Read the chapter on [Shapes of the book of shaders  (Links to an external site.)](http://patriciogonzalezvivo.com/2015/thebookofshaders/07/" \o "" \t "_blank)and complete all the proposed exercises: Rectangle exercises:

* + Experiment with the same code but using smoothstep() instead of step(). Note that by changing values, you can go from blurred edges to elegant smooth borders.
  + Change the size and proportions of the rectangle.
  + Do another implementation that uses floor().
  + Choose the implementation you like the most and make a function of it that you can reuse in the future. Make your function flexible and efficient.
  + Make another function that just draws the outline of a rectangle.
  + How do you think you can move and place different rectangles in the same billboard? If you figure out how, show off your skills by making a composition of rectangles and colors that resembles a Piet Mondrian painting.

Circles exercises:

* + Modify the above code in order to contain the entire circular gradient inside the canvas.

Distance field exercises:

* + Use step() to turn everything above 0.5 to white and everything below to 0.0.
  + Inverse the colors of the background and foreground.
  + Using smoothstep(), experiment with different values to get nice smooth borders on your circle.
  + Once you are happy with an implementation, make a function of it that you can reuse in the future.
  + Add color to the circle.
  + Can you animate your circle to grow and shrink, simulating a beating heart? (You can get some inspiration from the animation in the previous chapter.)
  + What about moving this circle? Can you move it and place different circles in a single billboard?
  + What happens if you combine distances fields together using different functions and operations?

```  
pct = distance(st,vec2(0.4)) + distance(st,vec2(0.6));  
pct = distance(st,vec2(0.4)) \* distance(st,vec2(0.6));  
pct = min(distance(st,vec2(0.4)),distance(st,vec2(0.6)));  
pct = max(distance(st,vec2(0.4)),distance(st,vec2(0.6)));  
pct = pow(distance(st,vec2(0.4)),distance(st,vec2(0.6)));  
```

* + Make three compositions using this technique. If they are animated, even better!

Polar shapes exercises:

* + Animate these shapes.
  + Combine different shaping functions to cut holes in the shape to make flowers, snowflakes and gears.
  + Use the plot() function we were using in the Shaping Functions Chapter to draw just the contour.

Combining powers exercises:

* + Using this example, make a function that inputs the position and number of corners of a desired shape and returns a distance field value.
  + Mix distance fields together using min() and max().
  + Choose a geometric logo to replicate using distance fields.