HTML (Hypertext Markup Language)

(i) Elements

* They are denoted as h_1, h_2...
* They are written in order of importance with h_1 being the most important.
* They are denoted as:

  <h1> element </h1>

* p is an element used to start paragraphs
* (a) anchor element links another web page. The link text is placed before the closing tag. The page to be linked is put together with the oppening tag. - href attribute introduces the link

(ii) Comments

* Allows you to leave a message without affecting the browser display. It allows you to also make a code inactive. It starts with <!-- text -->

```
<html>
  <body>
  ~~~~
  </body>
</html>
```

Starting the code

(iii) Nesting

* This used to identify content areas in the code
* It is done by adding a main element around the important sections of the code.
* The nested components are indented 2 spaces behind the nesting component.

(iv) Adding img element

* Does not have a closing tag hence known as a self closing tag.

(v) Attributes

src - specifies the location of an image

alt - alternative text printed if the image doesn't load

target="_blank" - makes link open in a new tab

* Any text can be turned into a link by adding (a) tags around these words

(vi) Selection element (selection)
· Seperates content

(vii) Unordered list element (ul)
· This is use to make a list
· Listed values are nested using <li></li>

(viii) Figure element (figure)
· Represents self contained content and allows you to associate an image with a caption

(viii) Figcaption element (figcaption)
· Adds the caption

(ix) Empasis (em)
· Allows you to empasize some parts of text
*
(x) Odered list element (ol)
· Gives a numbered list.

(xi) Strong element (strong)
· Indicates a text of strong importance/urgency.

(xii) Form element (form)
· Allows to accept user input.

(xiii) Action attribute (action)
· Indicates where the form should be sent.
~~(xiv)~~
(xiv) Input element (input) * Self closing
· Allows you several ways to collect data from a web form

(xv) Text attribute (text)
· Allows you to get texts from the user

(xvi) Placeholder text (placeholder)
- Gives people a hint about the kind of infor to enter into an input.

(xvii) Required attribute * Does not require a value
- Prevents a user from submitting a form when required info. is missing

(xviii) Button element (button)
- Creates a clickable button

(xix) Radio buttons
- For questions where you want only one answer from multiple choices

(xx) Label elements
- Helps to associate the text for an input element

(xxi) The [id] attribute
- Used to identify specific HTML elements Each id must have unique attribute for all other [id] values for the entire page

(xxii) Name attribute
- Makes selecting one radio button automatical deselects the other. The value on the attribu contains the labels of the two buttons

(xxiii) Value attribute
- Submits the actual value selected in the case that a radio button is used when multiple choices a present

(xxiii) Fieldset element (fieldset)
· Groups related inputs and labels

(xxiv) Legend element (legend)
· Acts as a caption for the fieldset element

(xxv) For attribute (for)
· Used in a checkbox element while making a label for the checkbox

(xxvi) Declaration (<!DOCTYPE html>)
· Ensures the browser tries to meet industry-wide specifications

(xxvii) Style
· To set a specific appearance on an element perform the following command
Eg.
```
    <style>
        Element{property: value}
```
· It is possible to create a styles.css file where you can edit the style.
· Here style opening & closing tags are not required
· In order for the page to access the style file, it is necessary to add a (link) element with the rel attribute = stylesheet and the href=styles.c

* To make the page look similar on pc & mobile the following meta element is necessary
<meta name="viewport" content="width=device wid initial scale=1.0" />

xxviii) Div
· For design layout purposes.
· It is possible to set a specific width for the div element.
* You can use an id selector to target a specific element with an id attribute by placing # infront of the element's id value.

xxix) Commenting in CSS
eg. /* comment here */
* Once a colour has been ~~instructed~~ or any other element to ~~make~~ it function in the selected ~~Div~~ state (element as instructed)

xxx) Class selector
· Defined by a name with a dot directly infront of it.
eg .menu
· Instead of using and id attribute it makes use of a class attribute.

(xxxi) Article element (article)

(xxxii) Making items in line
* Add a class attribute to the article element with the value item.
· Using the class selector in css add a display property with the value inline-block
· When they move back towards each other add a width attribute to each class selector with an appropriate percentage.
· It is more appropriate to make the <p> element of the two values on the same line to make it possible to have even spacing without having to try various percentages.

xxxiii) Padding
• Gives space between content and sides
Eg padding-left: 20px

xxxiv) Max width property
• Creates a width that cannot be exceeded regardless of the width of the screen

xxxv) Font family (font-family)
• Used to select a common font for the text on the page
• sans-serif is a fairly common font that is very readable. other fonts (Impact)
* Incase one font is unavailable it is possible to create a fallback font. It is seperated by a comma

xxxvi) HR element (hr)
• Used to display a divider btw sections of diff. content.
* Self closing
* To make a link change colour after selection make use of a pseudo-selector
Eg a: visited { property Name: property Value; }

xxxvii) Encoding characters
• Set charset to utf-8 (a universal character set that includes almost every character from all human languages

xxxviii) META elements (meta)
• Self closing and give instructions to utilities to sections to be used in the page
• Each meta adds info about the page that cannot be expressed by other HTML elements

xxxix) Link element
· Makes ~~an~~ a CSS styleshit available with the rel attribute set to stylesheet and the href attribut set to styles.css

~~x~~ xL) Making seperate classes
· While making a class two values can be placed such that when the second class is selected in css it overides the first function

xLi) Making colours using (rgb)
· In the background-colour property use the value rgb(0,0,0) for black and vary the values of red, green and blue to get desired shades.

xLii) Making colours using hex values
· Include # then the hexadecimal or alphabetical values $_{\wedge}^{A-F}$ to generate colors
· In this format 00 is 0% of the colour and FF 100%.

0 1 2 3 4 5 6 7 8 9 A B C D E F

xLiii) HSL color model
* Hue, Saturation, lightness is another way to seperate colours
* CSS hsl function accepts 3 values: a no fro 0 to 360 for hue, a percentage from 0-100 for saturation and a percentage from 0-10 for lightens
* You must add % to saturation and lightnes values

XLiv) Gradient
. Used to make a color transition.
· The CSS linear gradient function allows you to control the direction of the transition along a line $ the colors used.
* The gradient makes an image $ is actually paired with the background property which can accept an image as a value.
* The linear-gradient function is very flexible

Eg linear-gradient (gradient Direction, color1, color 2, ...)
· Gradient direction is the direction of the line used for the transition. colo~~ur~~ 1 $ colour 2 are colo~~ur~~ arguments which will be used in the transition. These can be any type of colours including color keywords, hex, rgb or hsl
· A percentage can be added to the color to det. the space it will take.
· Incase you want the colour gradient in 180 deg it is not necessary to include it in the direction since it is the default.
· It is also not necessary to include color percentages since it is automatically calculated

XLV) Opacity
· How opaque something is
· 0% - completely transparent, 1.0 or 100% - completely opaque
~~* Channel~~ Alphachannel ⇒ similar to opacity
can be assigned by * Must have % in rgba
(i) rgb a
(ii) #RRGGBBAA
(iii) hsla

xLvi) Border-left
Using the property border-left; width style color; it is possible to set them all in one command.

xLvii) Box-shadow
. Lets you apply one or more shadows around an element. Here is basic syntax:
box-shadow: offset X offset Y color;
. Here's how the offset X and offset Y values work:
i) both accept no. values in px & other css units
ii) a positive value moves the shadow to the right.
iii) a negative value moves the shadow to the left
iv) It is not necessary to put a ~~[illegible]~~ value for any when it is 0 since all browsers understand 0 means no change
x Spread Radius (spreadRadius) used to spread the reach of the shadow.
x Blur Radius(blurRadius) creates a blur on the shadow. If px is not included it assumed to be 0 it is placed b4 color.

xLviii) Viewpoint height (vh)
. Relative to 1% of the height of the viewpoint

xLix) Method attribute (method)
. Specifies how to send form-data to the specified URL. It can be sent via a GET request with method="get" or via a POST request as data in the data in the request body(wit
method="post")

L ~~xx~~) Root em (rem)unit
× Seperates labels
× A label element is selected in the css sheet with display set to block and margin to 0.5 rem o

L i) Inpute
.It is a self closing tag with a for attribute and an id attribute containing the name of the name of the element. The for attribute is placed in the label element.
· Requires a type ~~&~~ id or value ~~&~~ attribute if no space is required.

L ii) Fieldset
· Creates a region to fill with data sort of a grid

L iii) Label
· The for attribute is placed here

L iv) Type
: Specifies the input
→ text - for text
→ email - for emails
→ password → obscures or gives a warning

L v) MinLength attribute (minlength)
. Creates a min length for an input.

L vi) Pattern attribute (pattern)
· Defines a regular expression that the password must match to be considered valid.
Eg [a-z 0-5 {8,}]

Lvii) Select element (select)
It is a container for a group of option elements. Each option element is a label for each dropdown option

Lviii) Text area (textarea)
* Requires a closing tag
⇒ Acts as a text type input with the benefit of adding multiline input
⇒ To modify the size of the text area add specifying rows and cols attribute

Lix) Placeholder
· Accepts a text value which is displayed until the user starts typing.
* It is useful and good practice to give every submitable element a name attribute.

Lx) Class of inline
⇒ Alligns the info, placed in the input element

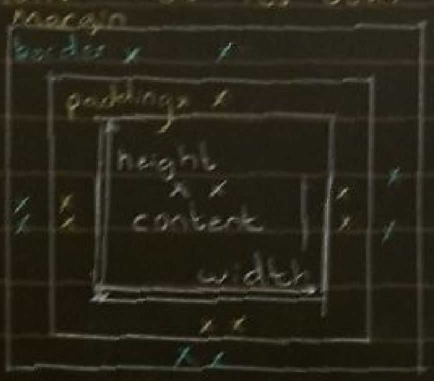
Lxi) Introducing a border
⇒ To introduce a border make use of a border property in the order w, t & colour

Lxii) Attribute selector
⇒ Make use of the following syntax
input [type=" value"] {
Types of inputs
* Checkbox
* Radio buttons

Lxiii) Adding a background image
Use the syntax background-image: url (link);
* To make it fill the whole screen use the property background-size and set it to cover

Box Model

- Every element has its own spacing & a border



- The content is like an ordered good that has been boxed. (Padding is like bubble wrap and the border is like the cardboard)
- When spacing margins or padding it is possible to space vertically and horizontally by use of the syntax margin: $\overset{\text{vertical}}{\text{value}}$ $\overset{\text{horizontal}}{\text{value}}$;
- Giving surrounding objects padding gives the enclosed something to push off of lest it seems to go into the surrounding object

- Overflow: hidden; - sets width back to original dimensions while giving an enclosed object something to push off of.
- * heights can also be expressed as percentages

- Blur - to blur add the property filter: blur(value px)
- * Border radius used to make corners smoother using px values in the order $\overset{1}{\text{top}}$-left, $\overset{3}{\text{bottom}}$-right, $\overset{2}{\text{top-right}}$ $\overset{4}{\text{bottom-left}}$.
  can use one value
  can use one value

Rotate to rotate add the property transform: rotate(value)
* To rotate counter clockwise use a -ve value.

CSS Flexbox

·In order to make content fit inside a set box use the universal selector (*) and set the box-sizing property to border-box

·It is possible to turn everything uppercase by use of the property text-transform and the value uppercase.

→ Flexbox is ~~a~~ one dimensional CSS layout that can control the way items are spaced out and alligned within a container.

To use it:

·Give an element a display property of flex

→ Flexbox has a main and cross axis. The main axis is defined by the flex-direction property which has four possible values.

→ Row

→ Row-reverse

→ Column

→ column-reverse

→ flex-wrap determines how items behave when the container is too small. setting it to wrap allows items to move to the nxt column or row while nowrap prevents this from happenin even shrinking items if needed.

→ justify-content determines how items insid the flex container are positioned in the main axis

→ align ~~a~~ items. positions flex items along the cross axis.

→ Object fit - det the aspect ratio to be used. T retain the aspect ration of each item set it to cover.

→ gap - sets gaps btw items. Can be row-gap, column-g or just gap as aproperty and a px value is set.

::after pseudo element creates an element that is the last child of the selected element.

Typography

· The art of styling your text to be easily readable and suit its purpose.
· To import the Open Sans font family with weight values 400, 700 and 800 use the value of the href attribute as
 https://fonts.googleapis.com/css?family=Open+Sans:400,700,800
* To set a fallback font-family seperate it from the main font-family by use of a comma
* Use the border property to add borders.
* Make sure to set box-sizing to border box for the whole page using the universal selector.
* letter-spacing - can be used to adjust the space btw each character of ~~t~~ text in an element.

* The :not pseudo selector can be used to select all elements that do not mat the CSS rule given.
<i> element - used to italicize

3-01-024

Accessibility

· Making a webpage easy for people to use

Meta elements
· Gives details on the page
·

Examples of meta ~~attributes~~ definitions: attributes:

(i) charset - gives the character encoding

(ii) name = gives the type of definition to be run
~~viewport~~ tells the browser how to rende
def viewport
the page (attribute → content)

SEO - Search Engine Optimization

* Contents of an attribute are called definitions

iii) content - explains what the browser should
do and what the reader should know

* We make use of semantic HTML elements
to provide the structure of your page

Navigation

→ header element - used to introduce a page a
well as provide a navigation menu.

→ main element - contains core content of your
page

→ nav element

→ To enable navigation ad an unordered list with
items inside the <li> element anchored.

Scalable vector graphics (SVG)

→ Contains a path attribute that allows an imag
to be scaled without affecting the resoluti
of the resultant image.

To add a max width use the syntax

```
#id/.class {  width: max(value px, value vw);
          }
```

The child combinator selector [>] is used btw

Examples of meta ~~definitions~~ ~~attributes~~: attributes:

(i) Charset -gives the character encoding

(ii) ~~viewport~~ name -gives the type of definition to be run
det -viewport -tells the browser how to render the page (attribute → content)

SEO -search Engine Optimization.

* Contents of an attribute are called definitions

iii) content - explains what the browser should do and what the reader should know

* We make use of semantic HTML elements to provide the structure of your page

Navigation

→ header element -used to introduce a page as well as provide a navigation menu.

→ main element -contains core content of your page

→ nav element

→ To enable navigation ad an unordered list with the items inside the <li> element anchored.

Scalable vector graphics (SVG)

→ Contains a path attribute that allows an image to be scaled without affecting the resolution of the resultant image.

To add a max width use the syntax

```
#id/.class {
             width: max(value px, value vw);
           }
```

The child combinator selector $\boxed{>}$ is used btw

selectors to target only elements that match the second selector or are a direct child of the first selector.
. This can help when you have deeply nested elements and want to control the scope of your styling.

To make a form submit to a certain url use the syntax; method="post" action=" url " inside the form element.

The role attribute
→ it is part of the Web Accessibility Initiative(WA
and accepts preset values.
→ it can be used to indicate the purpose behind an element on the page to assistive technologies
* You can make use of a region role
→ Each region role requires a label. This can be done by use of the [aria-labelledby] attribut

CSS Extra1

- CSS positioning lets you set how you want an element to be positioned in the browser. It has a position property you can set to:

1. static .The default positioning for all elements. If you assign it to an element, you won't be able to move it around with top, right, left, or bottom.
2. relative .The element is still positioned according to the normal flow of the document, but the top, left, bottom, and right values become active.
3. absolute .When you use the absolute value for your position property, the element is taken out of the normal flow of the document, and then its position is determined by the top, right, bottom, and left properties.
4. fixed .A position property value that lets you make an element fixed to the page no matter where the user scrolls to on the page.You'll have to do some more markups to see how fixed positioning works.
5. sticky .It is a hybrid of relative and fixed positioning. It allows an element to **stick** to a specific position within its containing element or viewport, based on the scroll position.

Note: To see how sticky works, you have to place a couple of texts before and after your .cat-head div element. If you scroll up after that, you'll see that the .cat-head gets stuck to the top and remains there.

- Once you set the position property of the element, you can move the element around by setting a pixel or a percentage value for one or more of the top, right, left, or bottom properties.

NAVIGATION

- Two other semantic HTML elements are the footer and address elements. The footer element is a container for a collection of content that is related to the page, and the address element is a container for contact information for the author of the page.

Address Element.

- An example of an address element syntax is as shown below:

```
freeCodeCamp<br />
San Francisco<br />
California<br />
USA
```

- The br tags will allow each part of the address to be on its own line and are useful for presenting address elements properly.

MODDIFYING CURSOR

- To modify the hover color and cursor use the syntax below or similar;

```
nav>ul>li:hover{
  background-color: #dfdfe2;
  color: #1b1b32;
  cursor: pointer;
}
```

PREVENTING OVERFLOW

- To make content not to overflow regardless of the size of the screen use the following syntax on the targeted element:

```
***{
        overflow:hidden;
}
```

- On small screens, the unordered list in the navigation bar overflows the right side of the screen.Fix this by using Flexbox to wrap the ul content. Then, set the following CSS properties to correctly align the text:

```
align-items: center; padding-inline-start: 0; margin-block: 0; height: 100%;
* flex-wrap of wrap.
```

CHANGING SCROLL BEHAVIOUR

- Clicking on the navigation links should jump the viewport to the relevant section. However, this jumping can be disorienting for some users.
- Select all elements, and set the scroll-behavior to smooth.
- Certain types of motion-based animations can cause discomfort for some users. In particular, people with vestibular disorders have sensitivity to certain motion triggers.
- The @media at-rule has a media feature called prefers-reduced-motion to set CSS based on the user's preferences. It can take one of the following values:

1) reduce
2) no-preference

- It normally makes use of the following syntax;

```
@media (feature: value) {
  selector {
    styles
  }
}
```

CREATING KEYBOARD SHORTCUTS

- Finally, the navigation accessibility can be improved by providing keyboard shortcuts.
- The accesskey attribute accepts a space-separated list of access keys. For example:

```
<button type="submit" accesskey="s">Submit</button>
```

- Give each of the navigation links a single-letter access key.

Note: It is not always advised to use access keys, but they can be useful

MAKING TABLES IN CSS

- The thead and tbody elements are used to indicate which portion of your table is the header, and which portion contains the primary data or content.
- The tr element is used to indicate a table row.
- The td element indicates a data cell, while the th element indicates a header cell.
- Leave the td element empty. This element exists only to ensure your table has a four-column layout and associate the headers with the correct columns.
- To hide elements it is possible to make use of the overflow property set to hidden.
- It is possible to create a table with varying widths using the row and column span attribute.
-

PSEUDO SELECTORS

- The span[class~="sr-only"] selector will select any span element whose class includes sr-only

*The CSS clip property is used to define the visible portions of an element. The clip-path property determines the shape the clip property should take.

- The :first-of-type pseudo-selector is used to target the first element that matches the selector.Remember that your span elements are reversed, visually, so this will appear to be the second element on the page.

*The calc() function is a CSS function that allows you to calculate a value based on other values. For example, you can use it to calculate the width of the viewport minus the margin of an element:

```
.example {
  margin: 10px;
  width: calc(100% - 20px);
}
```

- The :not() pseudo-selector is used to target all elements that do not match the selector - in this case, any of your span elements that do not have the sr-only class. This ensures that your earlier rules for the span[class~="sr-only"] selector are not overwritten.

Rather than having to constantly double-check you are not overwriting your earlier properties, you can use the important keyword to ensure these properties are always applied, regardless of order or specificity.
 *Setting the border-collapse property to collapse allows cell borders to collapse into a single border, instead of a border around each cell.

- The [attribute="value"] selector targets any element that has an attribute with a specific value.
- The :nth-of-type() pseudo-selector is used to target specific elements based on their order among siblings of the same type.

*The key difference between tr[class="total"] and tr.total is that the first will select tr elements where the only class is total. The second will select tr elements where the class includes total.

- The ::first-letter pseudo-selector allows you to target the first letter in the text content of an element.

RESPONSIVE WEBPAGE DESIGN

- Responsive Design tells your webpage how it should look on different-sized screens.
- Browsers can apply default margin and padding values to specific elements. To make sure your piano looks correct, you need to reset the box model.
- Now that you have reset the html box model, you need to pass that on to the elements within as well. To do this, you can set the box-sizing property to inherit, which will tell the targeted elements to use the same value as the parent element.
- You will also need to target the pseudo-elements, which are special keywords that follow a selector. The two pseudo-elements you will be using are the ::before and ::after pseudo-elements.
- The ::before selector creates a pseudo-element which is the first child of the selected element, while the ::after selector creates a pseudo-element which is the last child of the selected element. These pseudo-elements are often used to create cosmetic content, which you will see later in this project.
- The float property can be used to separate two or more visual elements that have the same class. This can be

done by setting it to either left or right.

- The content property can be used to determine the content within a particular selected class,id or element.
- The @media at-rule, also known as a media query, is used to conditionally apply CSS. Media queries are commonly used to apply CSS based on the viewport width using the max-width and min-width properties.
- In the below example the padding is applied to the .card class when the viewport is 960px wide and below.

```
@media (max-width: 960px) {
 .card { padding: 2rem; }
 }
```

- Logical operators can be used to construct more complex media queries. The and logical operator is used to query two media conditions.

For example, a media query that targets a display width between 500px and 1000px would be:

```
@media (min-width: 500px) and (max-width: 1000px){ }
```

CSS VARIABLES

- In CSS, you can target everything with an asterisk. Add a border to everything by using the * selector, and giving it a border of 1px solid black. This is a trick that helps visualize where elements are and their size. You will remove this later.
- Also add a box-sizing of border-box to everything. This will make it so the border you added doesn't add any size to your elements.
- To use a variable, put the variable name in parentheses with var in front of them like this: var(--variable-name). Whatever value you gave the variable will be applied to whatever property you use it on.
- When a setting a variable it is possible to give it a fallback value by adding a comma to the variable-name inside the var, spacing then adding a new value.
- Variables are normally placed in the :root selector. This makes them usable everywhere. The :root selector is the highest selector.
- Your code is starting to get quite long. Add a comment above the .fb1 class that says FOREGROUND BUILDINGS - "fb" stands for "foreground building" to help people understand your code. Above the .bb1 class add another comment that says BACKGROUND BUILDINGS - "bb" stands for "background building". If you don't remember, comments in CSS look like this: /* Comment here */.
- Gradients in CSS are a way to transition between colors across the distance of an element. They are applied to the background property and the syntax looks like this:
  `gradient-type( color1, color2 );`
- In the example, color1 is solid at the top, color2 is solid at the bottom, and in between it transitions evenly from one to the next.
- You can specify where you want a gradient transition to complete by adding it to the color like this:
  `gradient-type( color1, color2 20%, color3 );`
- Here, it will transition from color1 to color2 between 0% and 20% of the element and then transition to color3 for the rest.
- Gradient transitions often gradually change from one color to another. You can make the change a solid line like this:
  `linear-gradient( var(--first-color) 0%, var(--first-color) 40%, var(--second-color) 40%, var(--second-color) 80% );`
- Changing the gradient type from linear-gradient to repeating-linear-gradient makes the four colors of your gradient repeat until it gets to the bottom of the element; giving you some stripes, and saving you from having to add a bunch of elements to create them.

```
margin: auto; width: 5vw; height: 5vw; border-top: 1vw solid #000;
border-bottom: 1vw solid #000; border-left: 1vw solid #999;
border-right: 1vw solid #999;
```

- The above properties create a container like structure.
- So far, all the gradients you created have gone from top to bottom, that's the default direction. You can specify another direction by adding it before your colors like this:
  `gradient-type( direction, color1, color2 );`
- If you want to add similar properties to different elements you can add a similar class to the elements and put the properties there.
- Adding circle closest-corner at 15% 15%, will move the start of radial gradient to 15% from the top and left. It will make it end at the closest-corner and it will maintain a circle shape. These are some keywords built into gradients to describe how it behaves.
- A media query can be used to change styles based on certain conditions, and they look like this:
  `@media (condition) { }`

❖ Vh- viewport height.
❖ Vw- viewport width.

ANIMATION

- The transform-origin property is used to set the point around which a CSS transformation is applied. For example, when performing a rotate (which you will do later in this project), the transform-origin determines around which point the element is rotated.
- The @keyframes at-rule is used to define the flow of a CSS animation. Within the @keyframes rule, you can create selectors for specific points in the animation sequence, such as 0% or 25%, or use from and to to define the start and end of the sequence.
- @keyframes rules require a name to be assigned to them, which you use in other rules to reference. For example, the @keyframes freeCodeCamp { } rule would be named freeCodeCamp.
- The animation-name property is used to link a @keyframes rule to a CSS selector. The value of this property should match the name of the @keyframes rule. Give your .wheel selector an animation-name property set to wheel.
- The animation-duration property is used to set how long the animation should sequence to complete. The time should be specified in either seconds (s) or milliseconds (ms).
- The animation-iteration-count property sets how many times your animation should repeat. This can be set to a number, or to infinite to indefinitely repeat the animation.
- The animation-timing-function property sets how the animation should progress over time. There are a few different values for this property;

linear	The animation has the same speed from start to end
ease	Default value. The animation has a slow start, then fast, before it ends slowly
ease-in	The animation has a slow start
ease-out	The animation has a slow end

ease-in-out	The animation has both a slow start and a slow end
step-start	Equivalent to steps(1, start)
step-end	Equivalent to steps(1, end)
cubic-bezier(*n*,*n*,*n*,*n*)	Define your own values in the cubic-bezier function Possible values are numeric values from 0 to 1
initial	Sets this property to its default value. Read about *initial*
inherit	Inherits this property from its parent element

- To make your cabin animation seem more like a natural swinging motion, you can use the ease-in-out timing function. This setting will tell the animation to start and end at a slower pace, but move more quickly in the middle of the cycle.

CSS GRID

- The loading attribute on an img element can be set to lazy to tell the browser not to fetch the image resource until it is needed (as in, when the user scrolls the image into view). As an additional benefit, lazy loaded elements will not load until the non-lazy elements are loaded - this means users with slow internet connections can view the content of your page without having to wait for the images to load.
- The Referer HTTP header contains information about the address or URL of a page that a user might be visiting from. This information can be used in analytics to track how many users from your page visit freecodecamp.org, for example. Setting the rel attribute to noreferrer omits this information from the HTTP request.
- The i element is used to insert icons in the webpage.
- Now you are ready to start putting together the grid layout. CSS Grid offers a two-dimensional grid-based layout, allowing you to center items horizontally and vertically while still retaining control to do things like overlap elements.
- Begin by creating a main selector and giving it a display property set to grid.
- Now you can style the layout of your grid. CSS Grid is similar to Flexbox in that it has a special property for both the parent and child elements.
- In this case, your parent element is the main element. Set the content to have a three-column layout by adding a grid-template-columns property with a value of 1fr 94rem 1fr. This will create three columns where the middle column is 94rem wide, and the first and last columns are both 1 fraction of the remaining space in the grid container.
- Use the minmax function to make your columns responsive on any device. The minmax function takes two arguments, the first being the minimum value and the second being the maximum. These values could be a length, percentage, fr, or even a keyword like max-content.
- Your magazine will have three primary sections. You already set the overall layout in the main rule, but you can adjust the placement in the child rules.
- One option is the grid-column property, which is shorthand for grid-column-start and grid-column-end. The grid-column property tells the grid item which grid line to start and end at.
- The CSS repeat() function is used to repeat a value, rather than writing it out manually, and is helpful for grid layouts. For example, setting the grid-template-columns property to repeat(20, 200px) would create 20 columns each 200px wide.
- Remember that the grid-column property determines which columns an element starts and ends at. There may be times where you are unsure of how many columns your grid will have, but you want an element to stop at the last column. To do this, you can use -1 for the end column.
- The object-fit property tells the browser how to position the element within its container. In this case, cover will set the image to fill the container, cropping as needed to avoid changing the aspect ratio.
- The default settings for CSS Grid will create additional rows as needed, unlike Flexbox.
- If you wanted to add more social icons, but keep them on the same row, you would need to update grid-template-columns to create additional columns. As an alternative, you can use the grid-auto-flow property.
- This property takes either row or column as the first value, with an optional second value of dense. grid-auto-flow uses an auto-placement algorithm to adjust the grid layout. Setting it to column will tell the algorithm to create new columns for content as needed. The dense value allows the algorithm to backtrack and fill holes in the grid with smaller items, which can result in items appearing out of order.
- Now the auto-placement algorithm will kick in when you add a new icon element. However, the algorithm defaults the new column width to be auto, which will not match your current columns.
- You can override this with the grid-auto-columns property.
- Much like Flexbox, with CSS Grid you can align the content of grid items with align-items and justify-items. align-items will align child elements along the column axis, and justify-items will align child elements along the row axis.
- Much like Flexbox, with CSS Grid you can align the content of grid items with align-items and justify-items. align-items will align child elements along the column axis, and justify-items will align child elements along the row axis.
- Magazines often use justified text in their printed content to structure their layout and control the flow of their content. While this works in printed form, justified text on websites can be an accessibility concern, for

example presenting challenges for folks with dyslexia.

- The place-items property can be used to set the align-items and justify-items values at the same time. The place-items property takes one or two values. If one value is provided, it is used for both the align-items and justify-items properties. If two values are provided, the first value is used for the align-items property and the second value is used for the justify-items property.

- How to use style float in HTML?

The float property can have one of the following values:

left - The element floats to the left of its container.
right - The element floats to the right of its container.
none - The element does not float (will be displayed just where it occurs in the text). ...
inherit - The element inherits the float value of its parent.

TRANSFORMS

- skew transform function takes two arguments. The first being an angle to shear the x-axis by, and the second being an angle to shear the y-axis by.
- The z-index property is used to move objects in front or behind other objects.
- To invert an axis use the following syntax;
  scale(axis): -1

NOTES

- Comments are messages left in the line of code to leave some certain necessary information which may be helpful in the future.
- Some styling can be done outside of CSS using similar coding syntax.

FRAMES

- This provides separate navigation bars to use if their is some information you always want to keep in view.