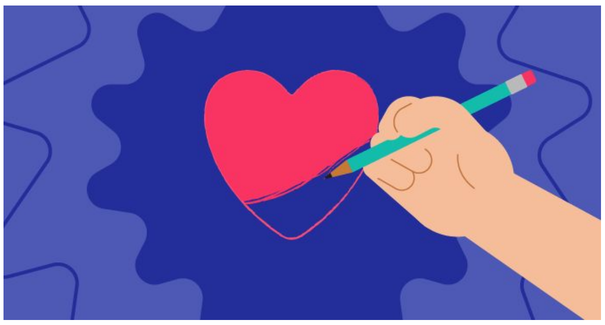
Anime expression

Introduction

What you will make

In this project, you will use HTML and CSS to create and style a webpage for an anime drawing tutorial.



Hypertext Markup Language (HTML) is used to create structure of a webpage. **Cascading Style Sheets (CSS)** describes exactly how a webpage should look. Without CSS, a webpage would look really boring.

You will:

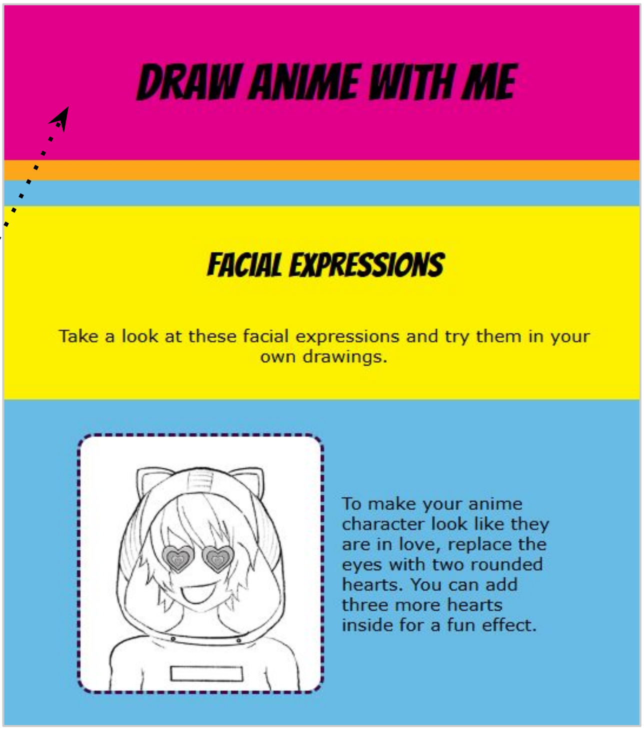
- Use **HTML** tags to structure a webpage
- Use **CSS** styles to apply layouts, colour palettes, and fonts to your webpage
- Add images and text content to your webpage

There are hundreds of millions of active **webpages**. If you use HTML with CSS, it can help your webpage attract people's attention.

Try it
Explore the anime expressions webpage.

- What do you like about it?
- What would you improve?

Go here to explore: rpf.io/anime-exp


DRAW ANIME WITH ME

FACIAL EXPRESSIONS

Take a look at these facial expressions and try them in your own drawings.

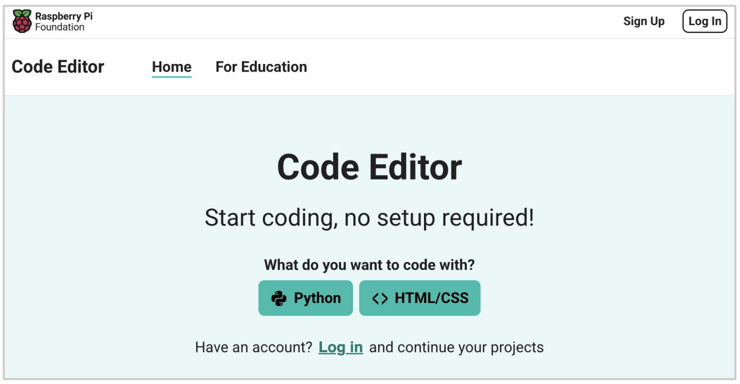
To make your anime character look like they are in love, replace the eyes with two rounded hearts. You can add three more hearts inside for a fun effect.

Where you will create:

Raspberry Pi Code Editor

The **Raspberry Pi Code Editor** can be used to write and run code in a web browser without installing any additional software and without the need to create an account (although if you're logged in to a Raspberry Pi Foundation account, your coding project is automatically saved).

1. Open Code Editor to write HTML code by writing this on a new browser rpf.io/codeeditor



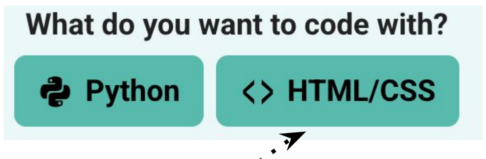
Code Editor

Start coding, no setup required!

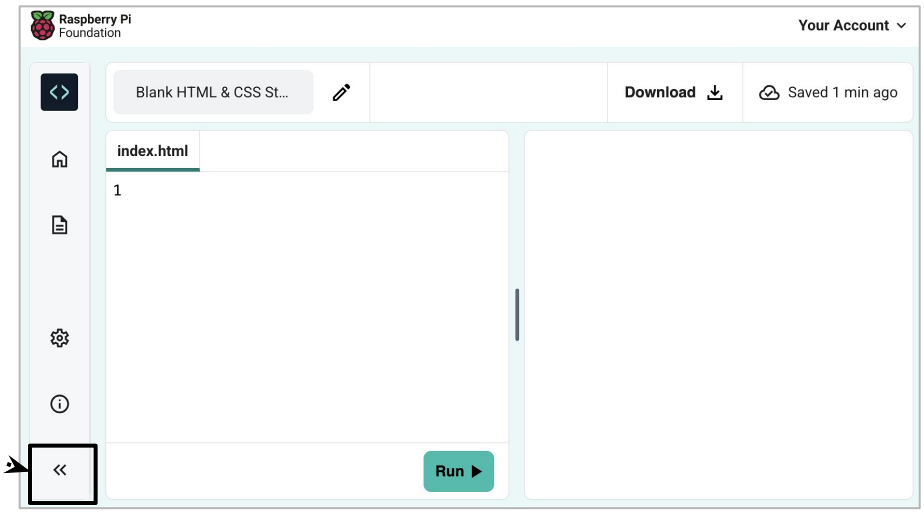
What do you want to code with?

Have an account? Log in and continue your projects

2. Click on the Start coding HTML/CSS button.



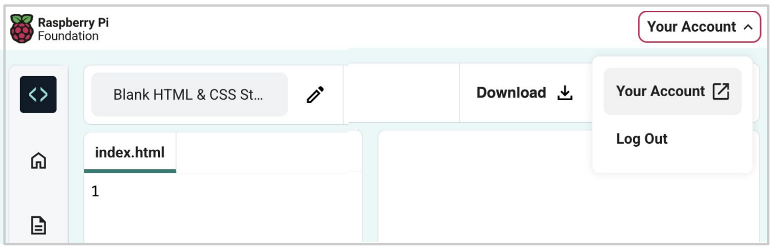
3. The Code Editor will open with a blank project.



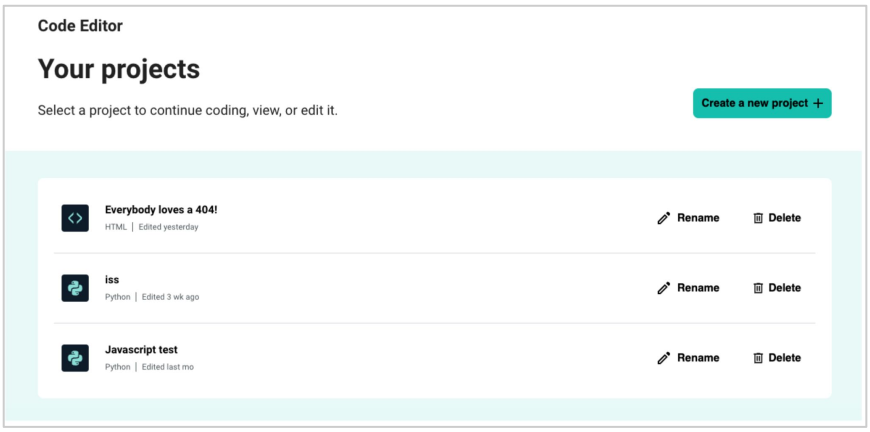
4. Here is the **expand button**, which helps in expanding and closing the sidebar panel of the code editor. The Project files panel will open. The blank project has two files: index.html and styles.css

5. You can access the Account menu from the top right of the Code Editor. When logged in, you will be able to access:

- My profile
- My projects
- Log out



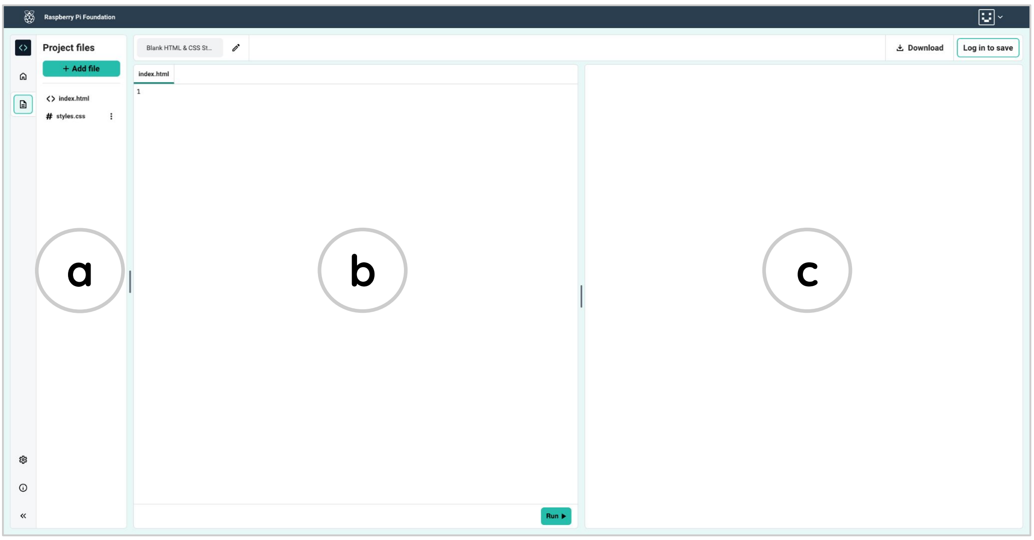
5. Clicking **My projects** will take you to the 'Your projects' page. Here, you can create a new project and see the projects list, where you have the option to continue, rename, or delete a project.



To return to the Code Editor, you can:

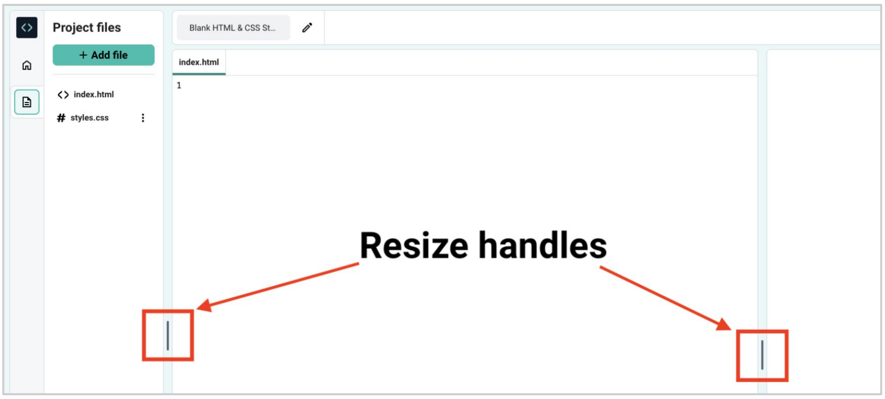
- Choose a saved project
- Create a new project
- Click the back button in your browser

5. There are three main panel in the code editor.



a. The Sidebar panel on the left
b. The Code panel in the middle
c. The Preview panel on the right

6. You can resize the panel using the handles.



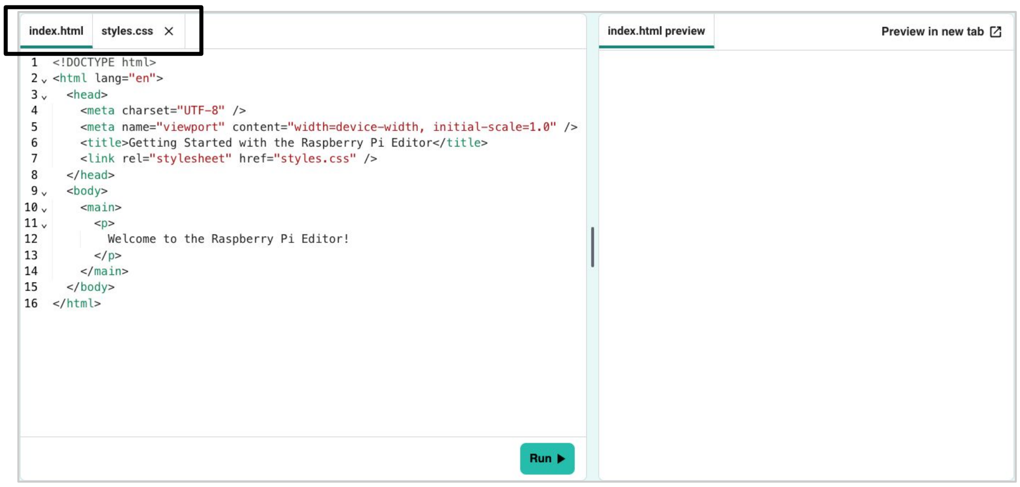
7. The project bar is at the top of the page and shows:

- The name of your project, with an edit button, so you can change the project name.
- A button to let you download your project files.
- A link to log in. This changes to 'save' when you are logged in.

8. The file name is at the top of the panel. In this example, the file name is:

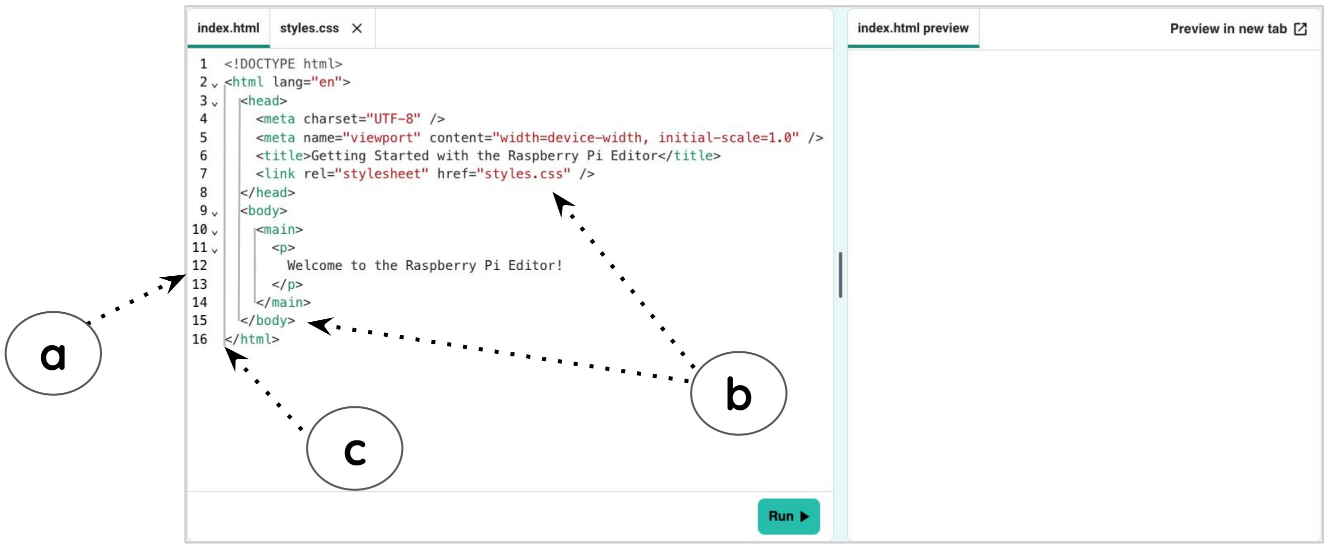
index.html : Where we can write the HTML code.

style.css : Where we can write the style sheet code.



```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Getting Started with the Raspberry Pi Editor</title>
    <link rel="stylesheet" href="styles.css" />
  </head>
  <body>
    <main>
      <p>
        Welcome to the Raspberry Pi Editor!
      </p>
    </main>
  </body>
</html>
```

9. The code panel includes the following features:



```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Getting Started with the Raspberry Pi Editor</title>
    <link rel="stylesheet" href="styles.css" />
  </head>
  <body>
    <main>
      <p>
        Welcome to the Raspberry Pi Editor!
      </p>
    </main>
  </body>
</html>
```

a. Line numbering

The Code Editor adds line numbers automatically to help you track your code and collaborate with others.

b. Syntax highlighting

Syntax highlighting makes code easier to read by using colours and styles for different elements. It helps developers see parts of the code clearly—for example, tags are green and values are red in the Code Editor.

c. Indentation

The Code Editor formats your code with spaces (indentation) to show structure. Indentation doesn't change how HTML looks in a browser, but it makes code easier to read. For example, <head> and <body> are child elements nested inside the parent <html>.

On this code editor, HTML and CSS code has to be written. To know more about the editor open this in a new browser: rpf.io/guide

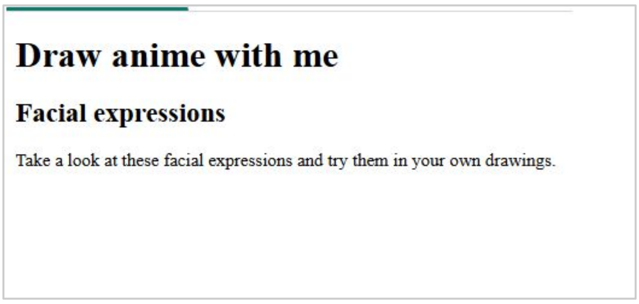
1. Start your webpage

In this step, you will add a header and an introduction to your anime webpage.

In HTML you can type words directly into the code to make the words appear, unformatted, on the webpage.

Step 1:
Open the starter project
http://rpf.io/anime-starter

Your starter project contains some HTML that you will learn more about throughout the project.

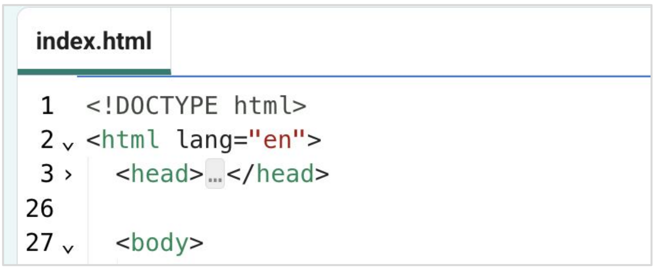


Draw anime with me

Facial expressions

Take a look at these facial expressions and try them in your own drawings.

Step 2:
Click on the small triangle next to line 3 to collapse the **<head>**.



```
1   <!DOCTYPE html>
2 ⌄ <html lang="en">
3 ›   <head>…</head>
26
27 ⌄   <body>
```

Add a header

Step 3:
Add **<h1></h1>** tags inside your **<header></header>** tags.

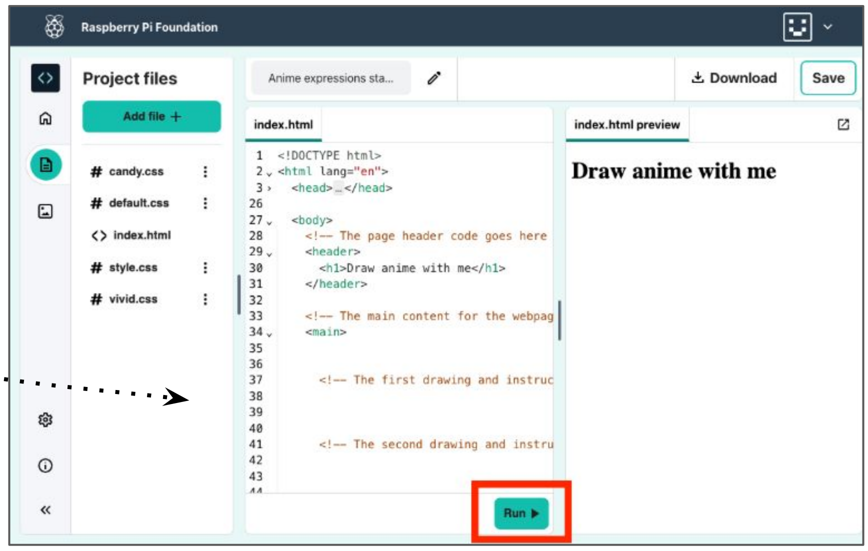
Typically, a webpage has three parts. A **header**, the **main** content, and a **footer**.

```
  <body>
    <!-- The page header code goes here -->
    <header>
      <h1></h1>
    </header>
```

Step 4:
Add the text Draw anime with me between the two <h1> tags.

```
<body>
  <!-- The page header code goes here -->
  <header>
    <h1>Draw anime with me</h1>
  </header>
```

Step 5:
Test: Click the **Run** button.
The output will appear on the right:



```
<!DOCTYPE html>
<html lang="en">
  <head>…</head>

  <body>
    <!-- The page header code goes here
    <header>
      <h1>Draw anime with me</h1>
    </header>

    <!-- The main content for the webpag
    <main>


      <!-- The first drawing and instruc



      <!-- The second drawing and instru
```

Draw anime with me

You will see that the text inside the **<h1>** tags is styled as bold with a large font.

Add the first section in your main content

Any main content should be placed between the **<main>** tags. On your webpage, the main content is broken down into sections. Your webpage needs an introduction section.

Step 6:
Add **<section></section>** tags between the **<main>** tags.

```
<!-- The main content for the webpage goes
<main>
  <section>

  </section>
    <!-- The first drawing and instructio
```

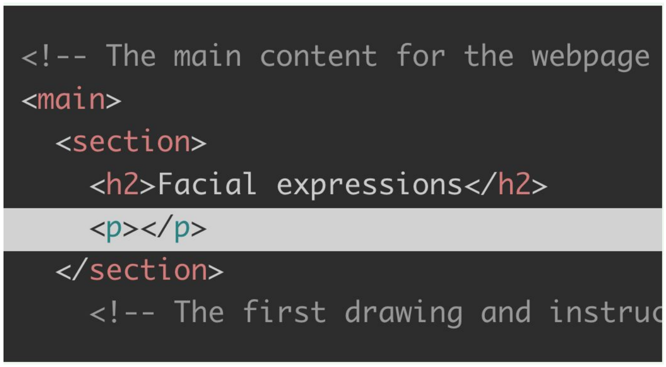
Tip: As you build your webpage, you will add other tags inside your section.

Step 7:
To add subheadings add the subheading tags **<h2>** between the **<section>** tags.
Now enter the subheading text **Facial expressions** between the **<h2>** tags.

```
<!-- The main content for the webpage goes
<main>
  <section>
    <h2>Facial expressions</h2>
  </section>
    <!-- The first drawing and instructio
```

Step 8:
Test: Click the **Run** button.
Notice how the text on your webpage is slightly smaller than the big heading above and has bold styling. This is because **<h2>** is a smaller heading than **<h1>**.

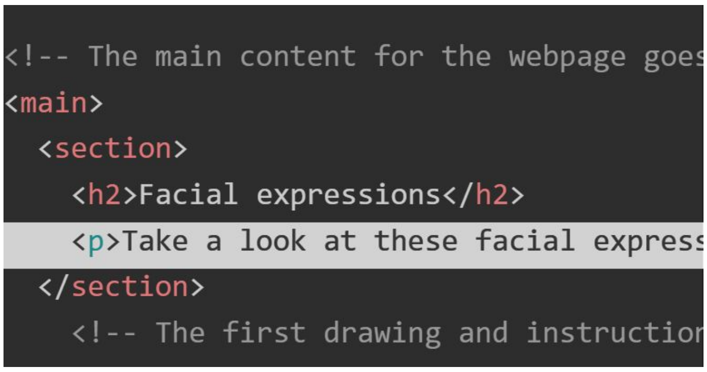
Step 9:
Now add a paragraph of text as an introduction to your anime webpage.
Underneath your **<h2>** heading code, add the paragraph **<p>** tags.



```
<!-- The main content for the webpage
<main>
  <section>
    <h2>Facial expressions</h2>
    <p></p>
  </section>
    <!-- The first drawing and instruc
```

Step 10:
Between the <p> tags, you need to add in this introductory text:

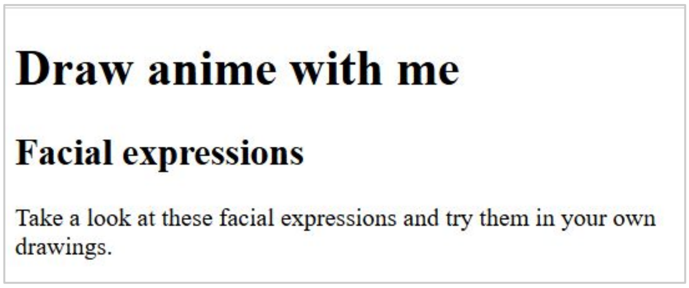
Take a look at these facial expressions and try them in your own drawings.



```
<!-- The main content for the webpage goes
<main>
  <section>
    <h2>Facial expressions</h2>
    <p>Take a look at these facial express
  </section>
    <!-- The first drawing and instruction
```

Step 11:
Test: Click the **Run** button.

Well done! Your page now has a header, a subheading, and an introductory paragraph.



Draw anime with me

Facial expressions

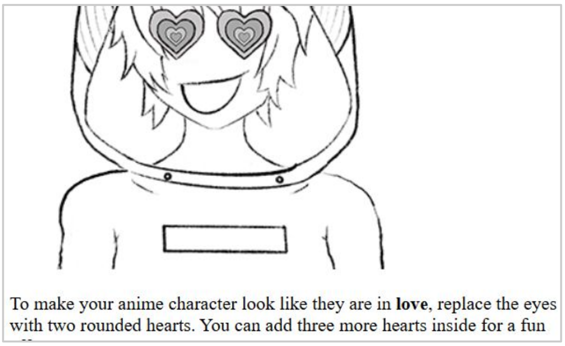
Take a look at these facial expressions and try them in your own drawings.

2. Add a facial expression

In this step, add the first drawing and instruction to your webpage.

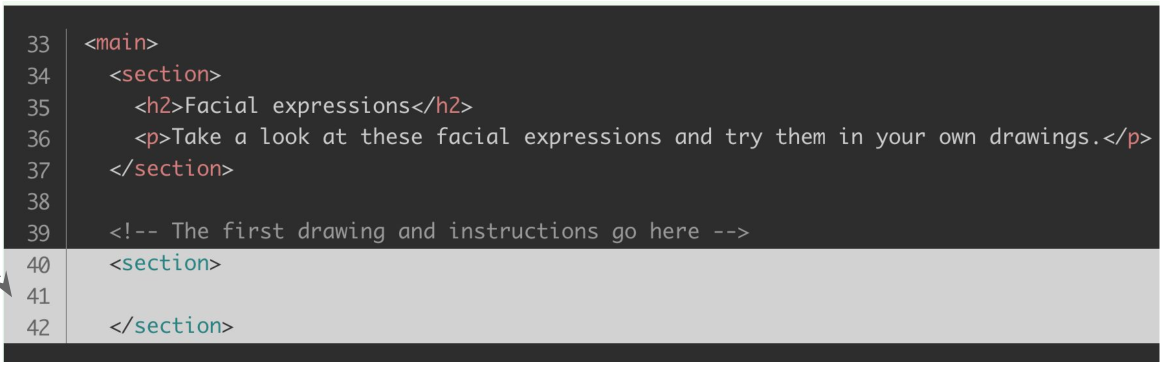
First, create a section for each facial expression on the webpage.

Find the comment **<!-- The first drawing and instructions go here -->**.



To make your anime character look like they are in **love**, replace the eyes with two rounded hearts. You can add three more hearts inside for a fun

Step 1:
Add in the **<section></section>** tags for your first drawing and instruction content.



```
<main>
  <section>
    <h2>Facial expressions</h2>
    <p>Take a look at these facial expressions and try them in your own drawings.</p>
  </section>

  <!-- The first drawing and instructions go here -->
  <section>

  </section>
```

Your starter project contains images to use in this project. To include an image on a webpage, you need to know the filename. First, add an image called **love.png**.

Step 2:
Inside your new section, add an **<img>** tag to display an image. The **src** **attribute** gives the name of the image.

The **<img>** tag doesn't have an end tag.

```
<!-- The first drawing and instructions
<section>
  <img src="love.png">
</section>
```

Step 3:
Add the **alt** attribute to provide alternative text for people who cannot view the image.

You can copy the description of your image and paste it into your code: **The love facial expression.**

```
<!-- The first drawing and instructions g
<section>
  <img src="love.png" alt="The love facia
</section>
```

Alternative (Alt) text is a description of an image and is important in accessible web design to describe images to people who are unable to see them. The text does not appear on the webpage but it is read aloud by screen readers.

Step 4:
Test: Click the **Run** button. The **love.png** image appears on your webpage.

Step 5:
Add a paragraph of text in **<p></p>** tags to describe how to draw the love anime facial expression.
You can copy the paragraph and paste it into your code:
<p>To make your anime character look like they are in love, replace the eyes with two rounded hearts. You can add three more hearts inside for a fun effect.</p>

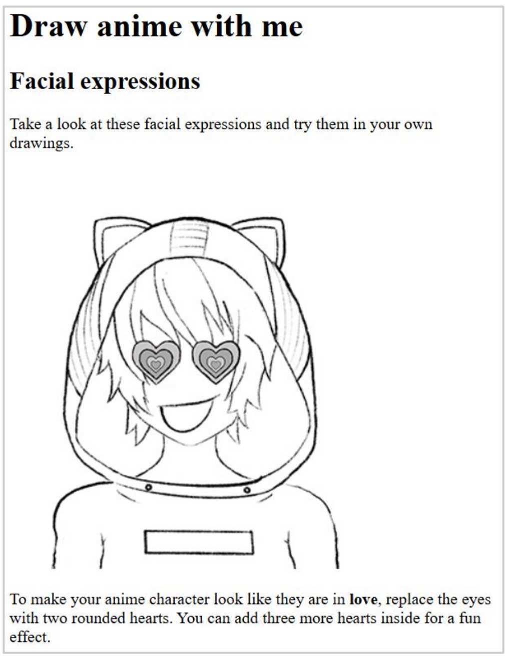
```
<!-- The first drawing and instructions g
<section>
  <img src="love.png" alt="The love facia
  <p>To make your anime character look li
</section>
```

The instructions appear below your image and the word **love** is in bold.

Step 6:
Add **<strong>** tags around the word 'love', it makes important text **bold**.

Test: Click the **Run** button.

```
expression.">
e they are in <strong>love</strong>,
```



Draw anime with me

Facial expressions

Take a look at these facial expressions and try them in your own drawings.

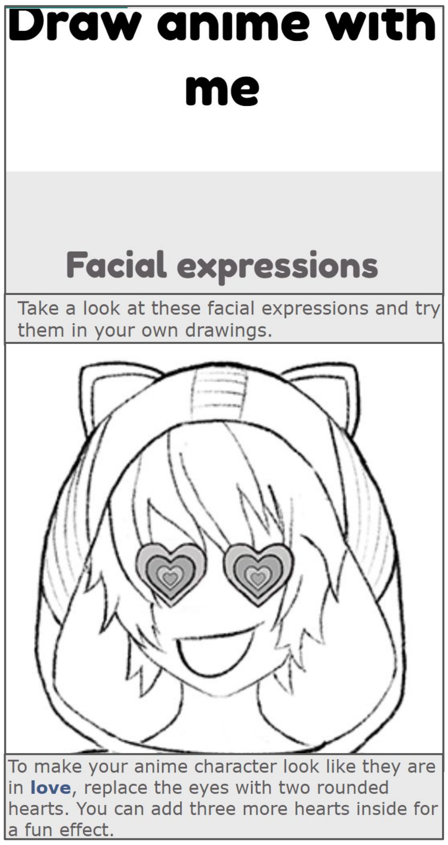
To make your anime character look like they are in **love**, replace the eyes with two rounded hearts. You can add three more hearts inside for a fun effect.

3. Style your page

You have used HTML to add tags to your webpage.

Now it is time to use CSS to add styles to your page.

This step shows you how to change the colours, fonts, and layout on your webpage.



Draw anime with me

Facial expressions

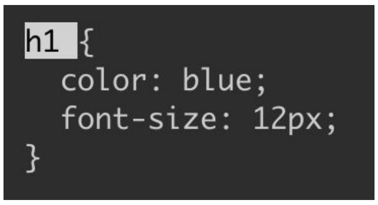
Take a look at these facial expressions and try them in your own drawings.

To make your anime character look like they are in **love**, replace the eyes with two rounded hearts. You can add three more hearts inside for a fun effect.

Cascading Style Sheets (CSS) is the language that you use to tell the web browser exactly how your webpage should look, which includes the positioning, colours, and fonts. We call this the style.

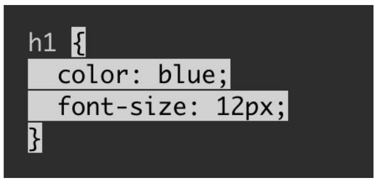
Every **rule** in CSS is made up of two parts: the **selector** and the **declaration.**

a. The **selector** is the part of HTML that you want to style. In this example it is h1.



```
h1 {
   color: blue;
   font-size: 12px;
}
```

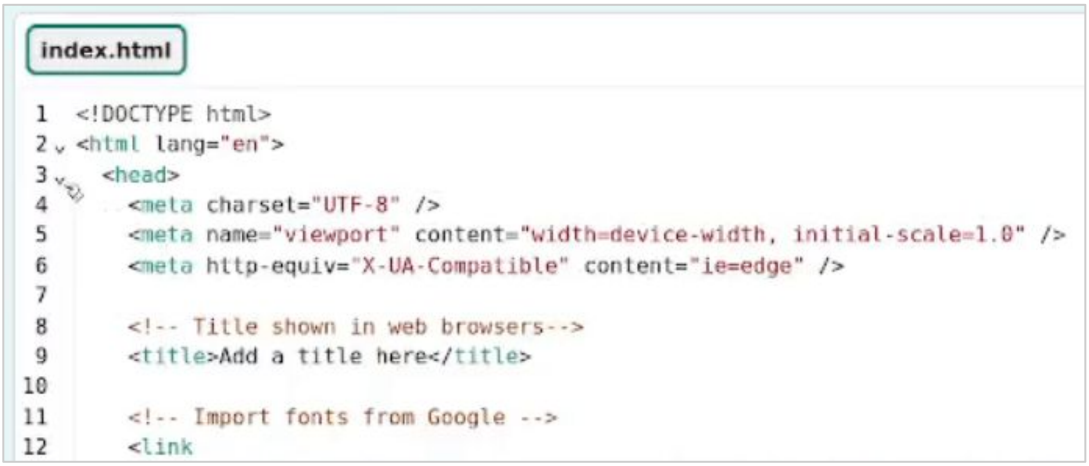
b. The declaration is in curly brackets {}. It gives instructions of the styles that should be used.



```
h1 {
  color: blue;
  font-size: 12px;
}
```

Link the CSS file

Step 1:
Unfold the **<head>** section of your code so that you can view the code inside it.

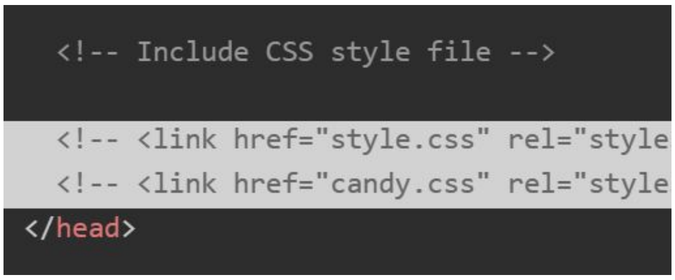


```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />

    <!-- Title shown in web browsers-->
    <title>Add a title here</title>

    <!-- Import fonts from Google -->
    <link
```

Step 2:
At the bottom of your **<head></head>** section, there are links to two CSS style sheets that are currently commented out so that they are ignored by the web browser.

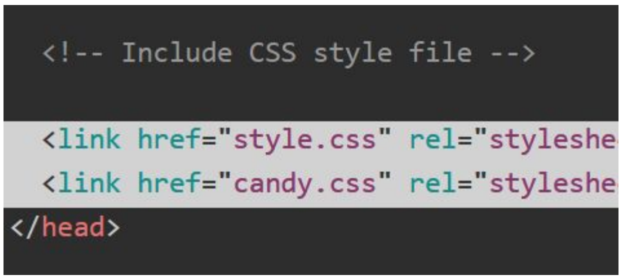
Remove the **<!--** and **-->** arrows from the start and end of both lines of link code:

Before



```
  <!-- Include CSS style file -->

  <!-- <link href="style.css" rel="style
  <!-- <link href="candy.css" rel="style
</head>
```

After



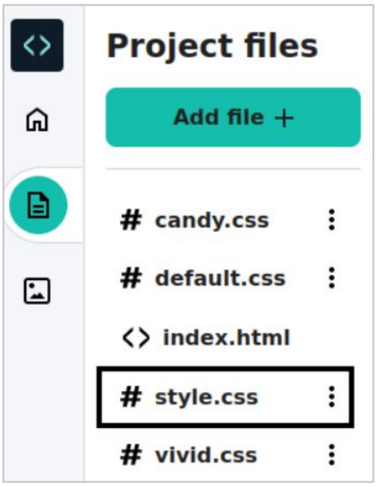
```
  <!-- Include CSS style file -->

  <link href="style.css" rel="styleshe
  <link href="candy.css" rel="styleshe
</head>
```

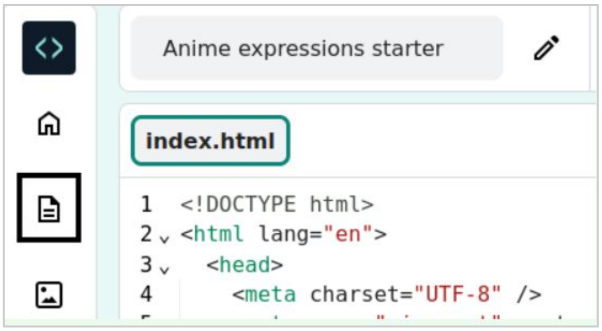
Step 3:
Test: Click the **Run** button.

HTML elements have default browser styles that you have seen as you have written your HTML code.

Take a look at your webpage in the right-hand pane. Notice that the styles and layout of your output has now changed.

Step 4:
Click on the **Project files** icon in the Code Editor then select the **style.css** file top open in in a new tab.



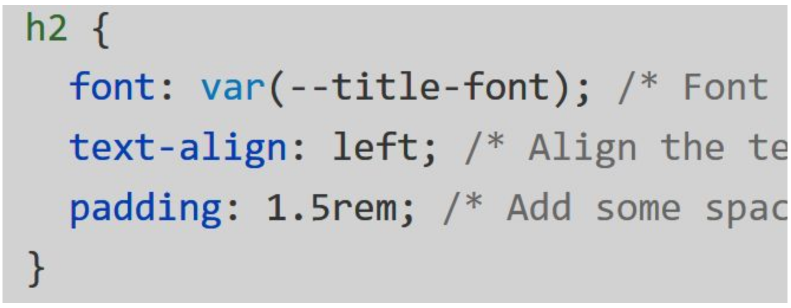


```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
```

This CSS file contains all of the CSS for your project. You will find out about some key parts of this CSS file as you create your webpage.

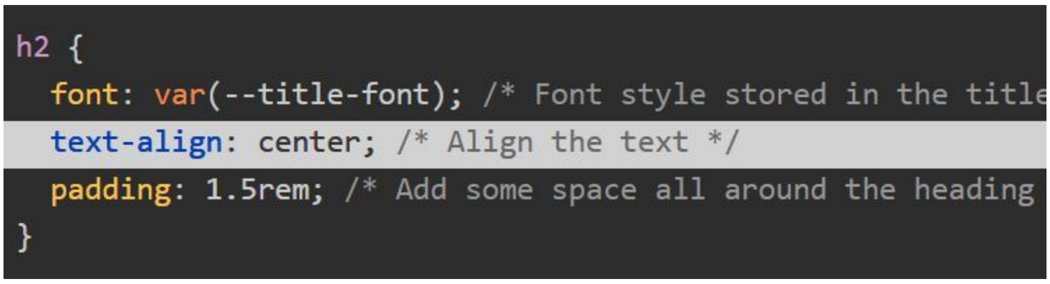
When you add CSS styling to an **element,** it applies that styling to every single element on the page that has the same tag.

Step 5:
Scroll down and find the rule that controls the style of the **<h2>**.



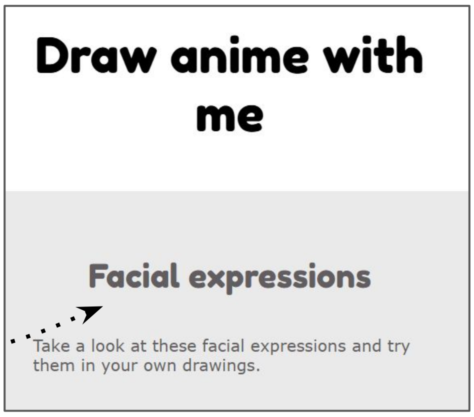
```
h2 {
  font: var(--title-font); /* Font
  text-align: left; /* Align the te
  padding: 1.5rem; /* Add some spac
}
```

Step 6:
At the moment, the **<h2>** heading is aligned to the left. Change the **text-align** property of the **h2** rule to **center**.



```
h2 {
  font: var(--title-font); /* Font style stored in the title
  text-align: center; /* Align the text */
  padding: 1.5rem; /* Add some space all around the heading
}
```

Step 7:
Test: Click the Run button.


Draw anime with me
Facial expressions
Take a look at these facial expressions and try them in your own drawings.

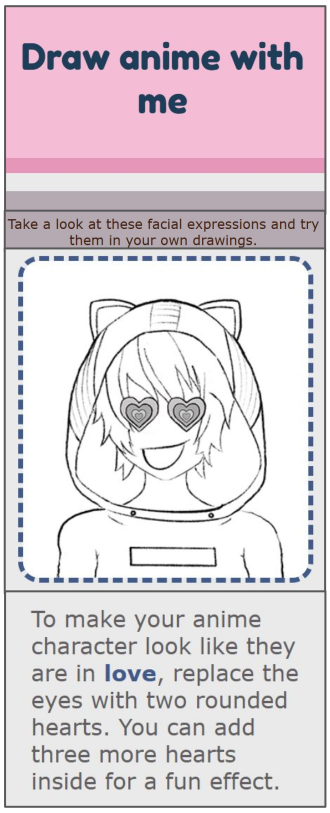
Look at your webpage and make sure the '**Facial expressions**' text is centred.

4. Style with classes

This step shows you how to add classes to customise the styles on your page.

If you want to apply styling to specific elements, you can create a **class** in a CSS file. You can then add a **class= attribute** to an element in your HTML code to let the browser know what styling should be applied.

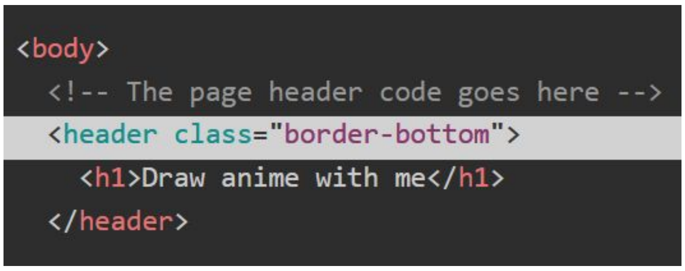
The class styling overrides any element styling that has already been applied. Notice that the changes take place as you add the classes to your code.


Draw anime with me
Take a look at these facial expressions and try them in your own drawings.
To make your anime character look like they are in love, replace the eyes with two rounded hearts. You can add three more hearts inside for a fun effect.

Your CSS file has a custom CSS class called **border-bottom**. This class adds a thick, solid-coloured line border to the bottom of any HTML element that uses it.

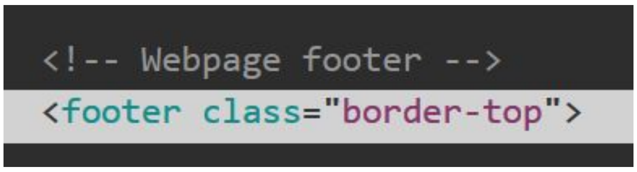
Step 1:
Go to your **index.html** file and find your **header**.

Add **class="border-bottom"** after the word **header** in your **header** tag.



```
<body>
  <!-- The page header code goes here -->
  <header class="border-bottom">
    <h1>Draw anime with me</h1>
  </header>
```

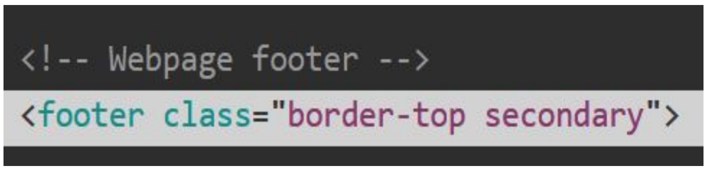
Step 2:
Add the **border-top** class to your **footer** code to apply a thick border to the top of your footer.



```
<!-- Webpage footer -->
<footer class="border-top">
```

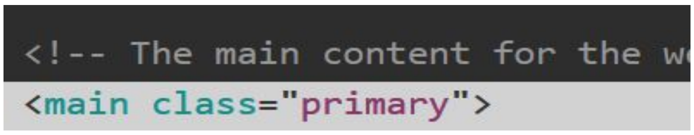
The **primary** class sets a contrasting background and text colour for most of the main content.
The **secondary** class sets an additional colour combination that look good with the colours in the **primary** class.

Step 3:
Add the **secondary** class to your **footer** code to apply a different colour background to your footer.



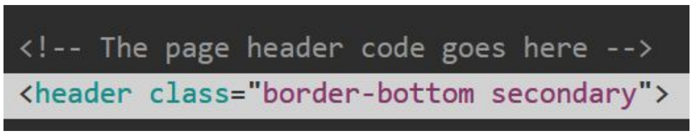
```
<!-- Webpage footer -->
<footer class="border-top secondary">
```

Step 4:
Add **class="primary"** to **<main>**



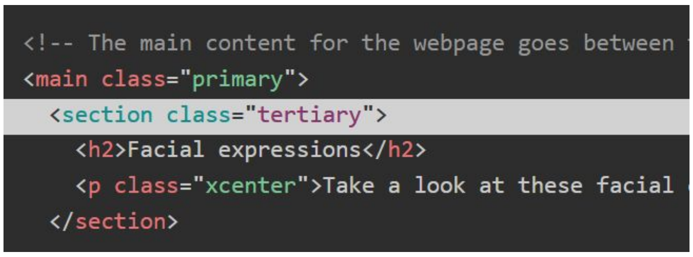
```
<!-- The main content for the w
<main class="primary">
```

Step 5:
Add **secondary** to **<header>**.



```
<!-- The page header code goes here -->
<header class="border-bottom secondary">
```

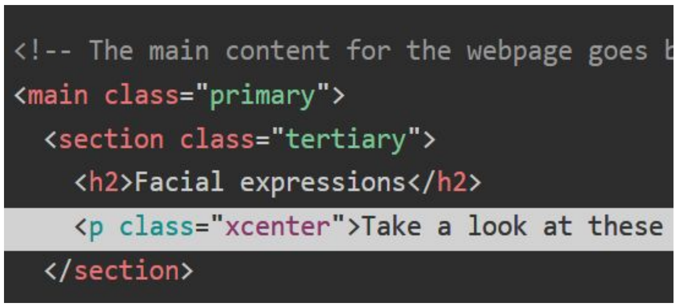
Step 6:
Add **class="tertiary"** to the **first** **<section>** element.



```
<!-- The main content for the webpage goes between 
<main class="primary">
  <section class="tertiary">
    <h2>Facial expressions</h2>
    <p class="xcenter">Take a look at these facial 
  </section>
```

The **xcenter class** in your CSS file aligns items horizontally across the page.

Step 7:
Add **class="xcenter"** to the **<p>** in the same section.

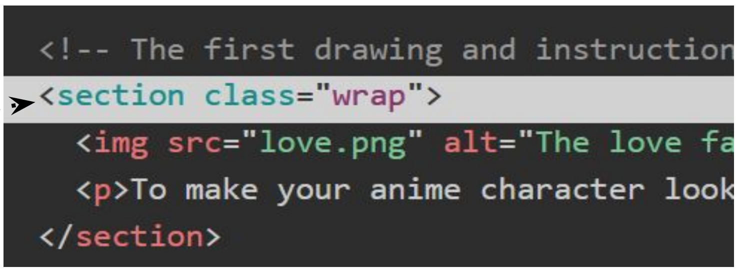


```
<!-- The main content for the webpage goes b
<main class="primary">
  <section class="tertiary">
    <h2>Facial expressions</h2>
    <p class="xcenter">Take a look at these
  </section>
```

Webpages can be viewed on many different devices and should be **responsive** to each device. This means that if a user views your page on a mobile phone, it should respond to a smaller screen and if they view it on a desktop PC, it should respond to a larger screen.

CSS can change the layout on a webpage, as well being used to change colours, fonts, and borders.

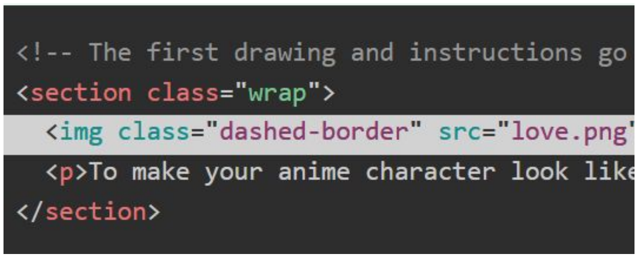
Step 8:
Find the second **<section>** . Add **class="wrap"** to the **<section>** tag.



```
<!-- The first drawing and instruction
<section class="wrap">
  <img src="love.png" alt="The love fa
  <p>To make your anime character look
</section>
```

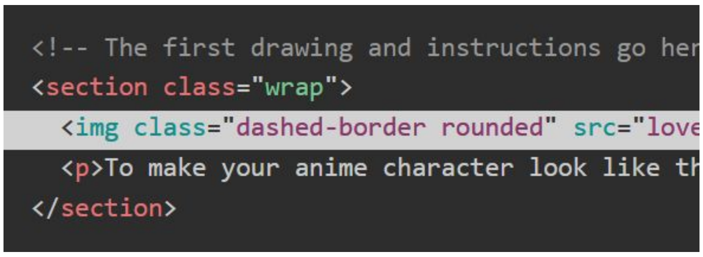
You can also add coloured borders in different styles to HTML elements. The **dashed-border** class in the style file creates a dashed border.

Step 9:
Add the **dashed-border** class to the **<img>**.



```
<!-- The first drawing and instructions go
<section class="wrap">
  <img class="dashed-border" src="love.png'
  <p>To make your anime character look like
</section>
```

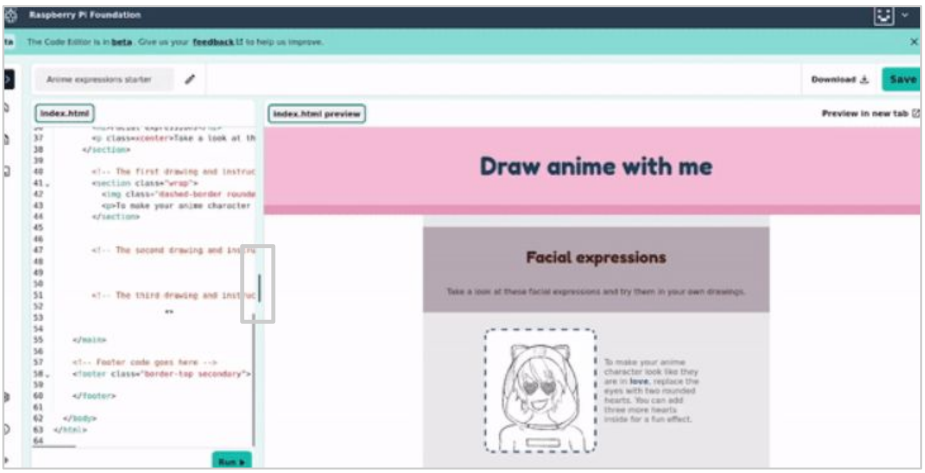
Step 10:
You can make the corners of an element rounded with the **rounded** class. Add the **rounded** class to the **<img>**.



```
<!-- The first drawing and instructions go her
<section class="wrap">
  <img class="dashed-border rounded" src="love
  <p>To make your anime character look like th
</section>
```

Step 11:
Test: Click the **Run** button.

Step 12:
- Drag the bar between the text editor and your webpage to make the webpage narrower.
- Drag the bar back after you test it, so you can see the image and text side-by-side.



Raspberry Pi Foundation
The Code Editor is in beta. Give us your feedback to help us improve.
Anime expressions starter
Download
Save
index.html
index.html preview
Preview in new tab
Run

```
    <p class=center>Take a look at th
  </section>

    <!-- The first drawing and instruc
    <section class="wrap">
      <img class="dashed-border rounde
      <p>To make your anime character
    </section>


    <!-- The second drawing and instru


    <!-- The third drawing and instruc
            ..

  </main>

  <!-- Footer code goes here -->
  <footer class="border-top secondary">

  </footer>

</body>
</html>
```

Draw anime with me

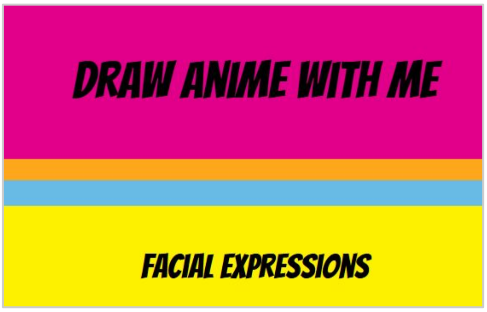
Facial expressions

Take a look at these facial expressions and try them in your own drawings.

To make your anime character look like they are in **love**, replace the eyes with two rounded hearts. You can add three more hearts inside for a fun effect.

5. Colours and fonts

In this step, you can try out different colour palette and font choices.



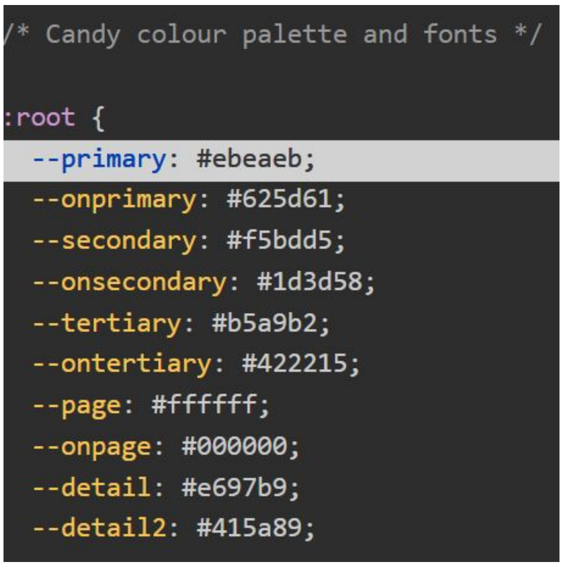
Now that you have started to add custom classes to your code, you may have noticed that colour has been added to the page. In CSS, you can use **variables** to create a colour palette for your webpage.

CSS variables start with two dashes: **--primary** .

Colours are specified using hexadecimal notation (hex) and begin with '#'. There are lots of websites where you can find hex colours to use.

Step 1:
Go to your **candy.css** file. This file sets the colour variables for the candy colour palette.

In the candy colour palette, the **--primary** variable is set to **#ebeaeb**, a pale grey.



```
/* Candy colour palette and fonts */

:root {
  --primary: #ebeaeb;
  --onprimary: #625d61;
  --secondary: #f5bdd5;
  --onsecondary: #1d3d58;
  --tertiary: #b5a9b2;
  --ontertiary: #422215;
  --page: #ffffff;
  --onpage: #000000;
  --detail: #e697b9;
  --detail2: #415a89;
```

Tip: CSS uses different comment markers to HTML. Multiline comments start with **/*** and end with ***/.** The browser ignores code that is inside the comment markers.

/* Candy colour palette and fonts */ is an example of a CSS comment.

You can also use variables for fonts. The **--header-font** is set to **3rem 'Fredoka One', cursive;**

3rem means that this font should be three times the normal font size.

'Fredoka One', cursive means that the browser should use the **'Fredoka One'** font if it can. If this font isn't available, the browser should use the **fallback font**, which is **cursive.**

Fonts for the web
Web designers carefully consider the font styles for their website.

The **three** most common categories of font are:

- Library fonts
- Web safe fonts
- Fallback fonts

Library fonts are typically imported from a third-party library such as Google Fonts. Companies sometimes pay a fee to use a font as part of their website branding.
Web safe fonts are standard fonts that should be available through any web browser. However, you can never be 100% sure that this is the case. Here is a list of web safe fonts:

- Arial
- Verdana
- Helvetica
- Tahoma
- Trebuchet MS
- Times New Roman
- Georgia
- Garamond
- Courier New
- Brush Script MT

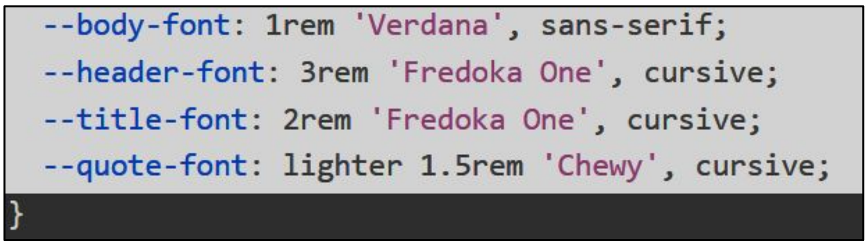
Fallback fonts are generic font families that are used to match the styling that the web designer would like to use. The main font families are:

- Serif: a font style typically used in print publishing, letters have tiny decorative edges called 'serifs'
- Sans-serif: a clean screen-readable font without the decorative edges

- Monospace: a font where each character uses the same width of space
- Cursive: a handwriting font
- Fantasy: a decorative font typically used for big headings

If a fallback font isn't listed, then the web browser uses the browser's default font, which is typically Times New Roman.

Step 2:
Find the variables that set the fonts for your webpage.



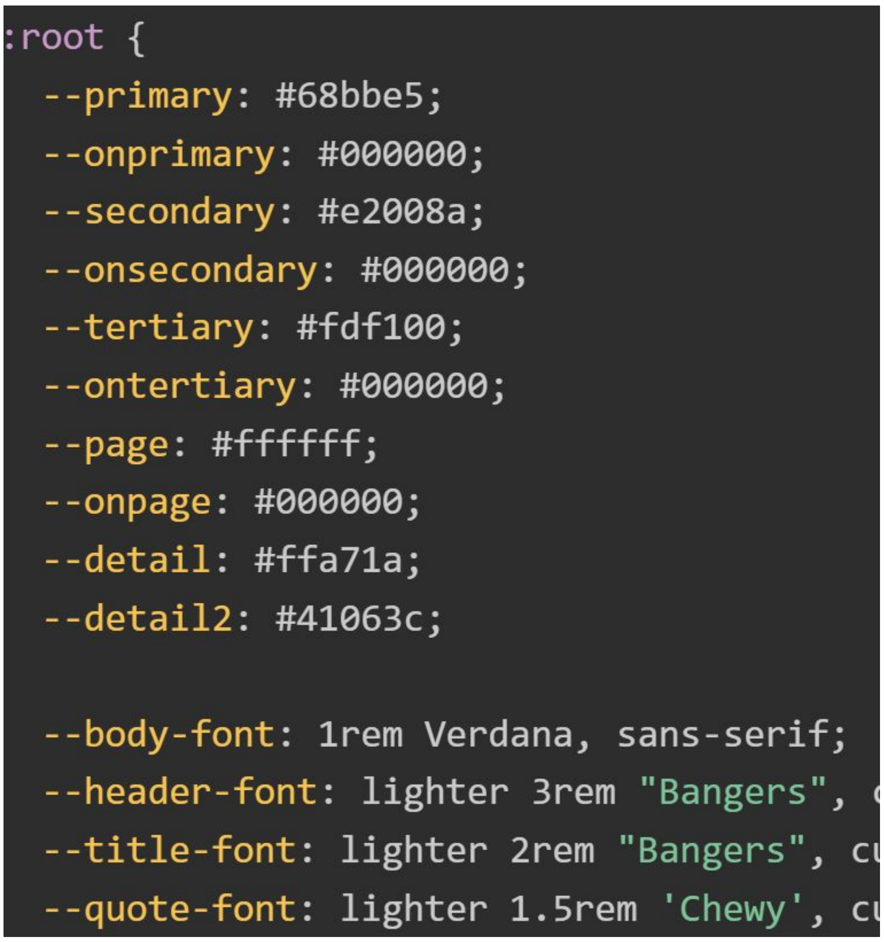
```
  --body-font: 1rem 'Verdana', sans-serif;
  --header-font: 3rem 'Fredoka One', cursive;
  --title-font: 2rem 'Fredoka One', cursive;
  --quote-font: lighter 1.5rem 'Chewy', cursive;
}
```

The **primary** colours are designed to be used the most in the main content of the page, followed by the **secondary** and then **tertiary** colours. This means that you can easily design new colour palettes and switch between them.

The starter project also includes a vivid colour palette file called **vivid.css**.

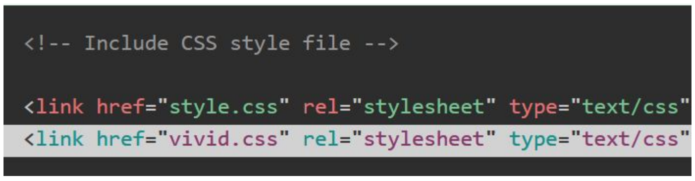
Step 3:
Find the **vivid.css** file.

Notice that the colour and font variables have the same names as in the **candy.css** file, but the colours and fonts used are different in this colour palette.



```
:root {
  --primary: #68bbe5;
  --onprimary: #000000;
  --secondary: #e2008a;
  --onsecondary: #000000;
  --tertiary: #fdf100;
  --ontertiary: #000000;
  --page: #ffffff;
  --onpage: #000000;
  --detail: #ffa71a;
  --detail2: #41063c;

  --body-font: 1rem Verdana, sans-serif;
  --header-font: lighter 3rem "Bangers", c
  --title-font: lighter 2rem "Bangers", cu
  --quote-font: lighter 1.5rem 'Chewy', cu
```

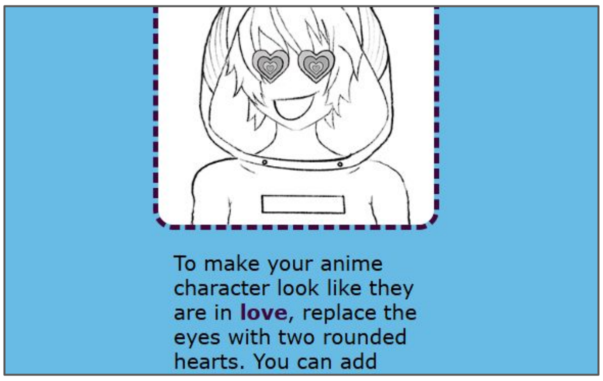
Step 4:
Go to **index.html** and change the CSS link code to link to the **vivid.css** file



```
<!-- Include CSS style file -->

<link href="style.css" rel="stylesheet" type="text/css"
<link href="vivid.css" rel="stylesheet" type="text/css"
```

Step 5:
Test: Click the **Run** button.

Make sure your webpage now uses the brighter colours and different fonts, as defined in the **vivid.css** file.



To make your anime character look like they are in **love**, replace the eyes with two rounded hearts. You can add