I. Introduction

The goal of this exercise is to practice the HTML and CSS web technologies you learned in this course. For Part 1, you are given screenshots and a description of a specific web page to replicate. You will recreate the web page exactly as shown in the provided screenshots. For Part 2, you will create your own list-based article with any styling you’d like.

II. Part 1: Backend Frameworks Listicle: listicle.html

A listicle is a somewhat trendy type of article that uses a list as its thematic structure.

For Part 1, you will create a listicle web page that describes the “3 Web Development backend (server-side) frameworks you should learn”. You should create the web page in listicle.html with styles defined in listicle-style.css.

A full-size screenshot of the entire page (with a screen width of 1280px) is linked here (ignore the white space around the header image).

a. Prerequisites

Note that you do not have to type in the contents of the page. The text of the listicle is contained within listicle-content.txt, which is included in your as3-supplementary.zip which contains starter code. Copy and paste the text of listicle-content.txt into your listicle.html file, then add the HTML tags necessary to style the page. The images are also in the images/ directory of the as3-supplementary.zip starter code. Maintain the folder structure with all files and folders since you are going to return it back to us when you finish.

b. Section measurements

The following screenshot gives the basic measurements for the webpage:
1. **Django**

   **Python**

   Django is a Model-View-Template framework that uses Python for web development. It’s used by some of the big names such as Pinterest, Mozilla, and Instagram. Django is a free, open-source, high-end framework written in the famously simple, flexible, and relatively easy-to-learn Python programming language. Initially released in 2005, it’s used by thousands of programmers every year, owing to its friendliness to beginners and advanced users alike.

   A “batteries included” approach to programming is a hallmark of both Python and Django, and it makes the framework an excellent choice for any developer who wants to build modern and robust web applications with a minimal amount of code. It’s a highly structured framework – it forces you to do things the Django way, rather than your own way. When building a site that requires a strong and secure foundation that protects transactions and sensitive data, such as an ecommerce site, Django is a great framework to use. It hides your source code by default, and it’s often one of the first frameworks to respond to a new vulnerability. It has a solid user authentication model with the ability to configure different users, and its core team also usually alerts other frameworks of patches they should make to maintain security. This all makes Django a very suitable choice when security is a top priority.

   **PROS:** Rapid development, security, great community, easy to learn (python inside), easy database management.

   **CONS:** Speed, not the right choice for really small sites, like static one-pagers or microservices, monolithic; components deployed together.

2. **Spring**

   **Java**

   Spring is a Model-View-Controller framework that uses Java, the all-time popular language. It’s used by websites like Wix, TicketMaster, and BillGuard. Spring possesses a lot of sister projects that boost its performance and let you scale your business easily. The fact that it uses Java, a strongly typed language, is a serious pro for many web developers. The learning curve might be quite steep though, especially if you...
c. Description of Work

c1. Font faces, sizes, and colors

Page Title:

- The font face is 'Roboto Slab' and the fallback font is serif.
- The font size is 36px.
- The font weight is normal (not bold).
- The font color is white.
- The title is center aligned (horizontally) to the header image and is 40% of the total height from the top of the image.

Profile Text:

- The font face is 'Source Sans Pro' and the fallback font is sans-serif.
- The font size is 18px.
- The font weight for the author name ("Pavlos Antoniou") is bold.
- The font style for the last updated ("Last updated Apr 7, 2019") is italic.
- The font color is rgb(51, 51, 51).
- The line height is 24px.
- The gray line under the “Last updated” line is 10px under the text, 2px wide, and the color is #e6e6e6.

Body Text:

1. **Django**  
   Python
   
   Django is a Model-View-Template framework that uses Python for web development. It's used by some of the big names such as Pinterest, Mozilla, and Instagram. Django is a free, open source, high-end framework written in the famously simple, flexible, and relatively easy-to-learn Python programming language. Initially released in 2005, it’s used by thousands of programmers every year, owing to its friendliness to beginners and advanced users alike.
Heading

• The font face is 'Roboto Slab' and the fallback font is serif.
• The font size is 36px.
• The font weight is bold.
• The number’s color is rgba(0, 0, 0, 0.25).
• The title color is rgb(51, 51, 51).

Subheading

• The subheading under the heading is a link and is described in the Links section (below).

Paragraphs

• The font face is 'Source Sans Pro' and the fallback font is sans-serif.
• The font size is 18px.
• The line height 32px.
• The font color is rgb(51, 51, 51).

Pros and Cons

| PROS: Rapid development, security, great community, easy to learn (python inside), easy database management. 
| CONS: Speed, not the right choice for really small sites, like static one-pagers or microservices, monolithic, components deployed together. |

• The phrases “PROS:” and “CONS:” are bold.

c2. Links

Here is a description of how links should work on your page:

• Each subheading of the page is a link. The URL for each link is listed in listicle-content.txt.
• The font face is 'Roboto Slab' and the fallback font is serif.
• The font size is 16px.
• The link color is #42b4d6.
• The link is not bold, and the link is not underlined unless you hover over the link.
• When you hover over the link, the link becomes underlined.
• Hints:
  o To remove an underline, you can use CSS property text-decoration: none;
  o To add an underline, you can use CSS property text-decoration: underline;
c3. Images

Header image

- The header is a background image (images/header.jpg).
  - **Note:** To reference the image from the css/ folder, the path needs to be ../images/header.jpg
- The background does not repeat.
- The background’s size is set to cover.
- The background position is anchored to the top.
- The background’s attachment is fixed.
- Its height is 500px.
- There is a semi-transparent color overlay on top of the header image, which is \texttt{rgba(0, 0, 0, .3)}.
  - **Note:** You can achieve this using the following CSS rule:
    
    \begin{verbatim}
    background-image: linear-gradient(rgba(0, 0, 0, .3), rgba(0, 0, 0, .3)), url(../images/header.jpg);
    \end{verbatim}

Profile image

- The profile image is images/avatar.png.
  - **Note:** To reference the image from the css/ folder, the path needs to be ../images/avatar.png
- Its width is 100px.
- It has rounded corners with radius 100%.
- It has a white border that is 1px thick.

d. Requirements

For full credit, please abide by the following:

**Use HTML and CSS we’ve learned in lab.** You should be able to complete this homework using HTML and CSS we’ve learned in lab. While you are not strictly forbidden from using HTML/CSS we haven’t talked about in lab, if you use it incorrectly or if you make poor stylistic choices, you may be marked down.

**Do not use JavaScript.** There’s no need to use JavaScript in this assignment.

**Respect Separation of Concerns.** Your HTML should describe the content of your page, and your CSS should describe your page’s appearance.

**Do not use an ordered list element (ol) to create the list layout.** There’s nothing wrong with doing this, per se, but it involves some fairly esoteric CSS that we did not cover in class. We recommend you write out the numbers “1.”, “2.”, etc. in the HTML instead of using ol.

**Follow the best practices** that were described in labs.
**Reduce redundancy in CSS.** Try not to have too many redundant styles, if there are ways to use inheritance or special selectors to concisely define style rules.

**Match the given appearance as closely as possible.** It’s fine if your web page is off from our description by a few pixels here and there, but noticeable differences may result in a lower homework score.

**Use descriptive names, consistent spacing,** and otherwise follow good code hygiene.

### III. Part 2: Your Listicle: my-list.html

Design your own listicle for a topic of your interest! You should write the content in `my-list.html` and the style in `my-list-style.css` which can be found in as3-supplementary.zip.

For full credit, please abide by the following:

- Cannot be an exact line-for-line copy of `listicle.html`
- Must have at least new 3 items in your list
- You can, but do not have to, refer to `listicle-style.css`.

You can spend as much or as little time on this part as you want, provided that your page does fall within the guidelines above.

**Ideas**

- **CSS animations:** When the real [Squarespace demo](https://www.squarespace.com/) loads, notice how the text animates in. We did not talk about how to do that in lab but you can try experimenting with some CSS animations on your page. You can check out this [W3Schools demo](https://www.w3schools.com/css/) to see how it works. Inspect the elements to see what CSS they use to get this effect.
- **“Parallax” Effect:** This is a trendy style where the background of a page is moved at a different speed than the foreground content while scrolling. In its simplest form, the background image stays in the same place as the page scrolls (as we saw in lab and on Part 1). Try implementing the parallax effect using CSS! You can check out this [W3Schools demo](https://www.w3schools.com/css/) to see how it works. Inspect the elements to see what CSS they use to get this effect.
  - Note: The example linked above is simple but not actually best practice. You can check out this [article](https://www.w3schools.com/css/) to try a more sophisticated implementation.

### IV. Submission

Create a .zip file that will contain all the folders and files consisting Part1 and Part2 and submit it to Moodle. The name of the zip file should be `username-HW3.zip`. 