**Step 2**

Within the head element, add a meta tag which sets the charset to UTF-8, and a title element with the value Rothko Painting.

Within the body element, add an img element with a src of https://cdn.freecodecamp.org/curriculum/css-box-model/diagram-1.png.

 <head>

    <meta charset="utf-8"/>

    <title>

      Rothko Painting

      </title>

  </head>

  <body>

    <img src="https://cdn.freecodecamp.org/curriculum/css-box-model/diagram-1.png"/>

  </body>

# Step 3

In the CSS box model, every HTML element is treated as a box with four areas.

Imagine you receive a box from your favorite online retailer -- the content is the item in the box, or in our case, a header, paragraph, or image element.

Change the src attribute in the <img> from https://cdn.freecodecamp.org/curriculum/css-box-model/diagram-1.png to <https://cdn.freecodecamp.org/curriculum/css-box-model/diagram-2.png>.

# Step 4

The content is surrounded by a space called padding, similar to how bubble wrap separates an item from the box around it.

Think of the border like the cardboard box your item was shipped in.

Change the src attribute to https://cdn.freecodecamp.org/curriculum/css-box-model/diagram-3.png

<img src="https://cdn.freecodecamp.org/curriculum/css-box-model/diagram-2.png">

   <img src="https://cdn.freecodecamp.org/curriculum/css-box-model/diagram-3.png">

# Step 5

Margin is the area outside of the box, and can be used to control the space between other boxes or elements.

Here the bottom element has a larger top margin, pushing it further down the page.

Now that you understand the CSS box model, let's get started on the Rothko painting.

Remove the <img> element.

# Step 6 Passed

Add a div element in the body.

Set the class attribute equal to canvas. For example, <div class="my-div">.

This will act as the canvas for your painting.

<div class=”canvas”></div>

# Step 7

Before you can start styling the div you added, you need to link your CSS to your HTML.

Add a link element to link your styles.css file. Set the href to styles.css, and remember to set the rel attribute to stylesheet.

# Step 8

Time for CSS.

Even though your <div> has no text, it's still treated as a box with content. Write a CSS rule that uses the .canvas class selector and set its width to 500 pixels. Here's a CSS rule that sets the width of the class card to 300 pixels:

.card {

width: 300px;

}

# Step 11

Every painting needs a frame.

Wrap the .canvas element in another div. Give that div the frame class.

<div class="frame">

    <div class="canvas">

    </div>

    </div>

# Step 12 Passed

Write a new rule using the .frame class selector.

Use the border shorthand declaration to give the .frame element a solid, black border with a width of 50px.

.frame{

  border:50px solid black;

}

# Step 13

The frame is much too wide.

In .frame, set its width to 500 pixels.

width:500px;

# Step 14

Use padding to adjust the spacing within an element.

In .frame, use the padding shorthand property to increase the space between the .frame and .canvas elements by 50px. The shorthand will increase space in the top, bottom, left, and right of the element's border and canvas within.

# Step 15

Use margins to adjust the spacing outside of an element.

Using the margin property, give the .frame element vertical margin of 20px, and horizontal margin of auto. This will move the frame down 20 pixels and horizontally center it on the page.

margin:20px auto;

# Step 16

Add a new div element inside of your .canvas element.

Give the new div the class attribute with a value of one. This will be your first rectangle.

<div class="one"></div>

# Step 17

Write a new rule that targets .one and set its width to 425 pixels.

# Step 18

Now set the height for .one to 150 pixels.

height:150px;

# Step 19

Set the background-color of .one to #efb762.

background-color:#efb762;

# Step 20

Use margins to position the .one element on the canvas.

Add the shorthand margin property with a vertical margin of 20px and a horizontal margin of auto.

margin:20px auto;

# Step 21

Now .one is centered horizontally, but its top margin is pushing past the canvas and onto the frame's border, shifting the entire canvas down 20 pixels.

Add padding of 1px to the .canvas element to give the .one element something solid to push off of.

padding:1px;

# Step 22

Adding 1 pixel of padding to the top, bottom, left, and right of the canvas changed its dimensions to 502 pixels x 602 pixels.

Replace the padding property with overflow set to hidden - changing the canvas back to its original dimensions.

 overflow:hidden;

use overflow instead of padding 1px

# Step 23

Add another div with a class value of two just below your one element. This will be your second rectangle.

<div class=”two” > </div>

# Step 24

Create a new CSS rule using the .two selector and set its width to 475 pixels.

.two {

  width: 475px;

}

# Step 25

Set the height of the .two element to 200 pixels.

height:200px;

background-color:#8f0401;

margin:auto;

# Step 28

Create a new div with a class value of three right under the .two element. This will be your third rectangle.

<div class="three"></div>

# Step 29

You don't always have to use pixels when sizing an element.

Create a new rule, .three, and set its width to 91%.

.three{

  width:91%;

}

# Step 30

Set the height of .three to 28%.

height:28%;

background-color:#b20403;

# Step 32

Center the .three element on the canvas by setting its margin to auto.

# Step 33

It's helpful to have your margins push in one direction.

In this case, the bottom margin of the .one element pushes .two down 20 pixels.

In the .two selector, use margin shorthand property to set top margin to 0, horizontal margin to auto, and bottom margin to 20px. This will remove its top margin, horizontally center it, and set its bottom margin to 20 pixels.

 margin: 0 auto 20px;

# Step 34

The colors and shapes of your painting are too sharp to pass as a Rothko.

Use the filter property to blur the painting by 2px in the .canvas element.

Here's an example of a rule that add a 3px blur:

p {

filter: blur(3px);

}

filter:blur(2px);

# Step 35

Create a rule that targets both .one and .two and increase their blur effect by 1 pixel.

.one , .two{

  filter:blur(1px);

}

# Step 36

Increase the blur of .three by 2 pixels.

filter:blur(2px);

# Step 37

The rectangles are too small and their edges don't have the soft quality of a painting.

Increase the area and soften the edges of .one by setting its box-shadow to 0 0 3px 3px #efb762.

box-shadow:0 0 3px 3px #efb762;

# Step 38

Use the same box-shadow declaration for .two, but change the color from #efb762 to #8f0401.

margin: 0 auto 20px;

  box-shadow:0 0 3px 3px #8f0401;

# Step 39

Add a box-shadow to .three with the values 0 0 5px 5px #b20403.

box-shadow:0 0 5px 5px #b20403;

# Step 40

The corners of each rectangle are still too sharp.

Round each corner of the .one element by 9 pixels, using the border-radius property.

border-radius:9px;

# Step 41

Use the border-radius property on the .two selector, to set its top-left radius and bottom-right radius to 8px, and top-right radius and bottom-left radius to 10px.

border-radius:8px 10px;

# Step 42

The border-radius property accepts up to four values to round the top-left, top-right, bottom-right, and bottom-left corners.

Round the top-left corner of .three by 30 pixels, the top-right by 25 pixels, the bottom-right by 60 pixels, and bottom-left by 12 pixels.

border-radius:30px 25px 60px 12px;

# Step 43

Rotate each rectangle to give them more of an imperfect, hand-painted look.

Use the transform property on the .one selector to rotate it counter clockwise by 0.6 degrees.

transform:rotate(-0.6deg);

# Step 44

Rotate the .two element clockwise by 0.4 degrees.

transform:rotate(0.4deg);

# Step 45

Rotate .three counter clockwise by 0.2 degrees.

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

transform:rotate(-0.2deg);