LEARN INTERMEDIATE CSS BY BUILDING A CAT PAINTING

Introduction:

Mastering CSS positioning is essential for creating visually appealing and responsive web layouts

In this course, you will build a cat painting. You'll learn about how to work with absolute positioning, the z-index property, and the transform property.

Step 1:

Begin with the basic HTML structure. Add a `DOCTYPE` reference of `html` and an `html` element with its `lang` attribute set to `en`. Also, add a `head` and a `body` element within the `html` element.

Step 2:

Within your `head` element, add a `meta` tag with the `charset` attribute of `utf-8`. Also add a `title` element with the text `fCC Cat Painting`.

Step 3:

Add a `link` element within your `head` element. For that `link` element, set the `rel` attribute to `stylesheet` and the `href` to `./styles.css`.

Step 4:

Use the universal selector to add `box-sizing: border-box;` to your CSS. This ensures elements include padding and border in their specified width and height.

Step 5:

Give your `body` element a `background-color` of `#c9d2fc`.

Step 6:

Back in your HTML, create a `main` element. Inside that `main` element, add a `div` element with the class `cat-head`.

Step 7:

Using a class selector, give the `.cat-head` element a width of `205px` and a height of `180px`. Also, give it a border of `1px solid #000` and a `border-radius` of `46%`.

Step 8:

To see the `.cat-head` element, give it a linear gradient background with `#5e5e5e` at `85%` and `#45454f` at `100%`.

You might not notice the difference between these two colors, but they are there.

Step 9:

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 `static`, `absolute`, `relative`, `sticky` or `fixed`.

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.

`static` is 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`.

Give `.cat-head` a `position` property of `static`, then set the `top` and `left` properties to `100px` each.

Step 10:

You could see that nothing happens. The `.cat-head` element did not move despite setting a `top` and `left` of `100px` each. But that's not the case with `relative` positioning.

When you use the `relative` value, the element is still positioned according to the normal flow of the document, but the `top`, `left`, `bottom`, and `right` values become active.

Instead of `static`, give your `.cat-head` a position of `relative`, and leave both `top` and `left` properties as they are.

Step 11:

The next position property is `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.

Set the `position` property of your `.cat-head` element to `absolute`, then set `top` and `left` properties to any positive pixel value.

Step 12:

`fixed` is 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. In your HTML, create a `div` element with the class `box`.

Step 13:

Use a class selector to give your `.box` element a width of `200px`, a height of `600px`, and a background color of `#000`. Also, give it a `position` of `absolute`, a `top` of `800px` and a `left` of `650px`.

Step 14:

Now replace the `position` property value of your `.cat-head` with `fixed`. Leave both `top` and `left` as they are.

After that, scroll up and down to see how the `fixed` value works.

Step 15:

The last position property value is `sticky`. `sticky` positioning 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.

Change the value of the `position` property of `.cat-head` to `sticky`, set `top` to `0`, then remove `left` and its value.

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 down after that, you'll see that the `.cat-head` gets stuck to the top and remains there.

Step 16:

You should now center the cat head.

Give the .cat-head element a position property set to absolute. Set a value of 0 for the right, left, top, bottom properties, then set its margin property on all sides to auto. That's one way to center an element vertically and horizontally using CSS positioning.

Step 17:

Remove the div element with class box because you don't need it anymore.

Step 18:

Also, remove the .box CSS rule and its declarations because you don't need them anymore.

Step 19:

Now you should work on the cat's ears. There will be a right and a left ear, and inside each, there will be an inner ear.

Inside your .cat-head element, create a div element with the class cat-ears.

Step 20:

Inside your .cat-ears element, create two div elements with the classes cat-left-ear and cat-right-ear respectively.

Step 21:

Create two `div` elements, the first inside the `.cat-left-ear` element with a class of `cat-left-inner-ear`, and the second inside the `.cat-right-ear` element with a class of `cat-right-inner-ear`.

Step 22:

Now you will learn a CSS trick to draw triangles.

This will be used to draw ears on the cat.

Using a class selector, give the `.cat-right-ear` element `height` and `width` properties set to `100px`. Set the `background-color` to `white`. Set the left and right borders to `35px solid blue`. Set the top and bottom borders to `35px solid red`.

Step 23:

Notice that you now have a white square with thick borders. The borders have diagonal edges which can be used to create triangles.

Within the same class selector adjust `height` and `width` to `0`. Set the left, right and top border to `transparent`.

Remove the `background-color` property, and you should see a triangle.

Step 24:

Now you can begin working on your cat's ears.

Clean up your review code by removing the `.cat-right-ear` selector and all of its properties.

Using a class selector, give the `.cat-left-ear` element a left and right border of `35px solid transparent` each. Also, set the bottom border to `70px solid #5e5e5e`.

Step 25:

Move the left ear into position by setting a position of `absolute`, a `top` of `-26px`, and a `left` of `-31px`.

Step 26:

Those edges are too sharp for an ear. So, set the `border-top-left-radius` to `90px` and the `border-top-right-radius` to `10px`.

Step 27:

To position the left ear properly, you can use CSS transform to rotate it in a certain degree.

The `transform` property allows you to modify the shape, position, and size of an element without changing the layout or affecting the surrounding elements. It has functions such as `translate()`, `rotate()`, `scale()`, `skew()`, and `matrix()`.

Set the `transform` property to `rotate(-45deg)` and see what happens.

Step 28:

Now you can work on the right ear of the cat. You have the HTML for it already.

Using a class selector, give the .cat-right-ear element a left and right border of 35px solid transparent each. Also, set the bottom border to 70px solid #5e5e5e.

Step 29:

Move the right ear into position with a position property set to absolute, a top of -26px, and a left of 163px.

Step 30:

As you did for the left ear, rotate the right ear at 45 degrees.

Step 31:

Remove the sharp border of the right ear by setting the border-top-left-radius to 90px and the border-top-right-radius to 10px.

Step 32:

The ears should always be placed above the part of the head it overlaps. You can do this with the z-index property.

z-index is a property you can use to define the order of overlapping HTML elements. Any element with a higher z-index will always be positioned over an element with a lower z-index.

To see z-index in action, set the z-index property of the left ear to -1.

Step 33:

That's not the behavior you want. You should make the ears display over the head so the borders that overlap with them don't show.

Instead of -1, set the z-index property of the left ear to 1.

Step 34:

Set the z-index property of the right ear to 1 so it always stays over the head.

Step 35:

Most cats have different colors in their ear and the inner part of the same ear. You can implement the same too. That's why you defined a div element for both right and left inner ears a while ago.

Using a class selector, give your .cat-left-inner-ear element a left and right border of 20px solid transparent each. Also give it a bottom border of 40px solid #3b3b4f.

Step 36:

Move the inner ear into position with a position property set to absolute, a top of 22px, and a left of -20px.

Step 37:

To remove all the pointed edges of the ear, set a bottom-right and bottom-left border radius of 40% each, a top-left border radius of 90px, and a top-right border radius of 10px.

Step 38:

It's time to work on the right inner ear. Using a class selector, give your .cat-right-inner-ear element a left and right border of 20px solid transparent. Also, give it a bottom border of 40px solid #3b3b4f.

Step 39:

Move the right inner ear into position with a position property set to absolute, a top of 22px and a left of -20px.

Step 40:

As you did for the left inner ear, remove the sharp edges of the right inner ear by setting a bottom-right and bottom-left border radius of 40%, a top-left border radius of 90px, and a top-right border radius of 10px.

Step 41:

You will now start working on the cat's eyes. Like the ears, the eyes will have inner eyes.

Create a div element with the class cat-eyes. Inside the .cat-eyes element, create two div elements with the class cat-left-eye and cat-right-eye respectively.

Step 42:

Create two div elements, one with the class cat-left-inner-eye inside the .cat-left-eye element and another with the class cat-right-inner-eye inside the .cat-right-eye element.

Step 43:

Using a class selector, give your `.cat-left-eye` element a `width` of `30px` and a `height` of `40px`. Also, give it a `background-color` of `#000`.

Step 44:

Move the left eye into position with a `position` property of `absolute`, a `top` of `54px`, and a `left` of `39px`.

Step 45:

To make the left eye look like an eye, give it a border radius of `60%`. Also, using the `transform` property, rotate it at `25` degrees.

Step 46:

Now you will work on the right eye by using the same approach.

Using a class selector, give your `.cat-right-eye` element a width of `30px` and a height of `40px`. Also, give it a background color of `#000`.

Step 47:

Move the right eye into position with a `position` property of `absolute`, a `top` of `54px`, and a `left` of `134px`.

Step 48:

To make the right eye look like an eye, give it a border radius of `60%`. Also, using the `transform` property, rotate it at `-25` degrees.

Step 49:

Those look like eyes, but you can still make them better. That's why you created two inner eyes div elements.

Using a class selector, give your .cat-left-inner-eye element a width of 10px and a height of 20px. Also, give it a background color of #fff.

Step 50:

Move the left inner eye into position with a position property of absolute, a top of 8px, and a left of 2px. Also, give it a border radius of 60% and rotate it at 10 degrees.

Step 51:

Using a class selector, give your .cat-right-inner-eye element a width of 10px and a height of 20px. Also, give it a background color of #fff.

Step 52:

Move the right inner eye into position with a position of absolute, a top of 8px, and a left of 18px. Also, give it a border radius of 60% and rotate it at -5deg.

Step 53:

It's time to work on the nose. In your HTML, create a div element with the class cat-nose.

Step 54:

Using a class selector, give your .cat-nose element a left and right border of 15px solid transparent each. Also give it a bottom border of 20px solid #442c2c.

Step 55:

Move the nose into position with a position property of absolute, a top of 108px, and a left of 85px.

Step 56:

Remove the sharp edges of the nose with border radius of 50% each on the top-left, bottom-right, and bottom-left corners. Also, rotate it at 180 degrees.

Step 57:

Now you will start working on the mouth. There will be a right line and left line for the mouth.

Create a div element with the class cat-mouth.

Step 58:

Inside your .cat-mouth element, create a div element with the class cat-mouth-line-left and another div with the class cat-mouth-line-right.

Step 59:

Using a descendant selector, select the two div elements inside the div with class cat-mouth. Give it a width of 30px, a height of 50px, and a border of 2px solid #000.

Step 60:

You are going to make the two mouth lines into an elliptical shape. So, give the .cat-mouth div selector a border color of black transparent transparent transparent and a border radius of 190%/190px 150px 0 0.

Step 61:

Using a class selector, give your .cat-mouth-line-left element a position of absolute, a top of 88px and a left of 74px. This would move it into the right position.

Step 62:

Using the transform property, rotate the left mouth line at 170 degrees.

Step 63:

Access your .cat-mouth-line-right element with a class selector, then move it into the right position with a position of absolute, a top of 88px and a left of 91px.

Step 64:

Rotate the right mouth line at `165` degrees.

Step 65:

The last thing you will work on is the whiskers. There are going to be 6 of them, meaning there will be three on each side.

Create a `div` element with the class `cat-whiskers`.

Step 66:

Inside the `.cat-whiskers` element, create two `div` elements with the class `cat-whiskers-left` and `cat-whiskers-right`.

Step 67:

Inside the `.cat-whiskers-left` element, create three `div` elements with the classes `cat-whisker-left-top`, `cat-whisker-left-middle`, and `cat-whisker-left-bottom`.

Step 68:

Inside the `.cat-whiskers-right` element, create 3 `div` elements with the class `cat-whisker-right-top`, `cat-whisker-right-middle`, and `cat-whisker-right-bottom`.

Step 69:

Use a descendant selector to target the three `div` elements inside your `.cat-whiskers-left` element. Give it a `width` of `40px`, a `height` of `1px`, and a `background-color` of `#000`.

Step 70:

As you did in the previous step, use a descendant selector to target the three div elements inside your .cat-whiskers-right element. Give it a width of 40px, a height of 1px, and a background-color of #000.

Step 71:

Using a class selector, move the .cat-whisker-left-top element into place with a position of absolute, a top of 120px, and a left of 52px.

Step 72:

Rotate the left top whisker at 10 degrees.

Step 73:

Use a class selector to target the .cat-whisker-left-middle element. Then move it into place with a position property set to absolute, a top of 127px, and a left of 52px.

Step 74:

Rotate the left middle whisker at 3 degrees.

Step 75:

Using a class selector, move the .cat-whisker-left-bottom into position with a position of absolute, a top of 134px, and a left of 52px.

Step 76:

Rotate the left bottom whisker at -3 degrees.

Step 77:

Now you will work on moving the right whiskers into place. Use class selector to target the .cat-whisker-right-top element and give it a position of absolute, a top of 120px, and a left of 109px.

Step 78:

Rotate the top-right whisker at -10 degrees.

Step 79:

Use a class selector to target the .cat-whisker-right-middle element, then move the right middle whisker into position with a position of absolute, a top of 127px, and a left of 109px.

Step 80:

Rotate the right middle whisker at -3 degrees.

Step 81:

Use class selector to target the .cat-whisker-right-bottom element, then move it into place with a position of absolute, a top of 134px, and a left of 109px.

Step 82:

Rotate the bottom-right whisker at 3 degrees.

With this final step, your cat painting is now complete.