

Second Project: Advanced HTML and CSS

Sprint 2

Final Stage

Introduction

Here's the brief for the final stage of your second project at TripleTen. In this stage, you'll finish the last three sections of your webpage: the menu, the coffee shop sections, and the footer. After that, you'll set up some CSS transition effects.

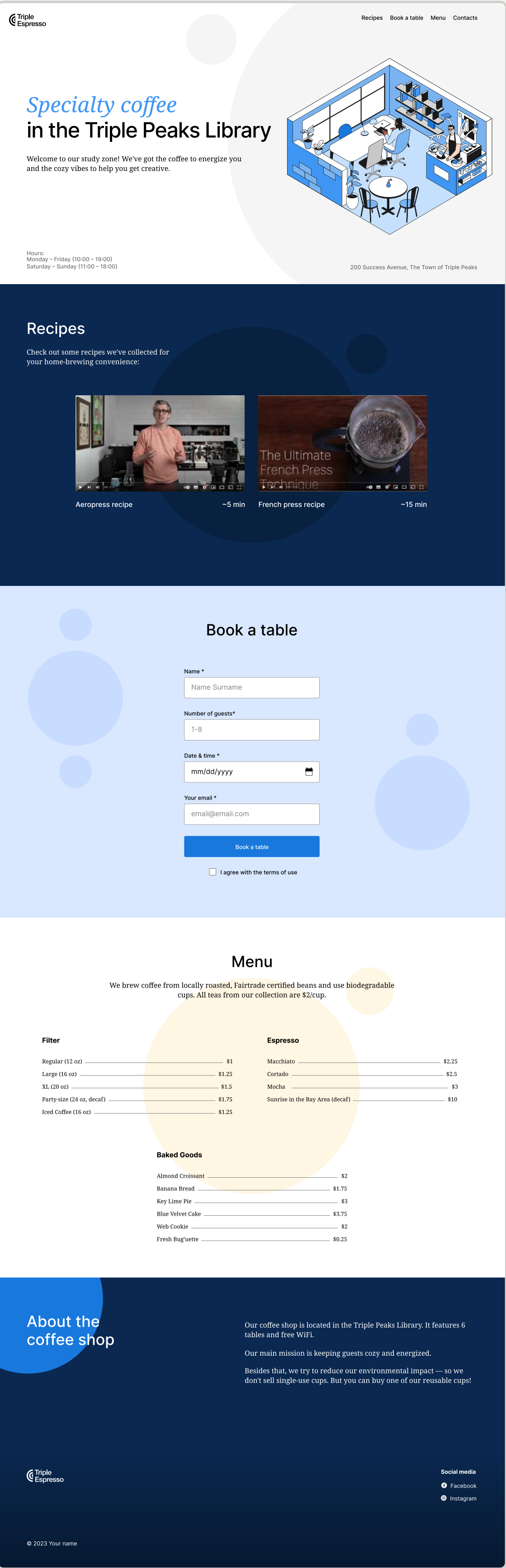
Work through the tasks to complete your webpage!

Table of contents

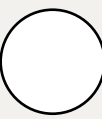
Assignment overview	02
1. Section: Menu	03
Required elements and class names	03
Sizes and spacings	04
Typography	05
2. Section: About the coffee shop	06
Required elements and class names	06
Sizes and spacings	07
Typography	08
Animation	09
3. Footer	10
Required elements and class names	10
Sizes and spacings	12
Typography	12
4. Smooth Transitions	13
5. Self-review	13


Assignment overview

Here is how your webpage should look by the end of the project.




Colors:


 #ffffff (header and menu background)


 #000000 (main font color)

 #3F96F3 (accent text)


 #1878DC (button, circle)

 #838383 (input borders and placeholders)

 #6F6F6F (hours and address)

 #0A2750 (recipes background)

 #D9E7FF (reservation background)

 #0A2850 (about background)

1.1. Section: Menu

1Menu

2We brew coffee from locally roasted, Fairtrade certified beans and use biodegradable cups. All teas from our collection are \$2/cup.

3

5Filter

7

6

Regular (12 oz) \$1

Large (16 oz) \$1.25

XL (20 oz) \$1.5

Party-size (24 oz, decaf) \$1.75

Iced Coffee (16 oz) \$1.25

4

Espresso

Macchiato \$2.25

Cortado \$2.5

Mocha \$3

Sunrise in the Bay Area (decaf) \$10

Baked Goods

Almond Croissant \$2

Banana Bread \$1.75

Key Lime Pie \$3

Blue Velvet Cake \$3.75

Web Cookie \$2

Fresh Bug'nette \$0.25

 Pro tip

The list divider is one of the trickiest parts of this design, so we'll help you out by showing you the suggested markup structure.

```
<li class="card__list-item">
  Name
  <span class="card__list-divider">
.</span>
  Price
</li>
```

Then, to style it as shown, you can turn the into a flex container. If you do this, there will be three flex items: the two pieces of text content and the tag.

You can use a flex item property to control how much of the available space the list divider element takes up and give it a bottom border to create the line. If you forget which property controls how a flex item grows, you can use our platform's search feature to find the relevant lesson. Or use the search engine of your choice to locate the relevant page on MDN.

Required elements and class names

Background: Set the background color to #fff. For the background image, provide the URL to background_menu.svg. The background doesn't repeat and its position is centered. Set the size value to cover.

1. block: menu, element: title.

2. block: menu, element: subtitle.

3. block: menu, element: cards. Notice how the third card is centered. It may look difficult at first, but this is where flexbox shows its true potential.

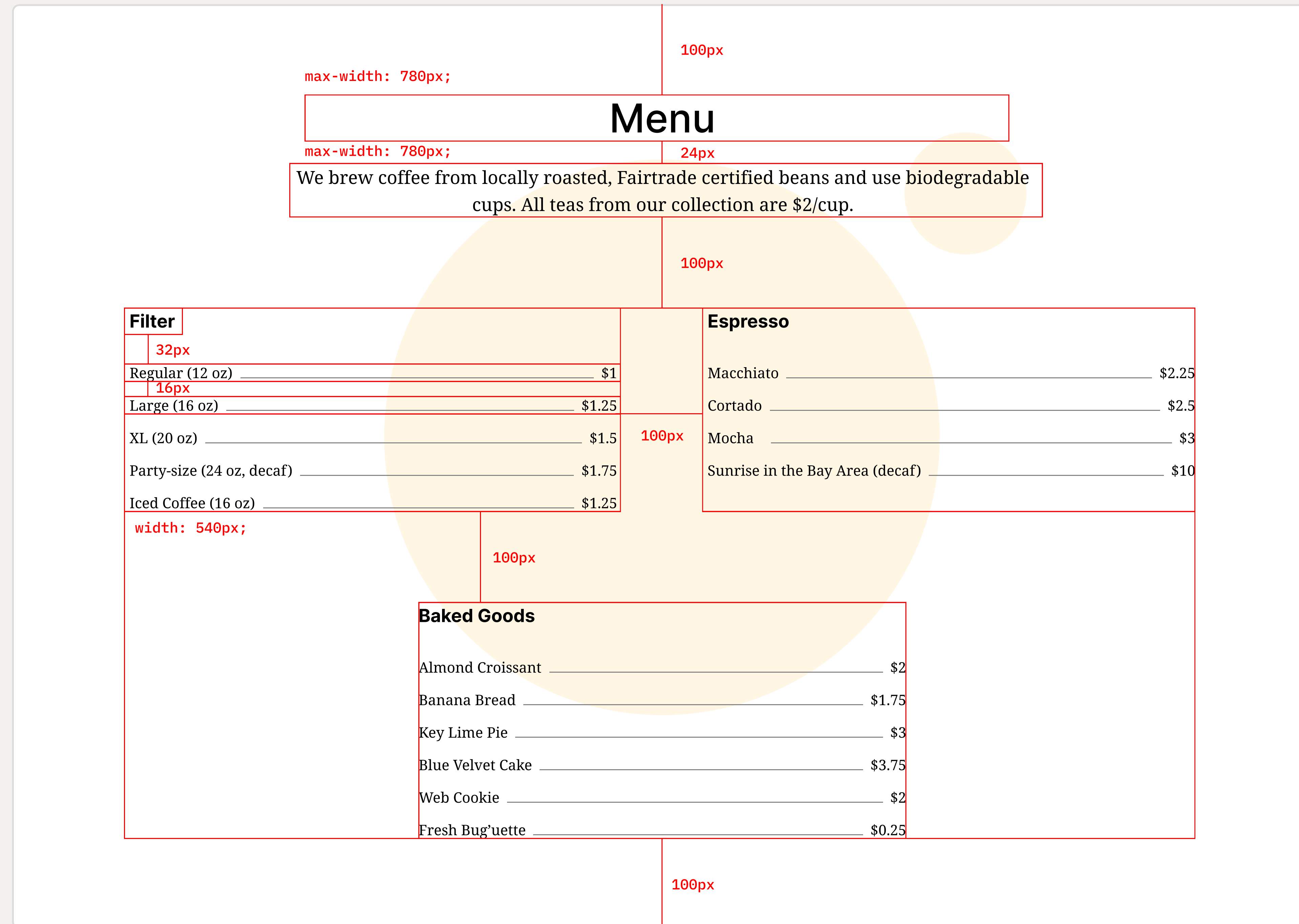
4. block: card.

5. block: card, element: title.

6. block: card, element: list-item.

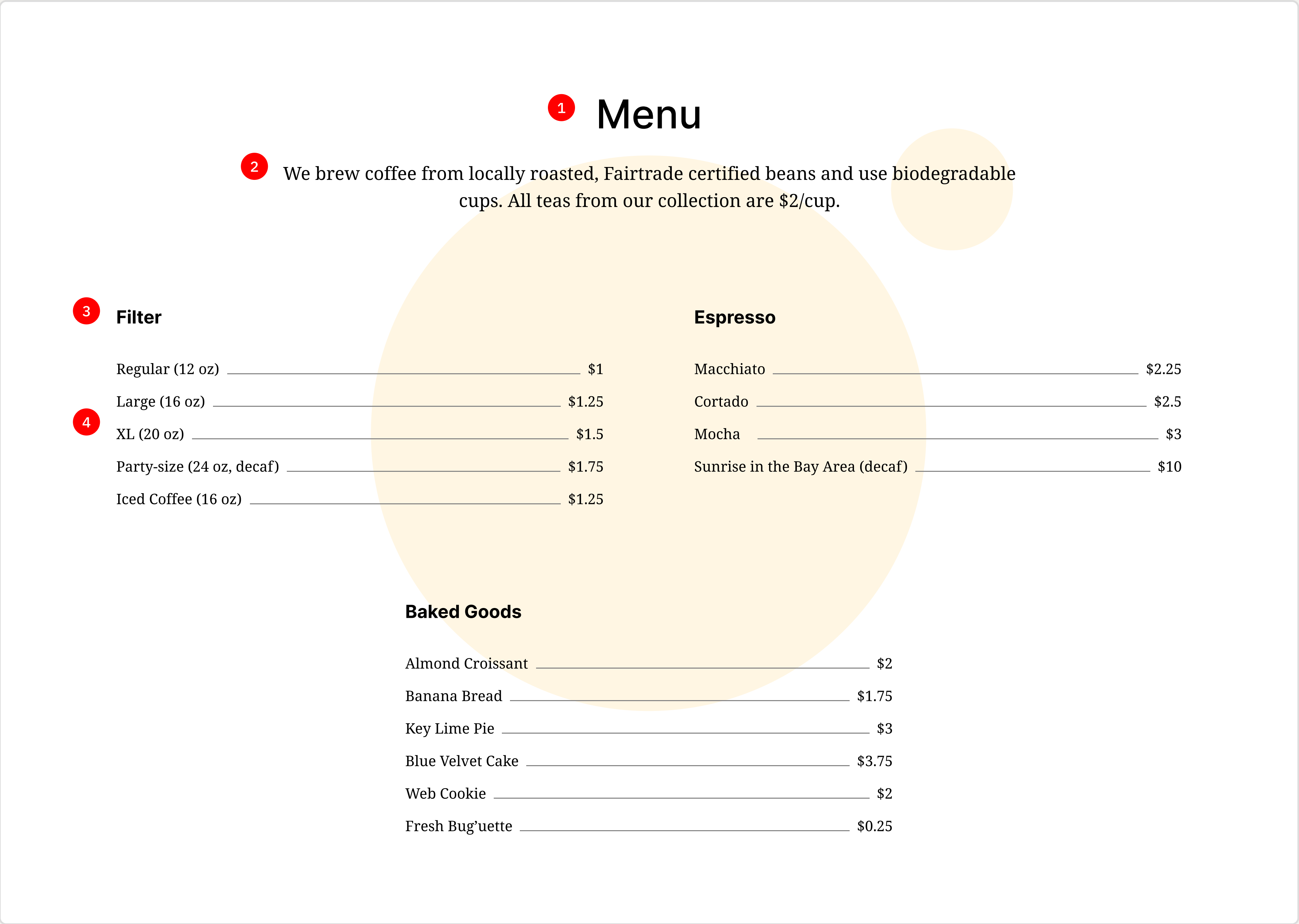
7. block: card, element: list-divider.

03



Sizes and spacing

Add all the sizes according to the picture above.
All blocks should be centered inside the section.



- 1

```
font-family: "Inter";
font-style: normal;
font-weight: 500;
font-size: 44px;
line-height: 52px;
```
- 2

```
font-family: "Noto Serif";
font-style: normal;
font-weight: normal;
font-size: 20px;
line-height: 30px;
```
- 3

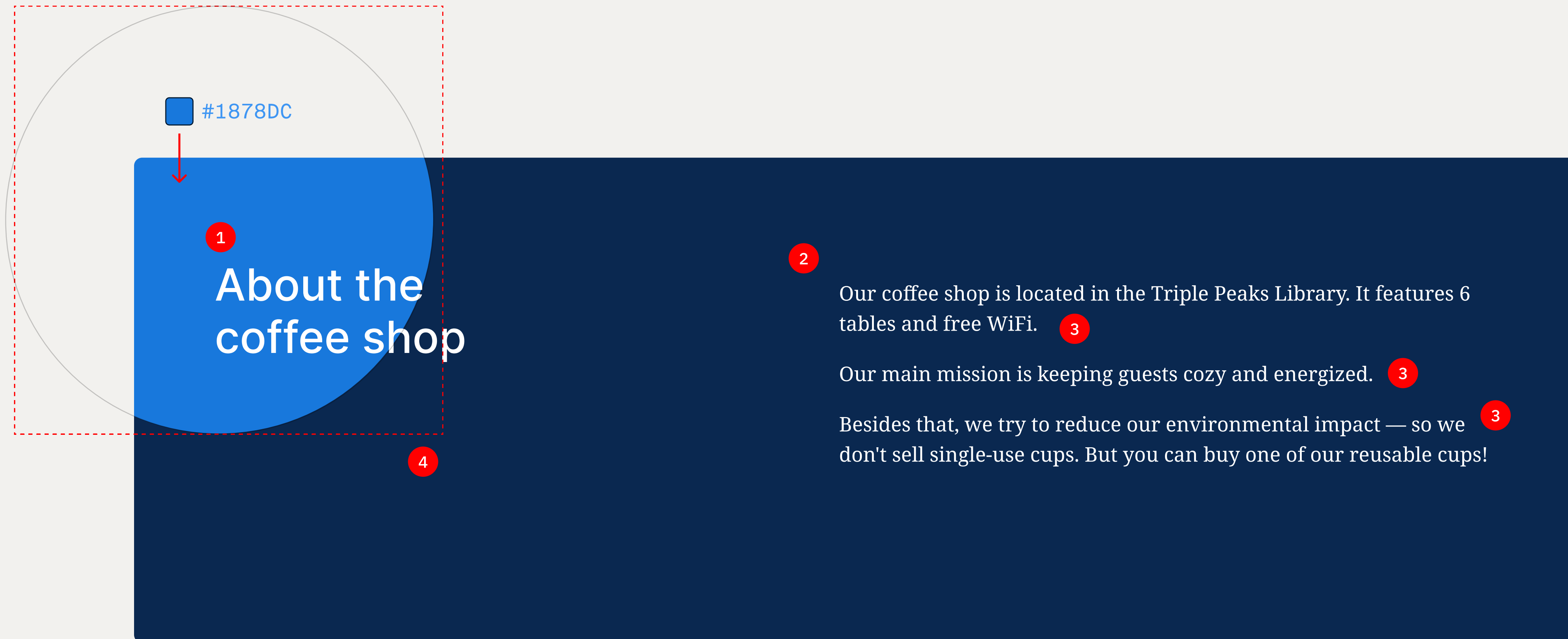
```
font-family: "Inter";
font-style: normal;
font-weight: bold;
font-size: 20px;
line-height: 30px;
```
- 4

```
font-family: "Noto Serif";
font-style: normal;
font-weight: normal;
font-size: 16px;
line-height: 20px;
```

Typography

The numbered code snippets correspond to their respective numbered text elements. The text and list item dividers should be given the primary font color.

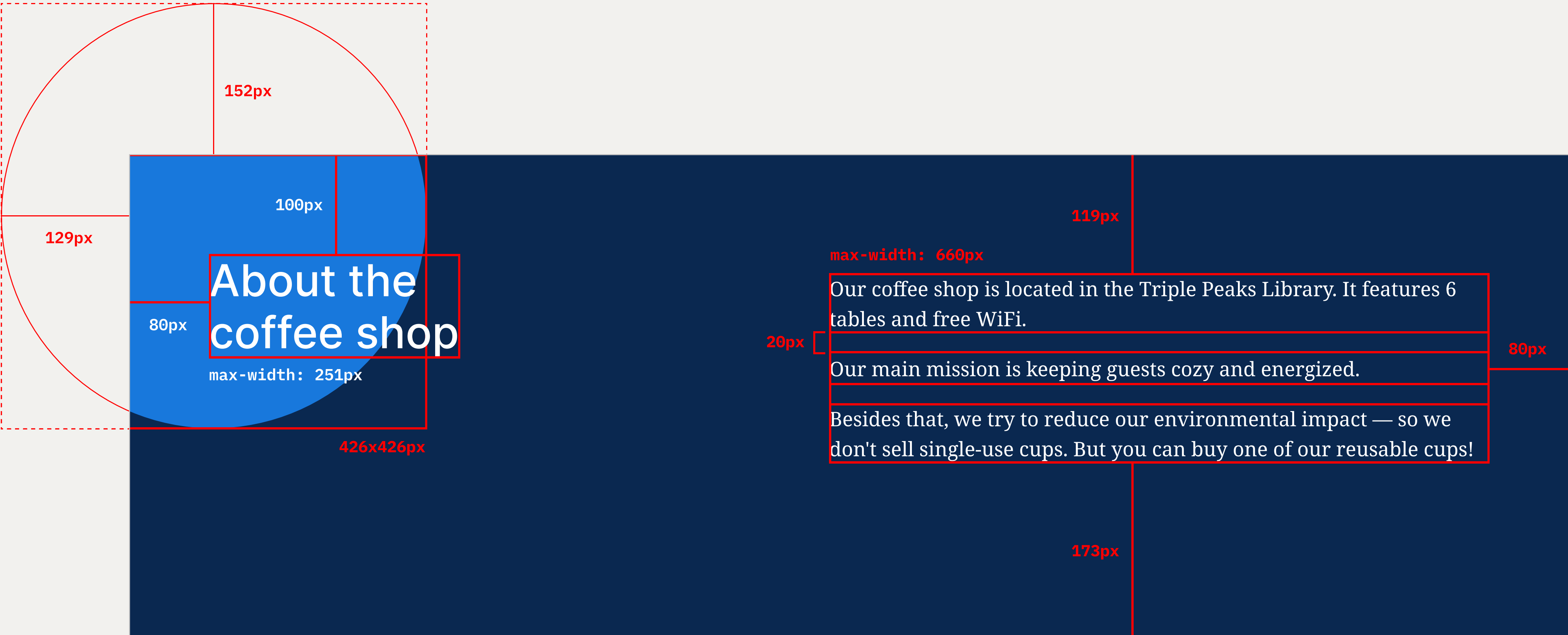
1.2. Section: About the coffee shop



Required elements and class names

Background: Set the background color to `#0A2850`.

1. block: `about`, element: `title`.
2. block: `about`, element: `content`.
3. block: `about`, element: `paragraph`.
4. block: `about`, element: `circle`.



Sizes and spacing

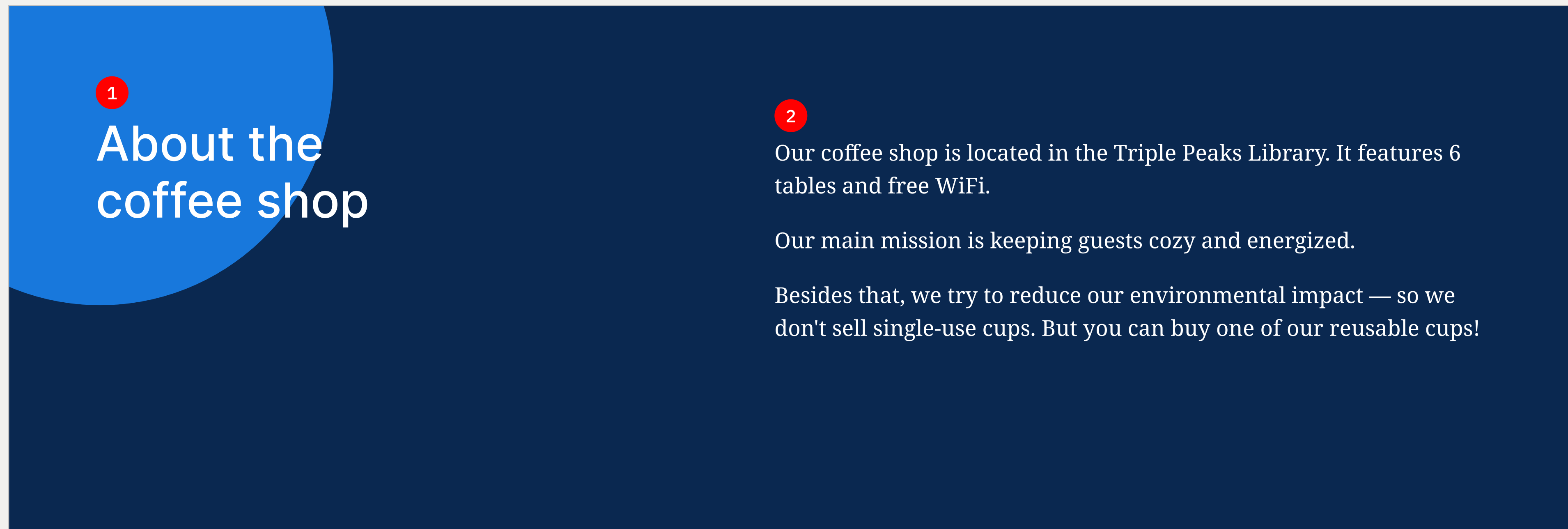
Add all of the sizes according to the picture above.

Notice the blue accent circle placed under the text. Use absolute position and assign z-index values to the elements of the sections to arrange them correctly.



Pro tip

You'll note there's no image corresponding to this element in the [images](#) directory. In this case, it's simple enough to create the circle using CSS. The trick to making an element look like a circle is to adjust its `border-radius` property. Specifically, if the element is a square (meaning its width and length are equal), then you can make it a circle by giving it `border-radius: 50%`.



1

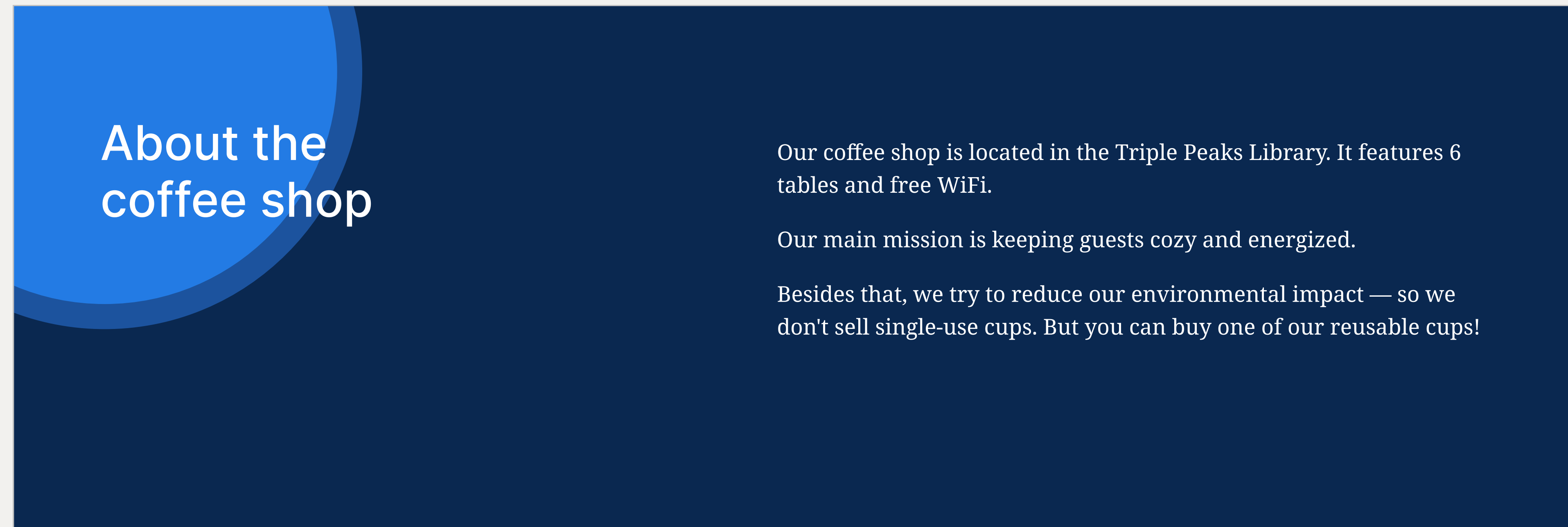
```
font-family: "Inter";  
font-style: normal;  
font-weight: 500;  
font-size: 44px;  
line-height: 52px;
```

2

```
font-family: "Noto Serif";  
font-style: normal;  
font-weight: normal;  
font-size: 20px;  
line-height: 30px;
```

Typography

The numbered code snippets correspond to their respective numbered text elements.



Animation

The blue circle will be animated to pulsate like a heart. You can see the animated version in the project description on the platform.

We need our base `about__circle` element to remain static, so we'll add a new one for the animation. Duplicate the `<div>` element with the static circle with its class, then add a BEM modifier to the duplicated one:

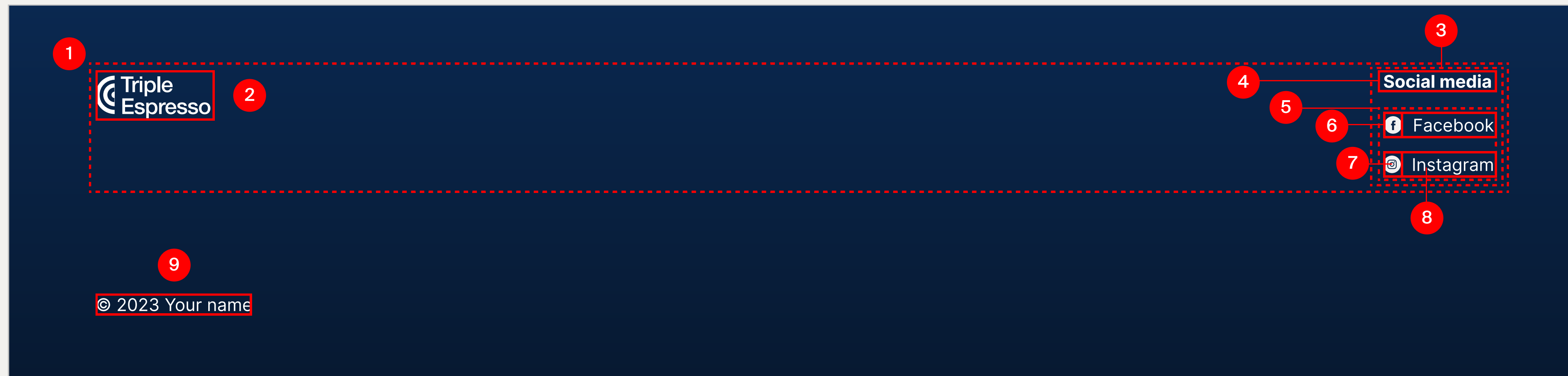
`about__circle_animation_blurred`.

Declare the animation with the `@keyframes` at-rule. Place this declaration in your `about.css` file. Give it a meaningful name, such as `pulsate`. The animation should have three frames:

1. At the start of the animation, make the animated circle invisible using the `opacity` property.
2. During the middle stage (50%), increase the opacity to `0.6`.
3. When the animation is complete, apply the `transform` property to scale the circle. In the scale function, specify any value to your liking, larger than `1` but no larger than `1.5`. Make the circle invisible again.

Implement the `animation` on the circle by adding the animation property to the modifier. Specify the animation's duration (`1.3s`), the timing function should be `ease-in-out`. There should be no delay, and the animation should be infinite.

1.3 Footer



Required elements and class names

Background: notice the fancy gradient. Figma provides you with ready-made code for gradients, so let's use the code for this one

```
background: linear-gradient(180deg, #0A2850 0%, #071931 100%);
```

1. block: `footer`, element: `content`.
2. block: `footer`, element: `logo`.
3. block: `footer`, element `column`.
4. block: `footer`, element: `social-heading`.
5. block: `footer`, element: `list`.
6. block: `footer`, element: `list-item`.
7. block: `footer`, element: `social-icon`.
8. block: `footer`, element: `social-link`.
9. block: `footer`, element: `copyright`. Remember to write your name here :)



Sizes and spacing

Add all the sizes according to the picture above.



- 1

```
font-family: "Inter";
font-style: normal;
font-weight: bold;
font-size: 16px;
line-height: 20px;
```
- 2

```
font-family: "Inter";
font-style: normal;
font-weight: normal;
font-size: 16px;
line-height: 20px;
```
- 3

```
font-family: "Inter";
font-style: normal;
font-weight: normal;
font-size: 16px;
line-height: 20px;
```

Typography

The numbered code snippets correspond to their respective numbered text elements.

1.4 Smooth Transitions

Create the required smooth transitions for the following elements when they are hovered over.

1. The `opacity` of the `nav__link` elements should transition from `100%` to `50%`.
2. The `background-color` of the `form__button` element should transition from `#1878dc` to `#3f96f3`.
3. The `opacity` of the `footer__social-link` elements should transition from `100%` to `70%`.

You can specify any values you'd like for transition properties, such as the duration or timing function.

2. Self-review

You are nearing the finish line for this project stage. Before submitting, there is one last crucial task: 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.

Once you're finished, you're ready to submit the project. Awesome job!

