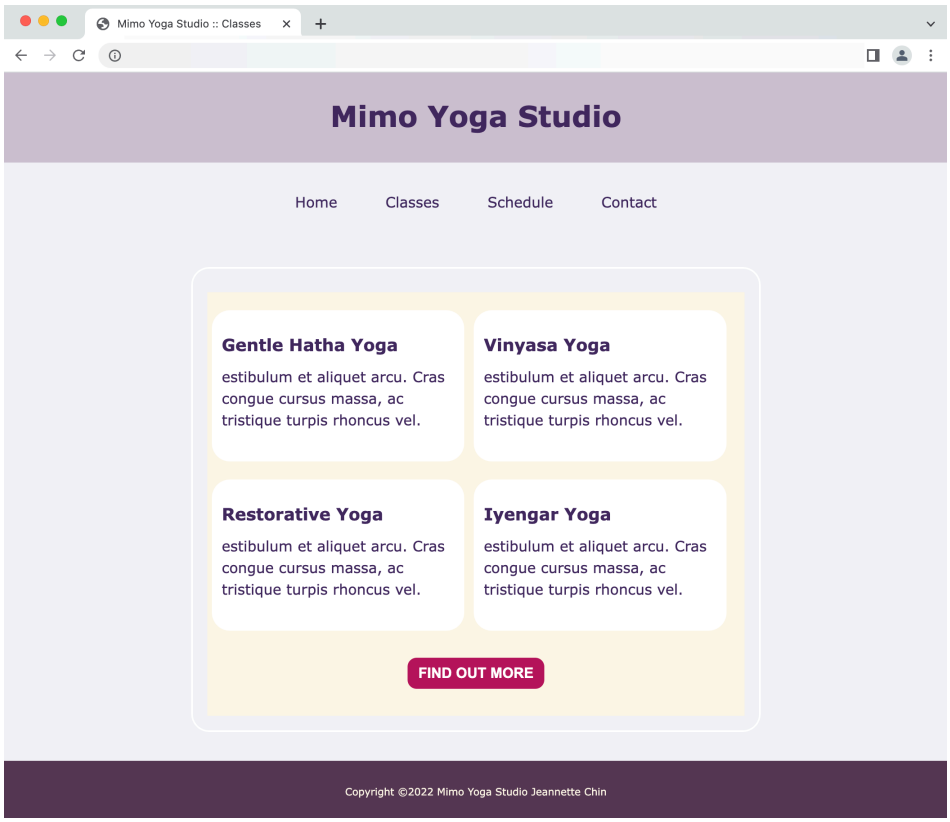


Figure 18. The `classes.html` page of Mimo Yoga Studio after applying CSS.

5. Next, we will add some images to this `classes.html` page. Download the `images.zip` from Blackboard and save this `images.zip` file in **yogacss** directory, as shown in Figure 19.

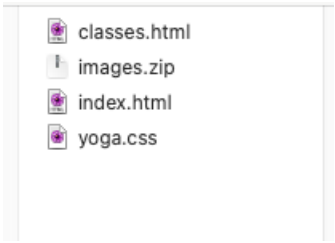


Figure 19. images.zip file

6. Unzip the `images.zip` file. You should see that there is a subfolder named `images` in your **yogacss** directory, as shown in Figure 20.

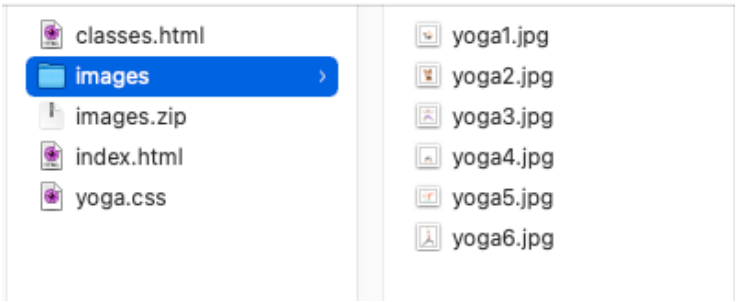


Figure 20. Unzipped images.zip

7. Now examine your `classes.html` and make sure that there are four `article` elements already coded here. See Figure 21.

```
<main>
    <section>
        <article class="service">
            <h3>Gentle Hatha Yoga</h3>
            <p>estibulum et aliquet arcu. Cras congue cursus massa, ac tristique turpis rhoncus vel. </p>
        </article>
        <article class="service">
            <h3>Vinyasa Yoga</h3>
            <p>estibulum et aliquet arcu. Cras congue cursus massa, ac tristique turpis rhoncus vel. </p>
        </article>
        <article class="service">
            <h3>Restorative Yoga</h3>
            <p>estibulum et aliquet arcu. Cras congue cursus massa, ac tristique turpis rhoncus vel. </p>
        </article>
        <article class="service">
            <h3>Iyengar Yoga</h3>
            <p>estibulum et aliquet arcu. Cras congue cursus massa, ac tristique turpis rhoncus vel. </p>
        </article>
        <p>
            <button class="btn"> Find out more </button>
        </p>
    </section>
</main>
```

Figure 21. classes.html code fragment

8. We will group our images using `figure` element (see MDN). Edit the `classes.html` as follows:
    a. For each `article` element, add a child element `figure`, and for each child element `figure` add a child element `figcaption` . Your code should look similar to Figure 22.

```
<main>
    <section>
        <article class="service">
            <figure>
                <figcaption>
                </figcaption>
            </figure>
            <h3>Gentle Hatha Yoga</h3>
            <p>estibulum et aliquet arcu. Cras congue cursus massa, ac tristique turpis rhoncus vel. </p>
        </article>
        <article class="service">
            <figure>
                <figcaption>
                </figcaption>
            </figure>
            <h3>Vinyasa Yoga</h3>
            <p>estibulum et aliquet arcu. Cras congue cursus massa, ac tristique turpis rhoncus vel. </p>
        </article>
        <article class="service">
            <figure>
                <figcaption>
                </figcaption>
            </figure>
            <h3>Restorative Yoga</h3>
            <p>estibulum et aliquet arcu. Cras congue cursus massa, ac tristique turpis rhoncus vel. </p>
        </article>
        <article class="service">
            <figure>
                <figcaption>
                </figcaption>
            </figure>
            <h3>Iyengar Yoga</h3>
            <p>estibulum et aliquet arcu. Cras congue cursus massa, ac tristique turpis rhoncus vel. </p>
        </article>
        <p>
            <button class="btn"> Find out more </button>
        </p>
    </section>
</main>
```

Figure 22. classes.html code fragment

b. Next, we will add an `img` element (MDN) as a child element to `figure` element. Your code should look similar to Figure 23.

```
<main>
    <section>
        <article class="service">
            <figure>
                <img src="" alt="">
                <figcaption>
                </figcaption>
            </figure>
            <h3>Gentle Hatha Yoga</h3>
            <p>estibulum et aliquet arcu. Cras congue cursus massa, ac tristique turpis rhoncus vel. </p>
        </article>
        <article class="service">
            <figure>
                <img src="" alt="">
                <figcaption>
                </figcaption>
            </figure>
            <h3>Vinyasa Yoga</h3>
            <p>estibulum et aliquet arcu. Cras congue cursus massa, ac tristique turpis rhoncus vel. </p>
        </article>
        <article class="service">
            <figure>
                <img src="" alt="">
                <figcaption>
                </figcaption>
            </figure>
            <h3>Restorative Yoga</h3>
            <p>estibulum et aliquet arcu. Cras congue cursus massa, ac tristique turpis rhoncus vel. </p>
        </article>
        <article class="service">
            <figure>
                <img src="" alt="">
                <figcaption>
                </figcaption>
            </figure>
            <h3>Iyengar Yoga</h3>
            <p>estibulum et aliquet arcu. Cras congue cursus massa, ac tristique turpis rhoncus vel. </p>
        </article>
        <p>
            <button class="btn"> Find out more </button>
        </p>
    </section>
</main>
```

Figure 23. classes.html code fragment

Notice I have included `alt` attribute. Question: What is this for?

c. Now for each `img` element, we add the source (`src`) to the image. The path to the image files is: `images/<the name of the image>`. Here is my code – see Figure 24.

```
<main>
    <section>
        <article class="service">
            <figure>
                <img src="images/yoga1.jpg" alt="">
                <figcaption>
                </figcaption>
            </figure>
            <h3>Gentle Hatha Yoga</h3>
            <p>estibulum et aliquet arcu. Cras congue cursus massa, ac tristique turpis rhoncus vel. </p>
        </article>
        <article class="service">
            <figure>
                <img src="images/yoga2.jpg" alt="">
                <figcaption>
                </figcaption>
            </figure>
            <h3>Vinyasa Yoga</h3>
            <p>estibulum et aliquet arcu. Cras congue cursus massa, ac tristique turpis rhoncus vel. </p>
        </article>
        <article class="service">
            <figure>
                <img src="images/yoga3.jpg" alt="">
                <figcaption>
                </figcaption>
            </figure>
            <h3>Restorative Yoga</h3>
            <p>estibulum et aliquet arcu. Cras congue cursus massa, ac tristique turpis rhoncus vel. </p>
        </article>
        <article class="service">
            <figure>
                <img src="images/yoga4.jpg" alt="">
                <figcaption>
                </figcaption>
            </figure>
            <h3>Iyengar Yoga</h3>
            <p>estibulum et aliquet arcu. Cras congue cursus massa, ac tristique turpis rhoncus vel. </p>
        </article>
        <p>
            <button class="btn"> Find out more </button>
        </p>
    </section>
</main>
```

Figure 24. classes.html code fragment

d. Next, for each `article` element, move the children `h3` and `p` elements and place them inside the `figcaption` element as shows in Figure 25. Your results should look similar to Figure 26.

```
<main>
    <section>
        <article class="service">
            <figure>
                <img src="images/yoga1.jpg" alt="">
                <figcaption>
                </figcaption>
            </figure>
            <h3>Gentle Hatha Yoga</h3>
            <p>estibulum et aliquet arcu. Cras congue cursus massa, ac tristique turpis rhoncus vel. </p>
        </article>
        <article class="service">
            <figure>
                <img src="images/yoga2.jpg" alt="">
                <figcaption>
                </figcaption>
            </figure>
            <h3>Vinyasa Yoga</h3>
            <p>estibulum et aliquet arcu. Cras congue cursus massa, ac tristique turpis rhoncus vel. </p>
        </article>
        <article class="service">
            <figure>
                <img src="images/yoga3.jpg" alt="">
                <figcaption>
                </figcaption>
            </figure>
            <h3>Restorative Yoga</h3>
            <p>estibulum et aliquet arcu. Cras congue cursus massa, ac tristique turpis rhoncus vel. </p>
        </article>
        <article class="service">
            <figure>
                <img src="images/yoga4.jpg" alt="">
                <figcaption>
                </figcaption>
            </figure>
            <h3>Iyengar Yoga</h3>
            <p>estibulum et aliquet arcu. Cras congue cursus massa, ac tristique turpis rhoncus vel. </p>
        </article>
        <p>
            <button class="btn"> Find out more </button>
        </p>
    </section>
</main>
```

Figure 25. before

```
<main>
    <section>
        <article class="service">
            <figure>
                <img src="images/yoga1.jpg" alt="">
                <figcaption>
                    <h3>Gentle Hatha Yoga</h3>
                    <p>estibulum et aliquet arcu. Cras congue cursus massa, ac tristique turpis rhoncus vel. </p>
                </figcaption>
            </figure>
        </article>
        <article class="service">
            <figure>
                <img src="images/yoga2.jpg" alt="">
                <figcaption>
                    <h3>Vinyasa Yoga</h3>
                    <p>estibulum et aliquet arcu. Cras congue cursus massa, ac tristique turpis rhoncus vel. </p>
                </figcaption>
            </figure>
        </article>
        <article class="service">
            <figure>
                <img src="images/yoga3.jpg" alt="">
                <figcaption>
                    <h3>Restorative Yoga</h3>
                    <p>estibulum et aliquet arcu. Cras congue cursus massa, ac tristique turpis rhoncus vel. </p>
                </figcaption>
            </figure>
        </article>
        <article class="service">
            <figure>
                <img src="images/yoga4.jpg" alt="">
                <figcaption>
                    <h3>Iyengar Yoga</h3>
                    <p>estibulum et aliquet arcu. Cras congue cursus massa, ac tristique turpis rhoncus vel. </p>
                </figcaption>
            </figure>
        </article>
        <p>
            <button class="btn"> Find out more </button>
        </p>
    </section>
</main>
```

Figure 26. after

e. At this point, save the file and open it using Chrome. Your output should be similar to Figure 27. You will notice that for smaller screens, the images are not in correct places, but we will fix this with CSS.

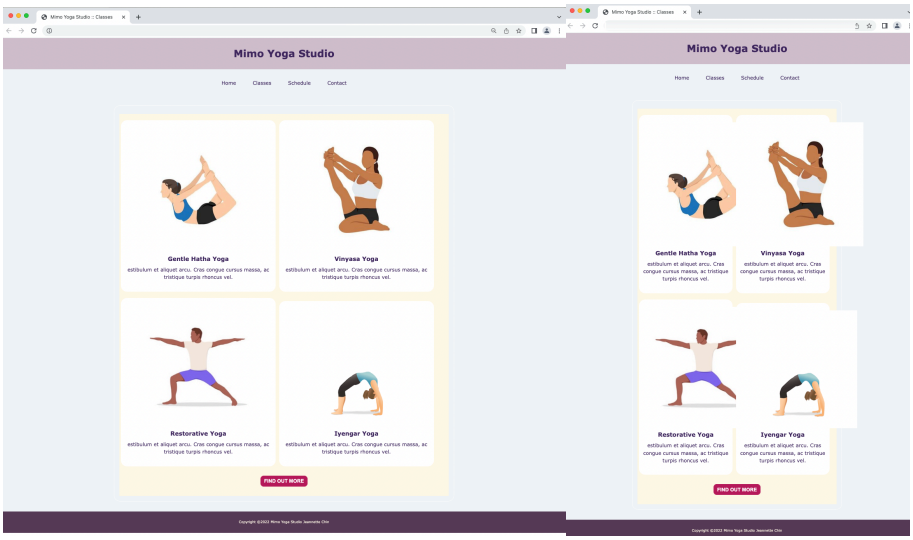


Figure 27. classes.html with images

f. Our `classes.html` markups look about right (obviously can be improved). We will create some style rules for the new elements we have just created. First add a new generic rule for the `figure` selector in your `yoga.css`, to reset the browser default `margin` to 0.

```
figure {
  margin: 0;
}
```

Figure 28. yoga.css

g. Next, we will fix the images. In your CSS, create a rule for `img` selector as follow:
   i. Width as 100%
   ii. Height is auto

h. Now save the `yoga.css` and refresh the browser. Test this with various screen sizes.

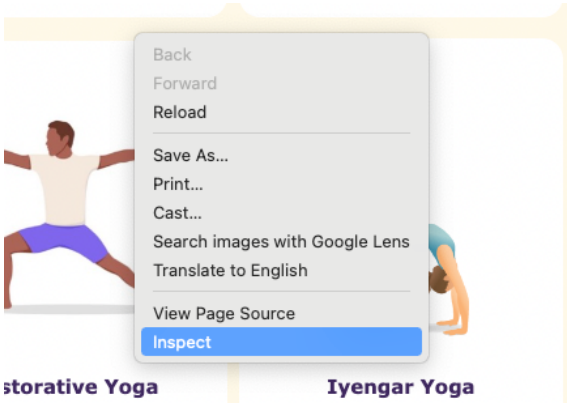


Figure 29. right click and select inspect option

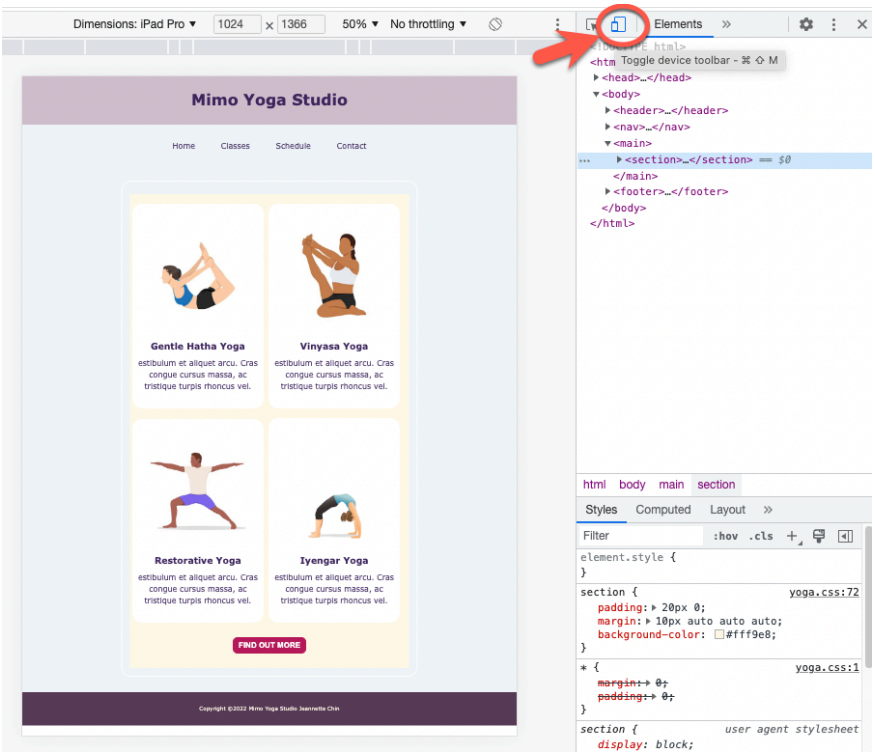


Figure 30. Select screen size options on Chrome

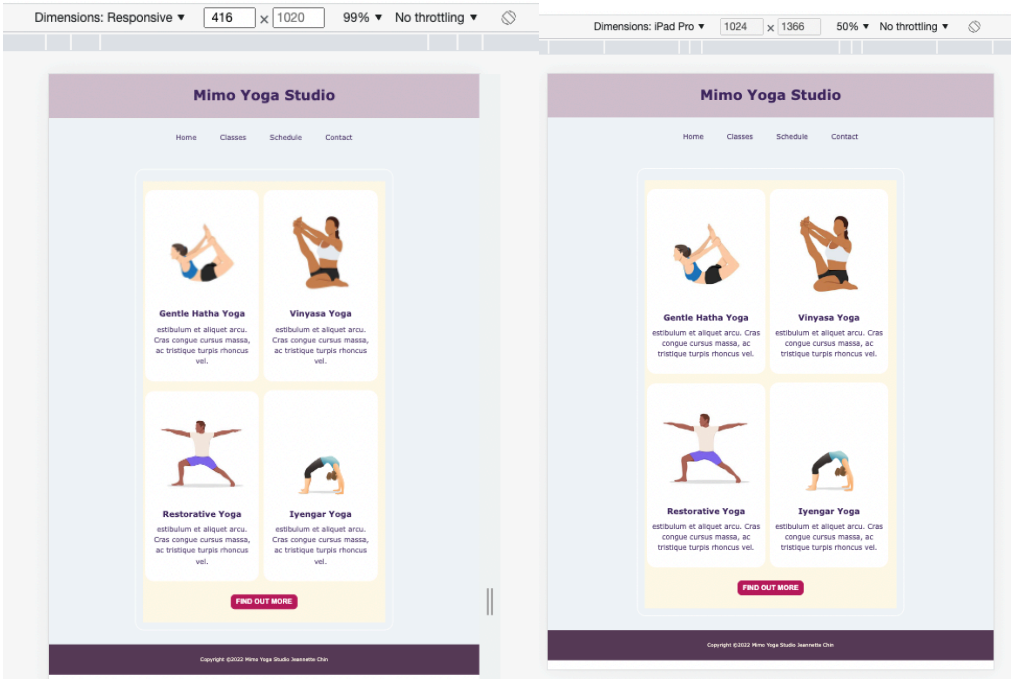


Figure 31. outputs depending on the selected screen size.

Well done! You have now successfully applied style rules via external CSS to HTML documents.