

First Project: Advanced HTML and CSS

Sprint 1



Introduction

Here's the brief for your first project at Practicum. If it appears larger than the previous tasks, that's because it is! It generally takes up to 20 hours to complete, but you've already practiced the individual concepts used in the project via the tasks. So, now it's time to put them all together.

To help, we've divided the project into several parts (see the table of contents).

! Pro tip:

Try to finish a certain portion of the project every day and submit it before the soft deadline. That way it won't distract you once you start working on the next sprint.

Additionally, remember to take regular breaks. Some students prefer to take a 5-minute break every 25 minutes, whereas others like a 10-minute break every hour. Everyone will work by a different schedule, but it's been proven that breaks help boost productivity and creativity. Find a rhythm that works and try to stick with it.

It's also perfectly normal to forget things while you're still learning. If you don't think you remember something, try crafting a specific search engine query. For instance: "how to specify a CSS font family".

You can always ask your tutor or fellow students for help on Slack.

Table of contents

Assignment overview	02
1. General styles and semantics	03
Task 1. Adding a page wrapper	
Task 2. Semantics	
Task 3. Specifying sizes	
Task 4. Font settings	
2. Header	06
Task 5. Styling the menu using flexbox	
Task 6. Anchor links	
Task 7. Using absolute positioning	
Task 8. Implementing hover states	
3. Section: Staff picks	10
Task 9. Creating a flexbox container	
*Extra task. Adding your books	
4. Section: Events	12
Task 10. Adding one more hover state	
Task 11. Making complex compositions with z-index	
5. Section: Membership	15
Task 12. More flexbox practice!	
6. Section: About the library	16
Task 13. Arranging text	
Task 14. Adding an image below text content	
7. Footer	18
Task 15. HTML & CSS markup	
8. Self-review	20

Assignment overview

In **pic. 1**, you'll see how your webpage should look after finishing the 15 tasks described in the brief. Take a moment to study how the webpage provided ([index.html](#)) differs from the design.

The tasks are organized as you'd approach any regular HTML/CSS project. We'll first start with some common styles and settings, then we'll go over the design from top to bottom, refining the webpage to look exactly like it does in the design.

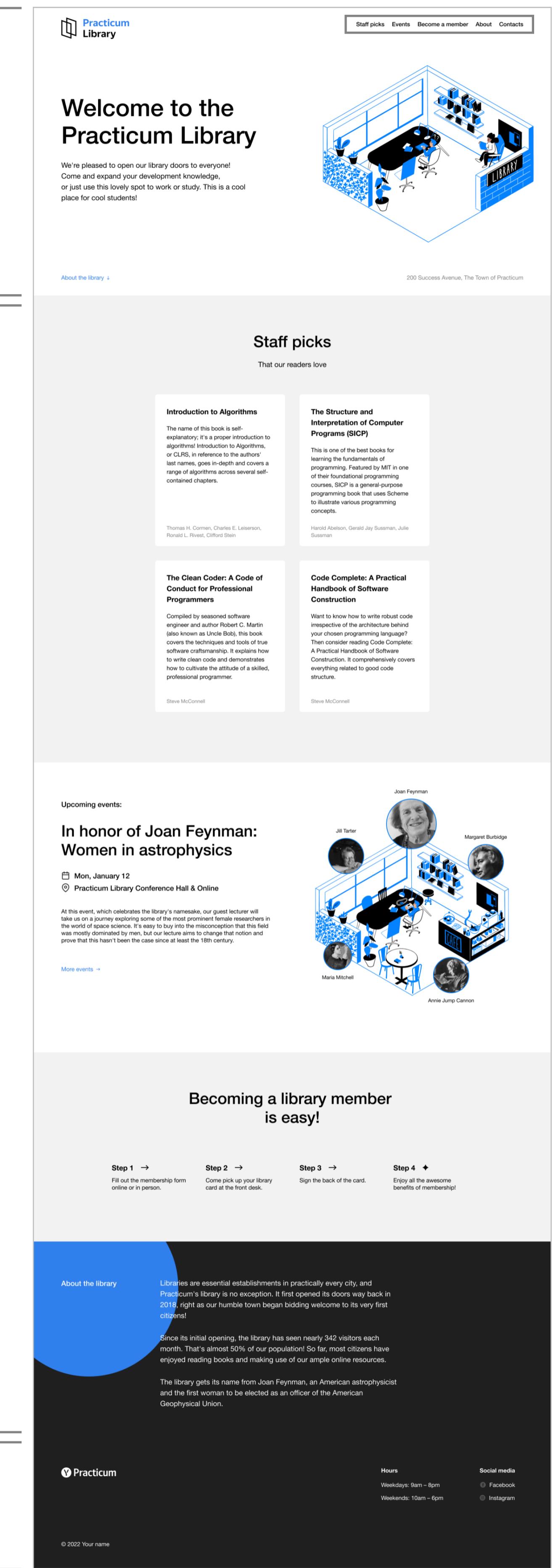
Colors:

- **#ffffff** (main background, contrast font color)
- **#000000** (main font color)
- #f2f2f2** (contrast background)
- #1f1f1f** (footer background)
- **#2f80ed** (links, accents)
- **#1f69c9** (link hover)
- #c1c1c1** (link hover in footer)

Site header:
header block

Main page content:
content block

Site footer:
footer block



Navigation bar:
nav block

Section:
staff block

Section:
events block

Section:
membership block

Section:
about block

1. General styles and semantics

! Pro tip:

Before you start working on the code, launch a live server for [index.html](#) so you can see your changes as you make them. Remember to save your files to have the changes rendered.

Task 1. Adding a page wrapper

Let's start by wrapping the entire page content in a new `<div>` element with the [page](#) class.

Next, go to the styles file ([style.css](#)) and add the new class [page](#) at the very top of the file, since it's the top-most parent class of the webpage. Specify styles like so:

- The block should have a minimum width of [1100px](#) and a maximum width of [1600px](#).
- The block should be centered.

! Pro tip:

After implementing a wrapper like this, you can easily control your content width. This is useful on large screens because your content won't be stretched too wide and become unreadable. It's also helpful for small screens as it will prevent certain issues, such as content from overlapping.

Task 2. Semantics

Carefully review the sections above. Then, replace the `<div>` elements in the HTML code accordingly with the appropriate semantic tags. Remember to update both the opening and closing tags.

Task 3. Specifying sizes

This task is useful practice for locating relevant styles for an HTML element. It will also help you gain familiarity with the code provided.

Find HTML elements 1-7 shown in **pic. 2 – 5**, check which CSS classes are assigned to them, and then find the style in [styles/style.css](#). Using CSS, specify sizes for the respective elements.

Find the [header__image](#) (Element 4) and uncomment it. We had previously hidden this part.

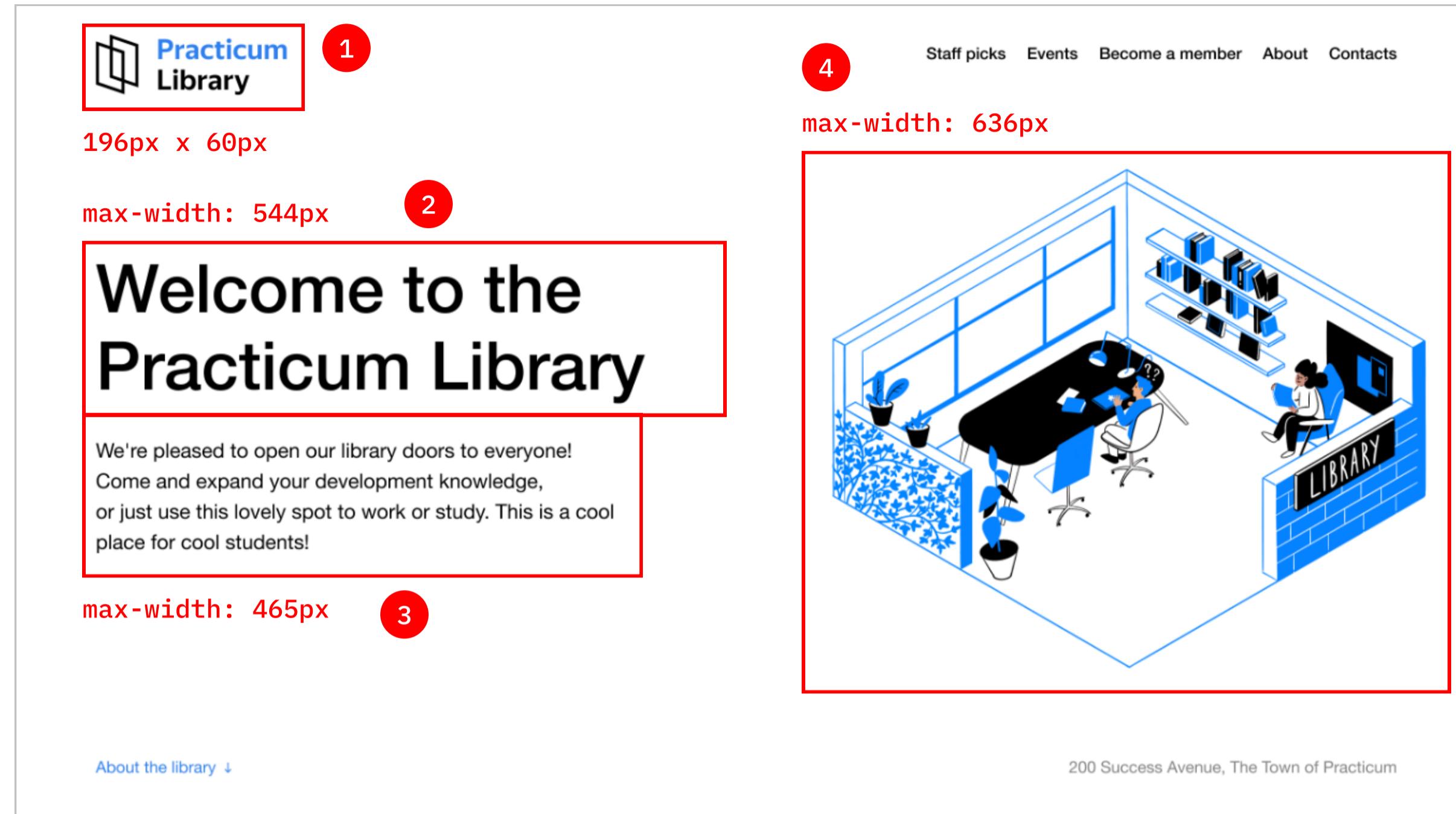
Mind how your page's appearance changes in the browser and pay attention to how parent settings affect the appearance of child elements and those that share the same class.

! Pro tip:

Use the [max-width](#) property for your elements to prevent them from exceeding the size of their parent window (or the browser).

This also applies to images, but there is no need to specify [max-width](#) for images with a width or height less than [320px](#) since 320 pixels is the smallest screen size in active use (iPhone 5).

Finally, there is no need to specify image heights, seeing as modern browsers calculate them automatically.



Pic. 2

Staff picks

That our readers love

Introduction to Algorithms

The name of this book is self-explanatory; it's a proper introduction to algorithms! Introduction to Algorithms, or CLRS, in reference to the authors' last names, goes in-depth and covers a range of algorithms across several self-contained chapters.

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein

The Structure and Interpretation of Computer Programs (SICP)

This is one of the best books for learning the fundamentals of programming. Featured by MIT in one of their foundational programming courses, SICP is a general-purpose programming book that uses Scheme to illustrate various programming concepts.

Harold Abelson, Gerald Jay Sussman, Julie Sussman

The Clean Coder: A Code of Conduct for Professional Programmers

Compiled by seasoned software engineer and author Robert C. Martin (also known as Uncle Bob), this book covers the techniques and tools of true software craftsmanship. It explains how to write clean code and demonstrates how to cultivate the attitude of a skilled, professional programmer.

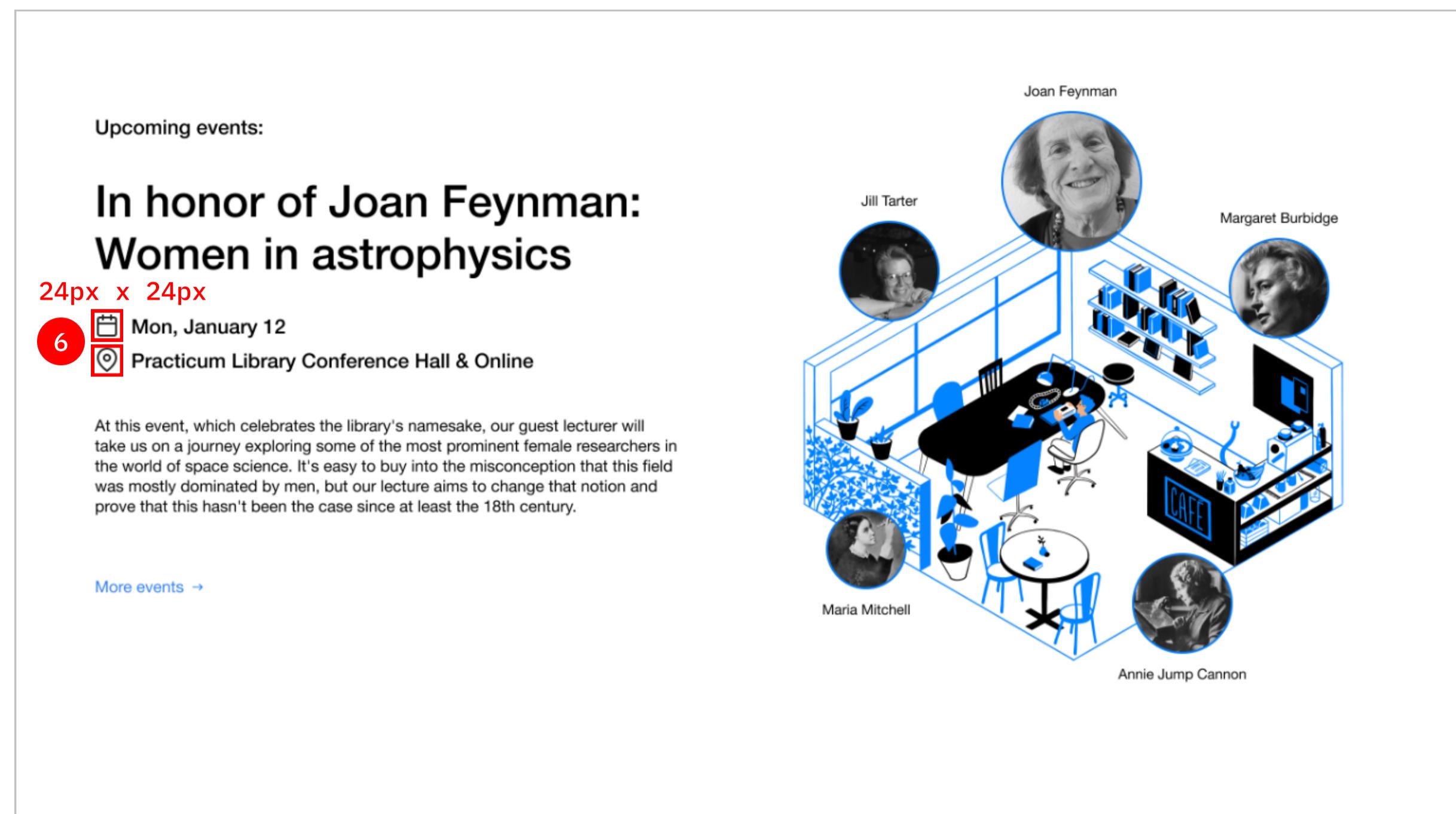
Steve McConnell

Code Complete: A Practical Handbook of Software Construction

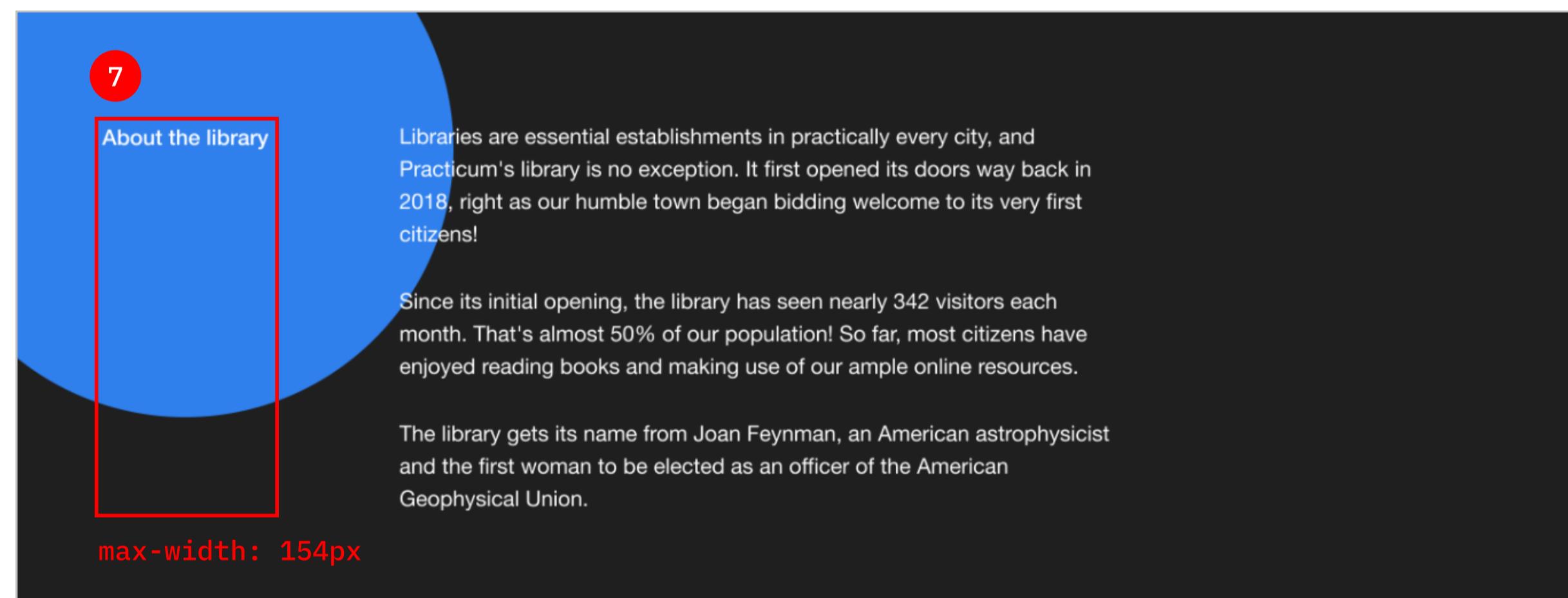
Want to know how to write robust code irrespective of the architecture behind your chosen programming language? Then consider reading Code Complete: A Practical Handbook of Software Construction. It comprehensively covers everything related to good code structure.

Steve McConnell

Pic. 3



Pic. 4



Pic. 5

Task 4. Font settings

Write the font settings for the [page](#) block:

- **Font family:** Specify "[Helvetica Neue](#)" as the main font with a fallback of [Arial](#), and use the generic [sans-serif](#) family in case a user doesn't have either of the other fonts.

Note: Helvetica Neue is not available on Windows by default, so the font may fall back to [Arial](#) or another [sans-serif](#) font — this is normal.

- **Font size:** Set this to [16px](#).
- **Line height:** Set this to [20px](#).

! Pro tip:

While it makes sense to specify the font family for the entire page (since we're only using one), we will use different font sizes and line heights for different text elements. However, it's still more efficient to specify some general settings for the parent class and then override the parent style. For this to work, make sure these classes are placed after the parent class in the CSS file.

The other option is to specify all styles for all text element classes separately but it's not what we recommend.

2. Header

The screenshot shows the header and a portion of the main content area of the Practicum Library website. The header features a logo 'Practicum Library' with a stylized book icon. A red border highlights the navigation bar which includes links for 'Staff picks', 'Events', 'Become a member', 'About', and 'Contacts'. Below the header, the text 'nav block' is labeled. The main content area has a large title 'Welcome to the Practicum Library' and a paragraph of text: 'We're pleased to open our library doors to everyone! Come and expand your development knowledge, or just use this lovely spot to work or study. This is a cool place for cool students!'. To the right of the text is a 3D-style illustration of a library interior. The illustration shows a person sitting at a desk working on a computer, another person reading a book in a chair, and bookshelves filled with books. The word 'LIBRARY' is written on a brick wall. The bottom left of the content area has a link 'About the library ↓' and the bottom right has the address '200 Success Avenue, The Town of Practicum'. The entire content area is enclosed in a light gray box.

Task 5. Styling the menu using flexbox

To ensure that the menu fully conforms to the design, modify the `nav` and `nav-links` blocks.

For the `nav` block:

- It should be a flexbox container.
- Distribute the flexbox items using the `justify-content` property.
- Center the flexbox items using the `align-items` property.

Then, modify the `nav-links` block as follows:

- It should be a flexbox container too.
- Center the flexbox items using the `align-items` property.
- Specify the `column-gap` equal to `20px`.

The screenshot shows a website for the "Practicum Library". The header features a logo with two overlapping white rectangles and the text "Practicum Library". Below the header is a navigation bar with links: "Staff picks", "Events", "Become a member", "About", and "Contacts". A red box highlights the text "nav__link elements" above the navigation bar. The main content area has a large title "Welcome to the Practicum Library". Below the title is a paragraph: "We're pleased to open our library doors to everyone! Come and expand your development knowledge, or just use this lovely spot to work or study. This is a cool place for cool students!". To the right of the text is a 3D-style illustration of a library interior. The illustration shows a person sitting at a desk working on a computer, another person reading a book, and bookshelves filled with books. The word "LIBRARY" is written on a brick wall. A red box highlights the text "header__link element" above the "About the library" link. At the bottom right of the main content area is the address "200 Success Avenue, The Town of Practicum".

header__link element

[About the library](#)

Staff picks Events Become a member About Contacts
nav__link elements

Welcome to the Practicum Library

We're pleased to open our library doors to everyone!
Come and expand your development knowledge,
or just use this lovely spot to work or study. This is a cool
place for cool students!

LIBRARY

200 Success Avenue, The Town of Practicum

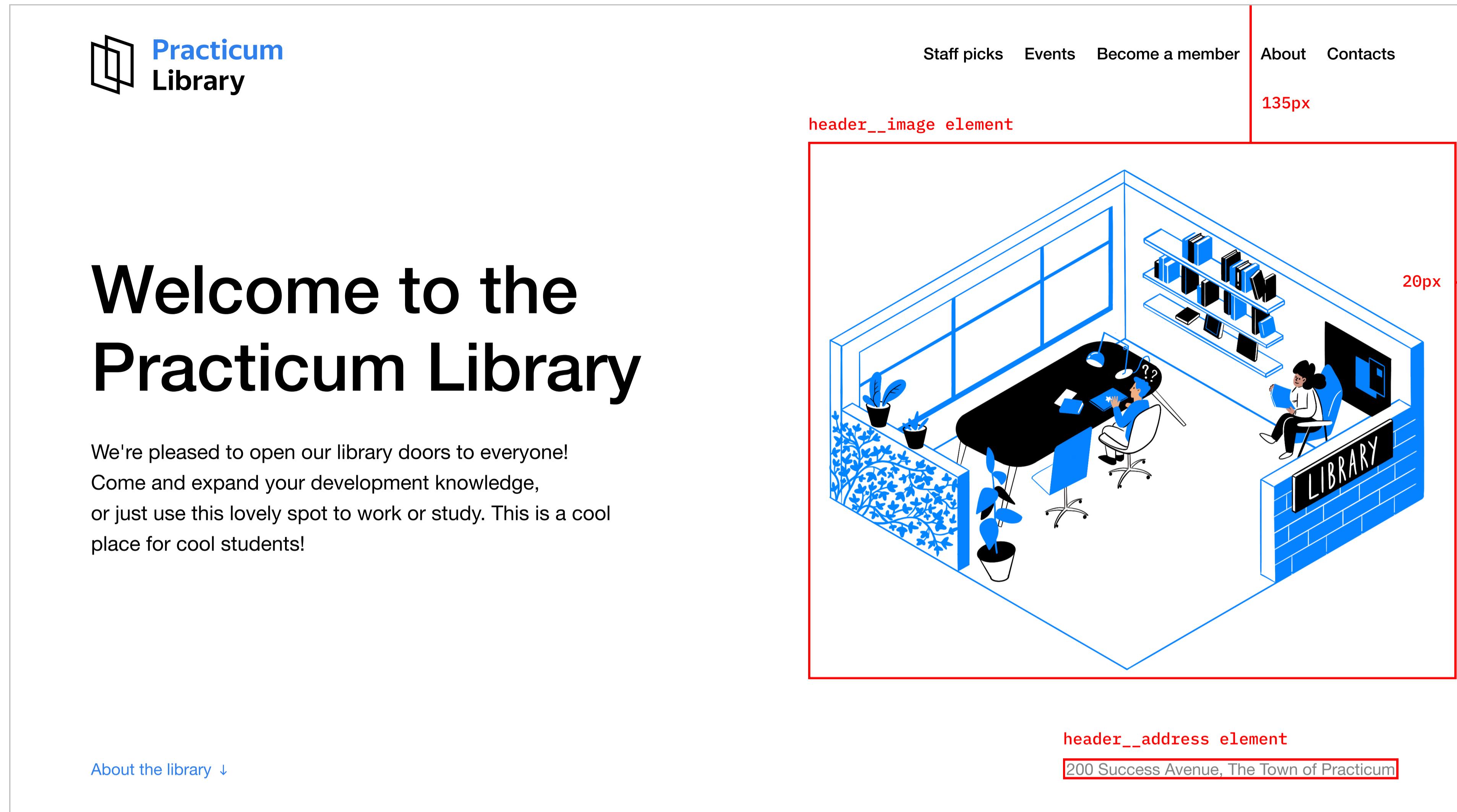
Task 6. Anchor links

Now, let's make our menu functional. The links should lead the user to their respective sections of the page. Complete the steps as follows:

- Assign `id` to the corresponding target sections: Staff picks, Events, Become a member, About, and Footer. The name of the `id` can be a custom selection, but it's good practice to keep it descriptive (in this case, we suggest making it identical to the section name). Use hyphens - as separators instead of space characters and use lowercase letters.

- Add the `href` attribute for the `nav__link` elements and use their corresponding `id` values. Remember to use `#` for links to the same page.

Also, add one more anchor link for the "About the library" text (the `header__link` element). It should lead to the corresponding section.



Task 7. Using absolute positioning

The header still doesn't look quite like the design. In the CSS file, use the `position` property with the value of `absolute` to place the image (`header__image`) as shown in the design. Use the correct values of the `top/right/bottom/left` properties to place it as necessary.

The screenshot shows the header of the Practicum Library website. At the top left is the logo 'Practicum Library' with a book icon. At the top right is a navigation bar with links: 'Staff picks', 'Events', 'Become a member', 'About', and 'Contacts'. Below the navigation bar, the text 'nav__link elements' is written in red. The main content area features a large black title 'Welcome to the Practicum Library'. Below the title is a paragraph of text: 'We're pleased to open our library doors to everyone! Come and expand your development knowledge, or just use this lovely spot to work or study. This is a cool place for cool students!' To the right of the text is a 3D-style illustration of a library interior. The illustration shows a person sitting at a desk working on a computer, another person sitting in a chair reading a book, and bookshelves filled with books. The word 'LIBRARY' is visible on a brick wall. At the bottom left of the content area, the text 'header__link element' is written in red, followed by a link 'About the library ↓' in a red-bordered box. At the bottom right, the address '200 Success Avenue, The Town of Practicum' is shown.

Task 8. Implementing hover states

In the header, we have two types of links: `nav__link` in the navigation bar and `header__link`. According to the design, their colors should change when the user hovers over them with their cursor. To this end, do the following:

- Find the `nav__link` class. Add a new `nav__link` class with the `:hover` pseudo-class directly below these rules. Inside this class, set the text color to `#1f69c9`.
- Do the same for `header__link`.

3. Section: Staff picks

Staff picks

That our readers love

Introduction to Algorithms

The name of this book is self-explanatory; it's a proper introduction to algorithms! Introduction to Algorithms, or CLRS, in reference to the authors' last names, goes in-depth and covers a range of algorithms across several self-contained chapters.

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein

The Structure and Interpretation of Computer Programs (SICP)

This is one of the best books for learning the fundamentals of programming. Featured by MIT in one of their foundational programming courses, SICP is a general-purpose programming book that uses Scheme to illustrate various programming concepts.

Harold Abelson, Gerald Jay Sussman, Julie Sussman

The Clean Coder: A Code of Conduct for Professional Programmers

Compiled by seasoned software engineer and author Robert C. Martin (also known as Uncle Bob), this book covers the techniques and tools of true software craftsmanship. It explains how to write clean code and demonstrates how to cultivate the attitude of a skilled, professional programmer.

Steve McConnell

Code Complete: A Practical Handbook of Software Construction

Want to know how to write robust code irrespective of the architecture behind your chosen programming language? Then consider reading Code Complete: A Practical Handbook of Software Construction. It comprehensively covers everything related to good code structure.

Steve McConnell

wrap all the cards

68px

max-width: 760px

40px

See the task on the next page →

Task 9. Creating a flexbox container

The cards are currently being rendered one after another, which is the normal behavior of `<div>` elements with the default `display` value. To arrange the cards as specified in the design, complete the following steps:

- Wrap all the cards in an appropriate list tag. Name it according to BEM: it should be an element of the `staff` block.
- In the CSS file, before the `card` rule, create a rule for your new class.
- Assign the `max-width` and `margin` properties. The block should be centered.
- Make the new element a flexbox container.
- Use the flexbox properties `flex-wrap`, `justify-content`, and `gap` to arrange cards exactly as they are shown in the design.

*Extra task. Adding your books

If you have the time and desire to do so, we suggest changing “Staff picks” to “Reader picks” and adding info based on your favorite books! If you have too much text and it begins overflowing, take this as a bonus challenge. A good search engine query to solve this would be “dealing with text overflow CSS”.

This will give your project a more personal feel and you’ll be able to display a bit more of your personality in your portfolio — your future employer and colleagues are likely to appreciate that!

4. Section: Events

Upcoming events:

In honor of Joan Feynman: Women in astrophysics

📅 Mon, January 12
📍 Practicum Library Conference Hall & Online

At this event, which celebrates the library's namesake, our guest lecturer will take us on a journey exploring some of the most prominent female researchers in the world of space science. It's easy to buy into the misconception that this field was mostly dominated by men, but our lecture aims to change that notion and prove that this hasn't been the case since at least the 18th century.

[More events →](#)

Task 10. Adding one more hover state

Here's a simple task: add another hover state for the link that reads "More events".

Task 11. Using positioning to make complex compositions

Uh-oh, our events section isn't finished yet: it's still missing the illustration from the design. Of course, it would be possible to insert it all in one picture, but let's practice positioning and use it to implement a complex composition.

Let's work on it step-by-step:

1. In [index.html](#), find where the `events__content` element ends. Right after it, add a new `events__cover` element with a `<div>`. We have the rule prepared in the CSS; uncomment it. You'll be creating all the elements inside it.
2. Add the `cafe.png` image that's already in the `/images` folder. Add an alt text and assign it the `events__image` class. Find this class in [style.css](#). Uncomment it and see how the appearance changes.

Upcoming events:

In honor of Joan Feynman: Women in astrophysics

Mon, January 12
Practicum Library Conference Hall & Online

At this event, which celebrates the library's namesake, our guest lecturer will take us on a journey exploring some of the most prominent female researchers in the world of space science. It's easy to buy into the misconception that this field was mostly dominated by men, but our lecture aims to change that notion and prove that this hasn't been the case since at least the 18th century.

[More events →](#)

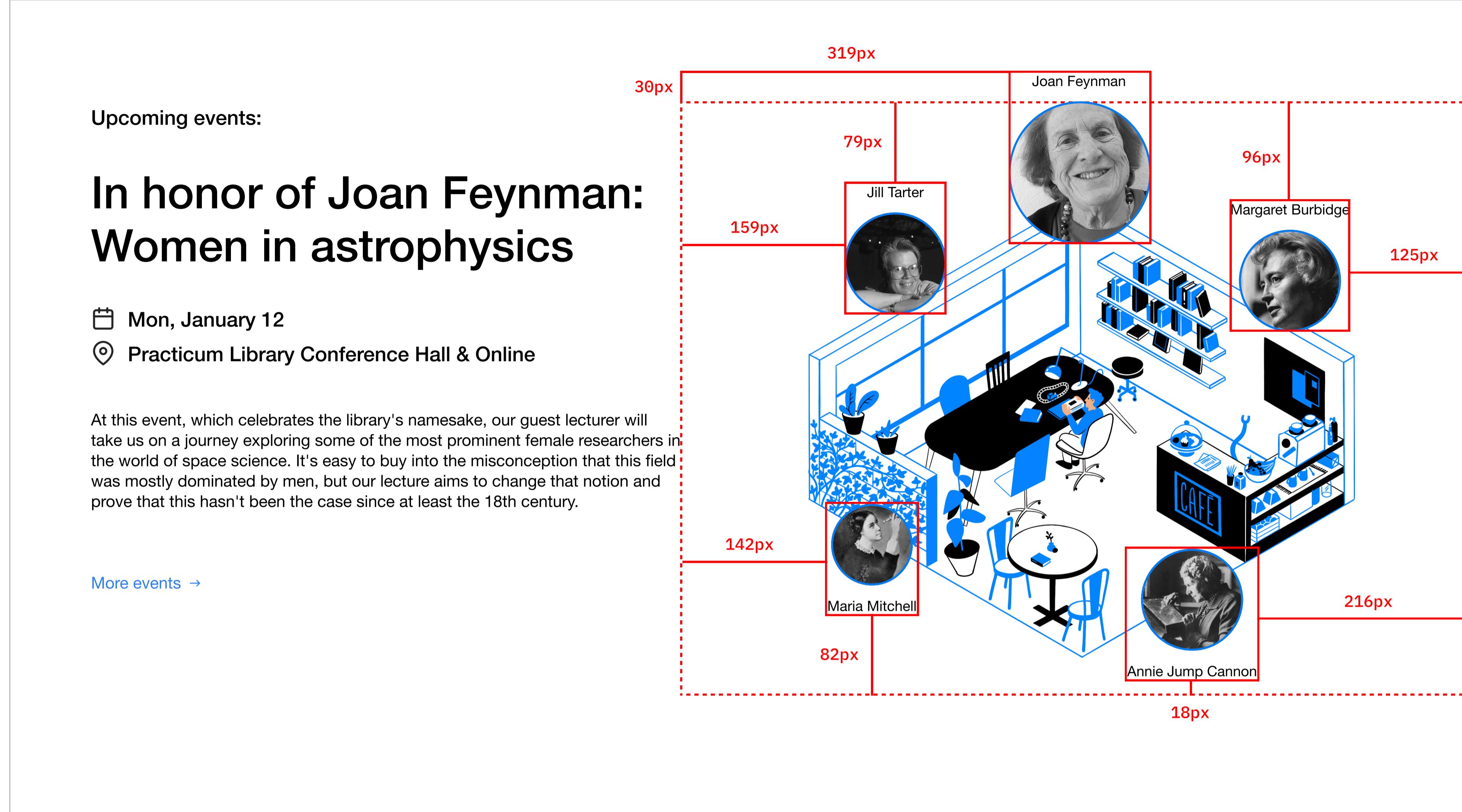
3. person element + person_name_feynman modifier
4. person__caption element
5. person__image element person_image_size_l modifier

Joan Feynman
Jill Tarter
Margaret Burbidge
Maria Mitchell
Annie Jump Cannon

3. Next, we'll work on the wrapper for a person's photo and inscription. Let's start with Joan Feynman. Right after the cafe image in HTML, create a `<div>` with two classes: `person` and `person_name_feynman`. In the CSS file, uncomment the `person` class and see what happens.

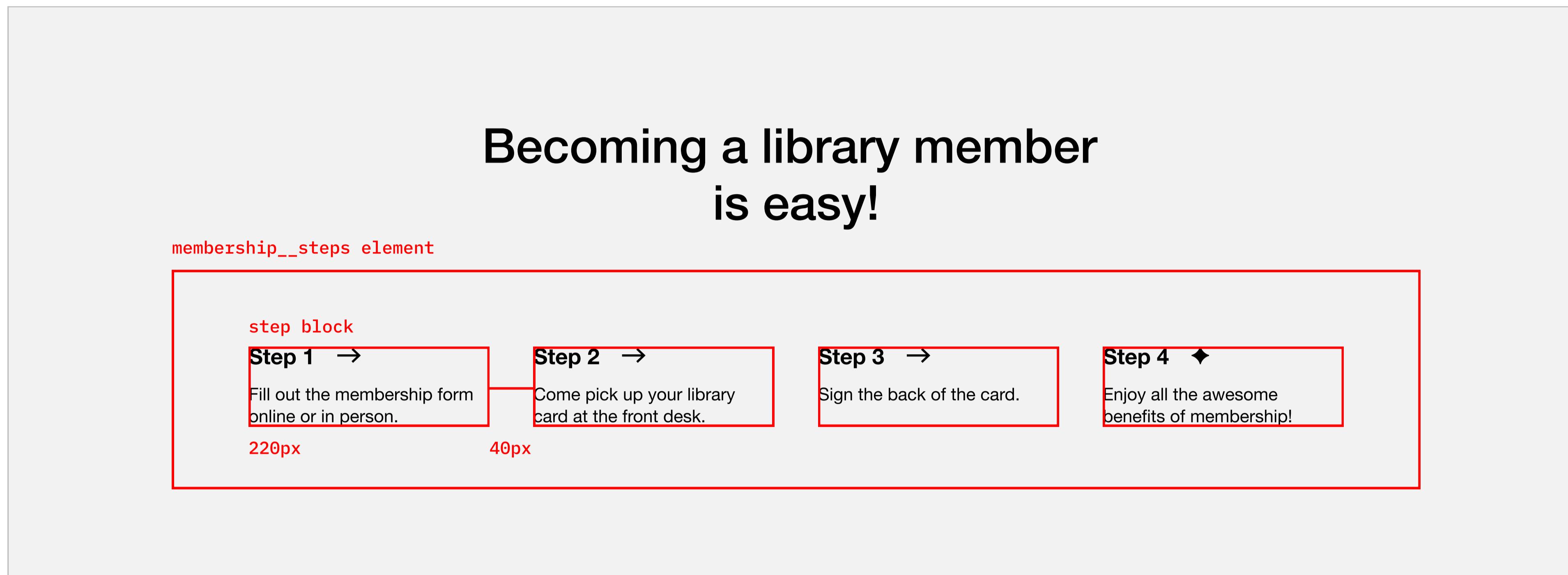
4. Inside this new block, add a paragraph saying "Joan Feynman" and assign it the `person__caption` class. Uncomment the corresponding styles.

5. After the paragraph, add Feynman's photo. It's already in the `/images` folder. Add an alt text and assign it two classes: `person__image` and the modifier `person__image_size_l`. Uncomment the styles of these classes and try to understand why we need three different classes. Next, uncomment the styles of the other two modifiers: `person__image_size_s` and `person__image_size_m`. Likewise, think about why we need 3 modifiers.



6. Use correct values of the `top/right/bottom/left` properties in the `person_name_feynman` class to place the block according to the design.
7. Repeat steps 3-6 using the appropriate classes and modifiers for the other people: Margaret Burbidge, Annie Jump Cannon, Maria Mitchell, and Jill Tarter. Notice that the paragraph is sometimes placed before the image but comes after the image at other times, and add the HTML elements accordingly. All the styles have been prepared for you, except those starting with `person_name_`. Use them to position all the badges according to the design.

5. Section: Membership

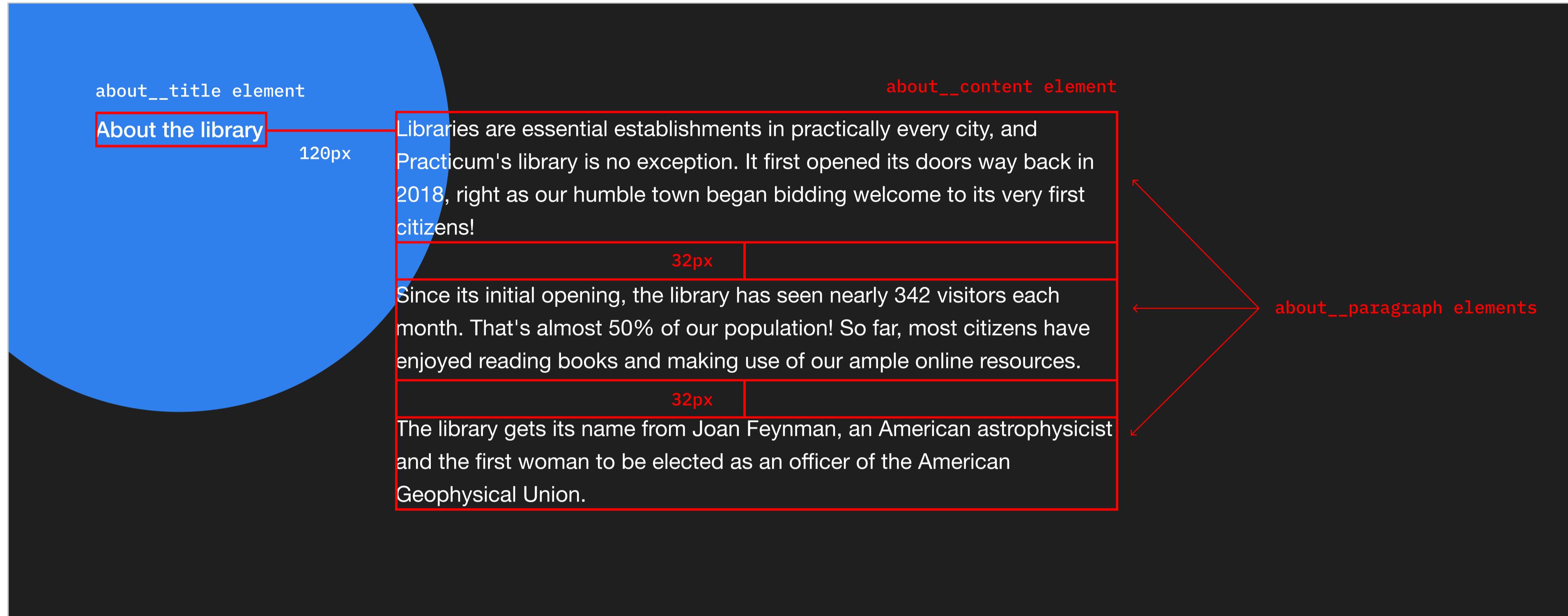


Task 12. More flexbox practice!

Update the CSS of the `membership__steps` and `step` classes to arrange the cards as shown in the design. Make sure the gap is correct. Notice that the cards are centered inside the `membership__steps` element.

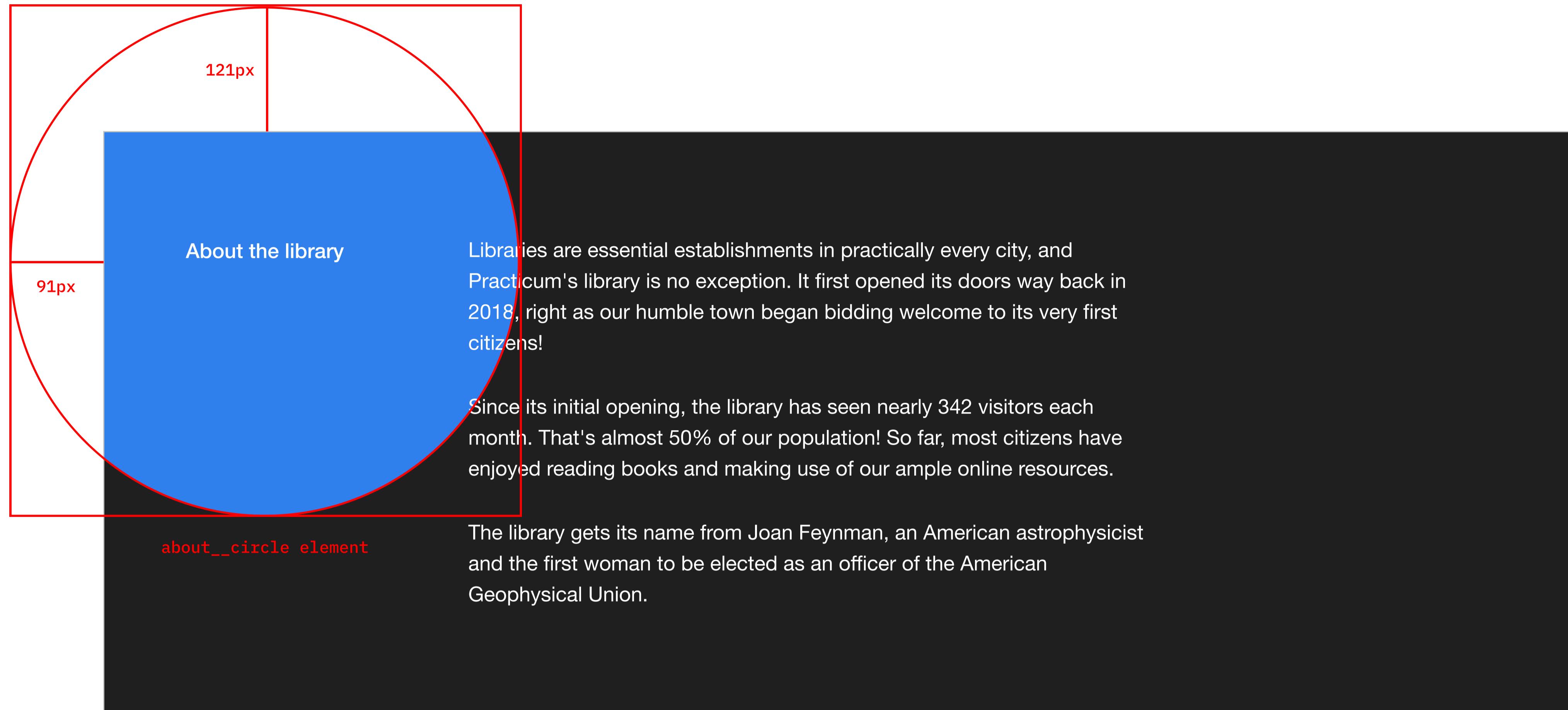
Use `flex-basis` to set the initial main size of the flex items.

6. Section: About the library



Task 13. Arranging text

Arrange the text as shown in the design by adding the necessary styles to `about`, `about__title`, `about__content`, and `about__paragraph`.



Task 14. Adding an image below text content

Notice the blue accent circle placed under the texts. In the HTML, uncomment the `<div class="about__circle"></div>` element. Finish the styles in `about__circle` to position the circle as shown in the design and assign `z-index` values to elements of the sections in order to arrange them correctly.

7. Footer



Task 15. HTML & CSS markup

And so we've come to the last section of the webpage. Your task is to mark up the footer from scratch based on the details shown in the image above. First, we'll mark up everything in HTML, and then we'll add the CSS styles. The appropriate selectors are already in `style.css`. Here's a step-by-step framework to guide you:

1. The logo, hours info, and social media form 3 columns. Let's keep them in one wrapper with the class `footer__columns`.
2. Inside the wrapper, create three `<div>` elements, each with the class `footer__column`.
3. Inside the first column, add a logo with the class `footer__logo`.
4. Assign the `footer__column_content_hours` class to the second column. Create a heading of the appropriate level that reads "Hours" and has the class `footer__column-heading`. The info below the column header should be a list with the class `footer__list`. Each line should be a list item with the class `footer__list-item`.

5. Assign the `footer__column_social` class to the third column. Create a heading of the appropriate level that reads "Social media" (notice that it's the same level as "Hours") with the class `footer__column-heading`. The info below the column should be a list with the class `footer__list`. Each line should be a list item with the class `footer__list-item`. Each list item should contain a link with the class `footer__column-link`. Add the text and icons inside the links so that they are both clickable, and assign the class `footer__social-icon` to icons. The links should lead to `#`.
6. Add an appropriate text element for the copyright. It should say: "© 2022 [Your name]". Assign it the class `footer__copyright`.



Now, let's switch to the styles. You'll need to do the following, and you can write styles in the order that's most convenient for you:

1. For the footer: set the background, text color, and padding according to the design.
2. Use the classes `footer__columns`, `footer__column_content_hours`, `footer__column_content_social` to arrange the blocks according to the design.
3. Specify the size for the logo and icons: `footer__logo` and `footer__social-icon`.
4. Arrange the content inside the second and third columns according to the design by setting the correct space between elements. Find a way to ensure there is no space after the last item.
5. In the third column, arrange the `footer__column-link` links so that their text and icons are arranged according to the design.
6. Implement the proper hover state for `footer__column-link`. The text color should change to `#c1c1c1`.
7. Specify the margins for `footer__copyright` based on the design. Make sure the copyright text is placed at the bottom of the footer.

8. Self-review

You are nearing the project finish line, and you've already completed the hardest part. Still, there is one last crucial task before you complete your project: conducting the self-review. To help, we've prepared a checklist:

- Compare each section of your webpage against the brief (open your project side-by-side with the brief and go through each point. Consider jotting down notes for yourself with things that you want to remember).
- Check the validity of your code using a [validator](#).
- Check your project against all the checklist items (you'll find this in the project description on the platform).
- Make sure to remove any redundant comment lines. Keep the comments that help you better understand the project structure and facilitate your understanding of complex features.

And with that, you've finally finished all the tasks for your very first project at Practicum. Awesome job!

