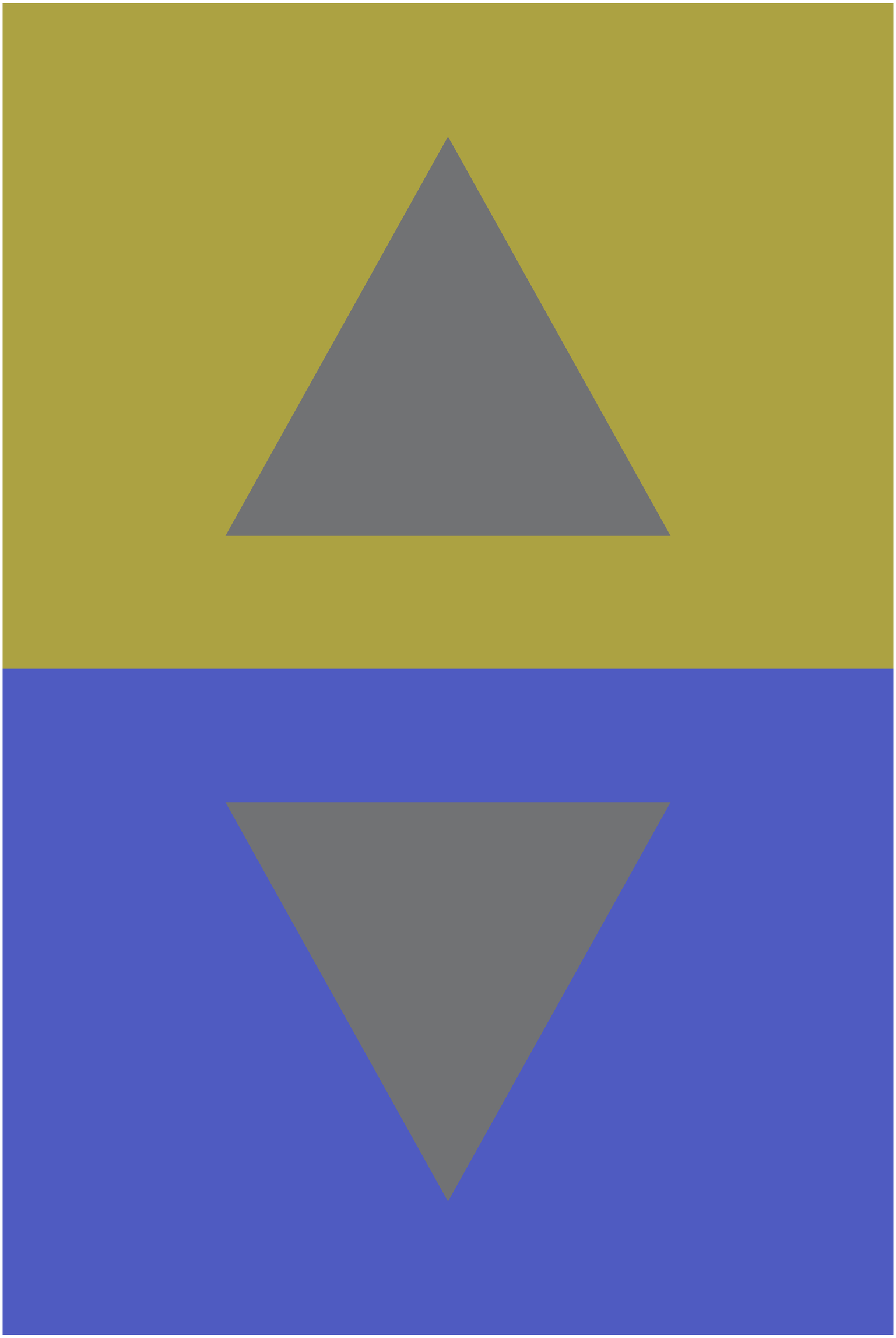
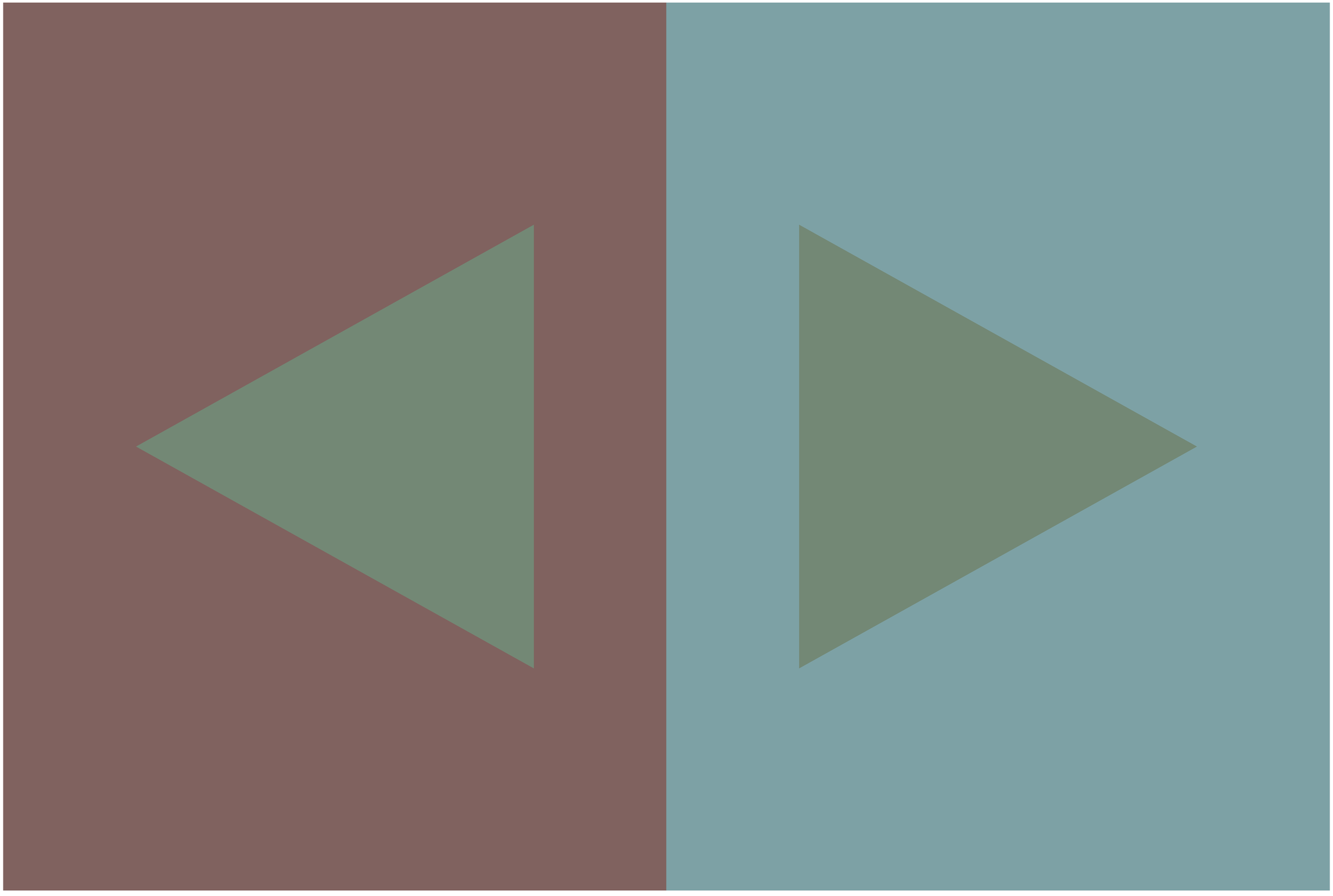
THE COLORIST COOKBOOK

"COOKING IS LIKE PAINTING OR WRITING A SONG. JUST AS THERE ARE ONLY SO MANY NOTES OR COLORS, THERE ARE ONLY SO MANY FLAVORS - IT'S HOW YOU COMBINE THEM THAT SETS YOU APART."

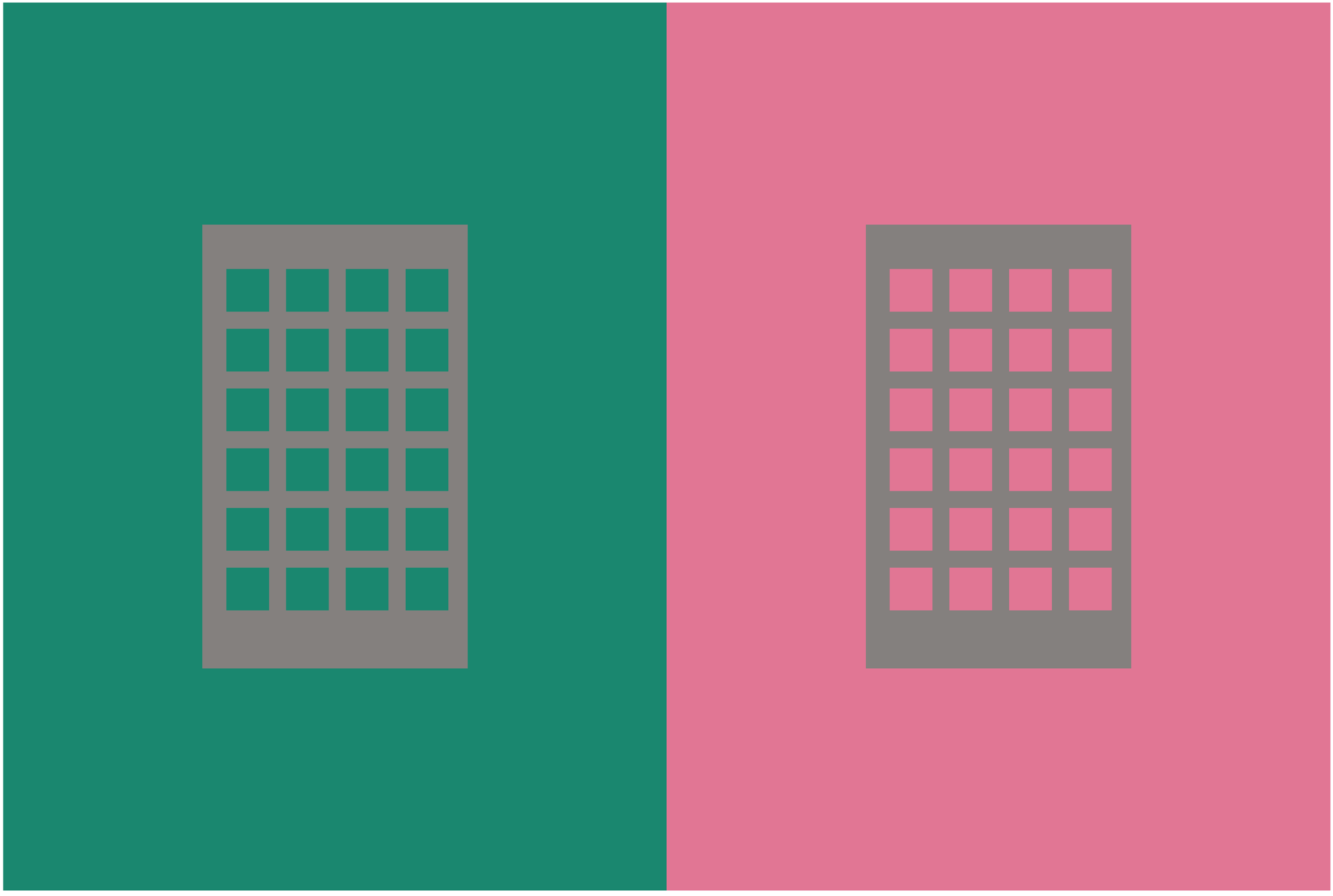


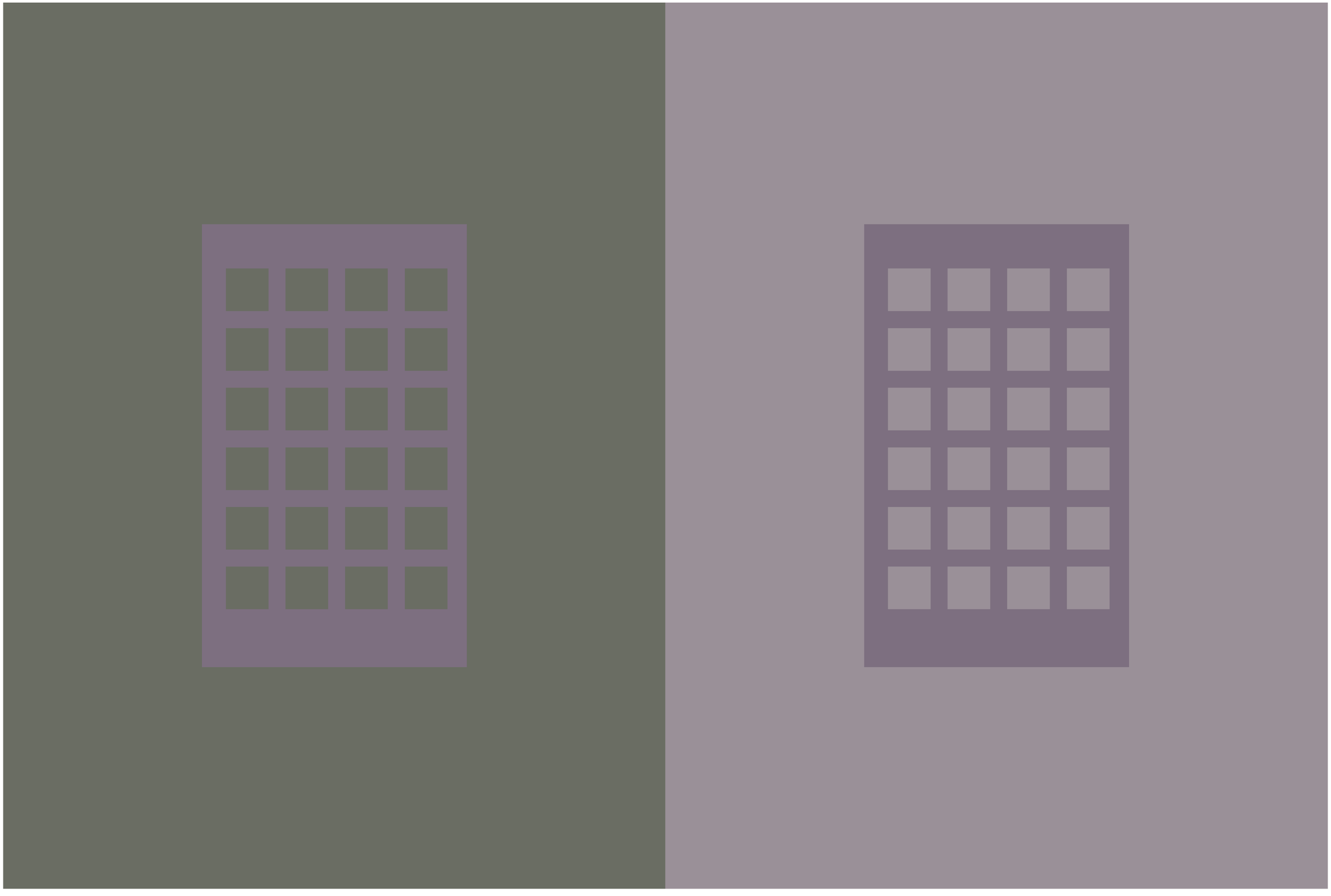
h

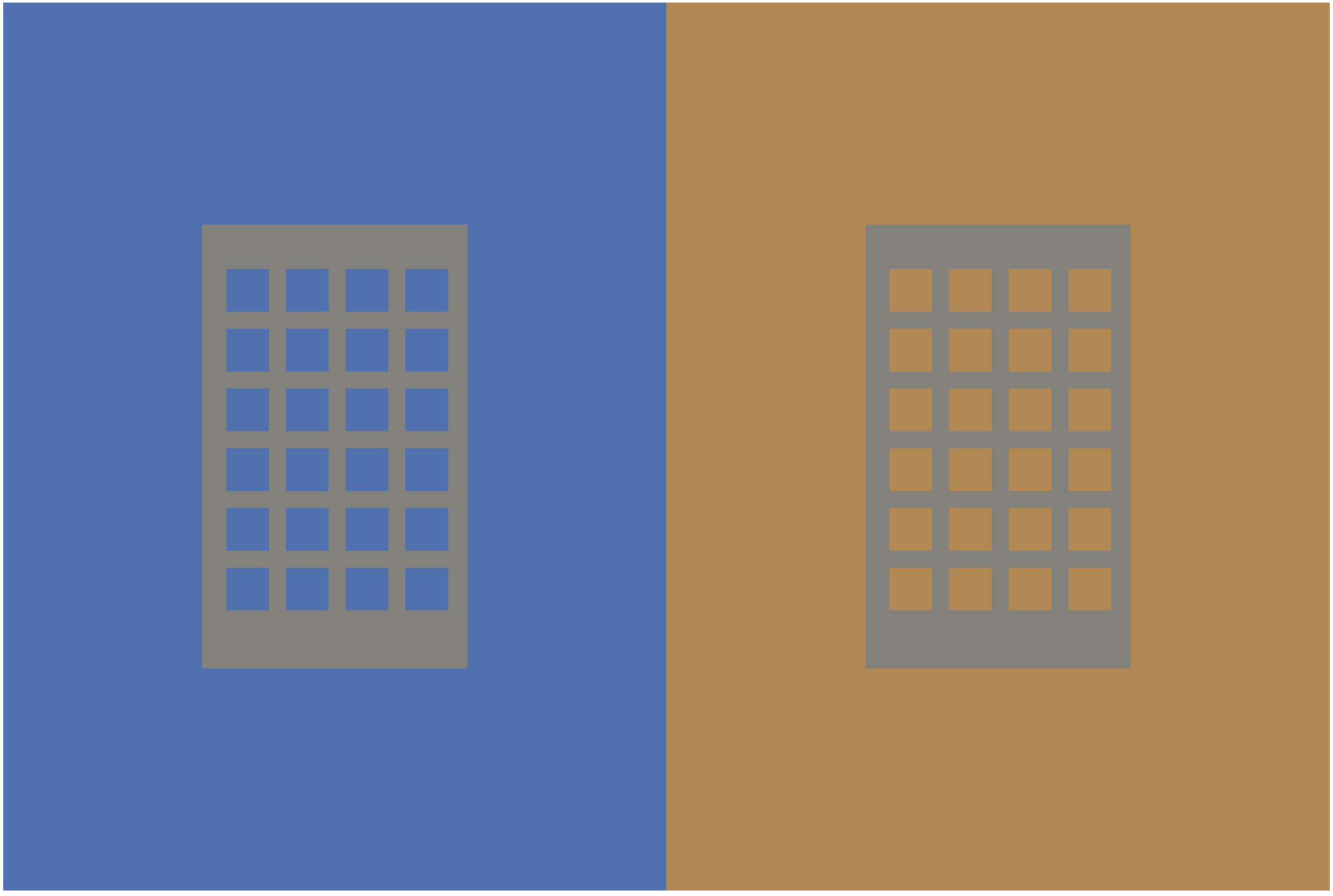


```
// recipe for simple simultaneous contrast

  // prepare the first color
  float r_col1 = random(2,84) + random(2,84) + random(2,84);
  float g_col1 = random(2,84) + random(2,84) + random(2,84);
  float b_col1 = random(2,84) + random(2,84) + random(2,84);

  color col1 = color(r_col1, g_col1, b_col1);

  // prepare the 'opposite' color to the first color
  // season with a touch of randomness
  color col2 = color(255 - r_col1 + random(-7,7),
                     255 - g_col1 + random(-7,7),
                     255 - b_col1 + random(-7,7));

// evenly mix the first two colors to create
// the 'middle' color
// (recommended) season with randomness
color mid = color((r_col1+red(col2))/2f + random(-15,15),
                  (g_col1+green(col2))/2f + random(-15,15),
                  (b_col1+blue(col2))/2f + random(-15,15)) ;

// pre-translate the transformation matrix
// to the size of your margins
pushMatrix();
translate(margin, margin);

rectMode(CORNER);
```

```
fill(col1);
stroke(col1);
rect(0,0,pgwidth/2f,pgheight);
fill(col2);
stroke(col2);
rect(pgwidth/2f,0,pgwidth/2f,pgheight);

// top with the middle color
fill(mid);
noStroke();

triangle(pgwidth*0.1,pgheight/2f,pgwidth*0.4,pgheight/4f, pgwidth*0.4,pgheight*0.75);
triangle(pgwidth*0.9,pgheight/2f,pgwidth*0.6,pgheight/4f, pgwidth*0.6,pgheight*0.75);

popMatrix();
```



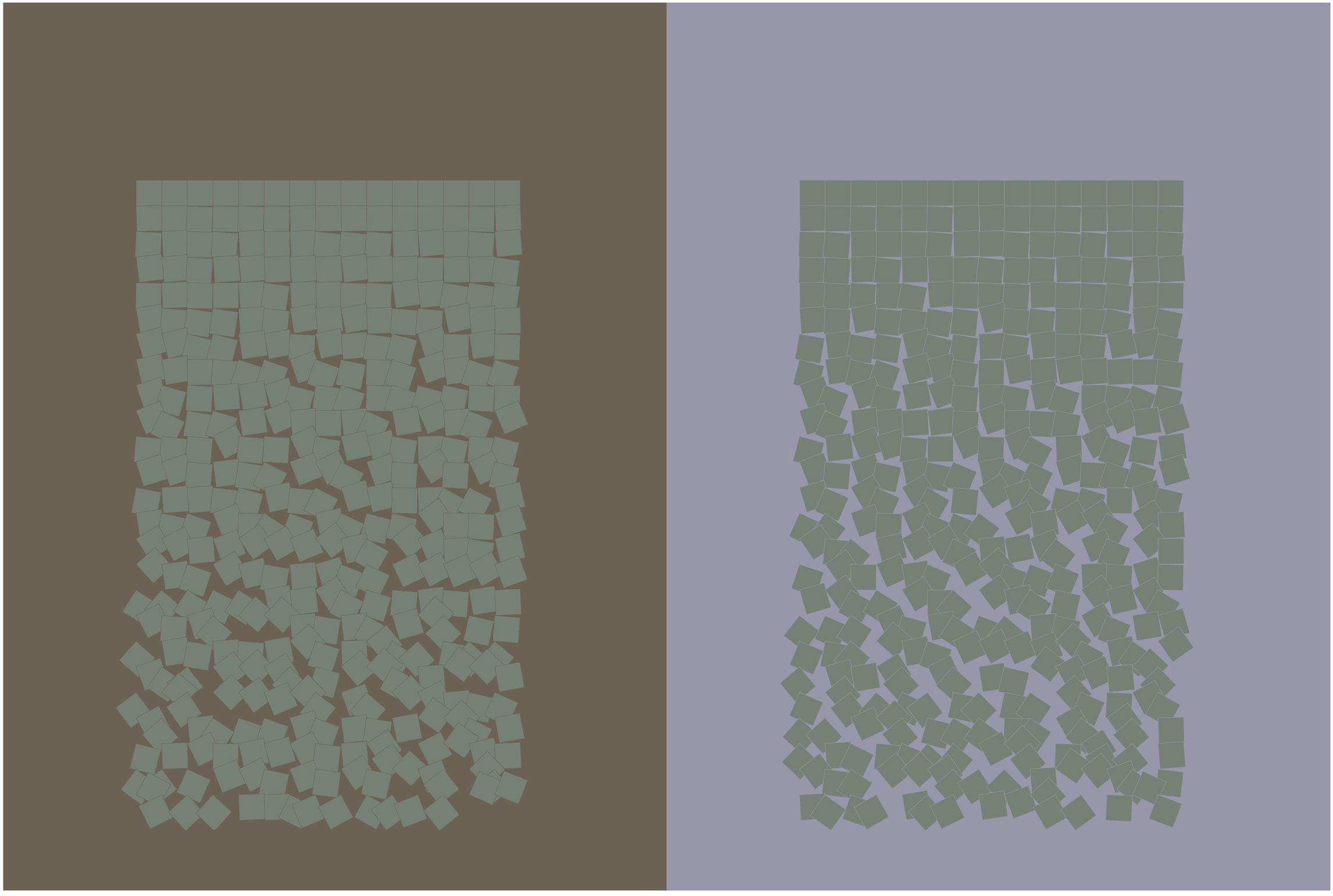




```
// recipe for holed simultaneous contrast

  // prepare the first color
  float r_col1 = random(2,84) + random(2,84) + random(2,84);
  float g_col1 = random(2,84) + random(2,84) + random(2,84);
  float b_col1 = random(2,84) + random(2,84) + random(2,84);

  color col1 = color(r_col1, g_col1, b_col1);

  // prepare the 'opposite' color to the first color
  // season with a touch of randomness
  color col2 = color(255 - r_col1 + random(-7,7),
                     255 - g_col1 + random(-7,7),
                     255 - b_col1 + random(-7,7));

  // evenly mix the first two colors to create
  // the 'middle' color
  // (recommended) season with randomness
  color mid = color((r_col1+red(col2))/2f + random(-15,15),
                    (g_col1+green(col2))/2f + random(-15,15),
                    (b_col1+blue(col2))/2f + random(-15,15)) ;

  // pre-translate the transformation matrix
  // to the size of your margins
  pushMatrix();
  translate(margin, margin);

  rectMode(CORNER);
```

```
fill(col1);
stroke(col1);
rect(0,0,pgwidth/2f,pgheight);
fill(col2);
stroke(col2);
rect(pgwidth/2f,0,pgwidth/2f,pgheight);

// top with the middle color
rectMode(CENTER);
fill(mid);
noStroke();
rect((pgwidth)/4f,pgheight/2f,pgwidth/5f,pgheight/2f);
rect((pgwidth*3f)/4f,pgheight/2f,pgwidth/5f,pgheight/2f);

// slice holes in the middle for a more dramatic effect
rectMode(CORNER);
fill(col1);
for(int rows = 0; rows < 6; rows++) {
  for(int cols = 0; cols < 4; cols++) {
    rect(pgwidth * 0.168 + 140*cols, pgheight * 0.3 + 140*rows,100,100);
  }
}

fill(col2);
float rand3 = random(-350,350);
float rand4 = random(-250,250);
pushMatrix();
//translate(rand3, rand4);
```

```
    rect(pgwidth * 0.668 + 140*cols, pgheight * 0.3 + 140*rows,100,100);
  }
}
popMatrix();

popMatrix();
```



```
// recipe for schotter simultaneous contrast

  // prepare the first color
  float r_col1 = random(2,84) + random(2,84) + random(2,84);
  float g_col1 = random(2,84) + random(2,84) + random(2,84);
  float b_col1 = random(2,84) + random(2,84) + random(2,84);

  color col1 = color(r_col1, g_col1, b_col1);

  // prepare the 'opposite' color to the first color
  // season with a touch of randomness
  color col2 = color(255 - r_col1 + random(-7,7),
                     255 - g_col1 + random(-7,7),
                     255 - b_col1 + random(-7,7));

  // evenly mix the first two colors to create
  // the 'middle' color
  // (recommended) season with randomness
  color mid = color((r_col1+red(col2))/2f + random(-20,20),
                    (g_col1+green(col2))/2f + random(-20,20),
                    (b_col1+blue(col2))/2f + random(-15,15)) ;

  // pre-translate the transformation matrix
  // to the size of your margins
  pushMatrix();
  translate(margin, margin);

  rectMode(CORNER);
```

```
fill(col1);
stroke(col1);
rect(0,0,pgwidth/2f,pgheight);
fill(col2);
stroke(col2);
rect(pgwidth/2f,0,pgwidth/2f,pgheight);

rectMode(CORNER);
fill(mid);
stroke(col1);
for(int rows = 0; rows < 25; rows++) {
  for(int cols = 0; cols < 15; cols++) {
    pushMatrix();
    translate(pgwidth * 0.1 + 60*cols, pgheight * 0.2 + 60*rows);
    float rotamnt = random(-rows*0.05,rows*0.05);
    rotate(rotamnt);
    rect(0,0,60,60);
    popMatrix();

  }
}
rectMode(CORNER);
fill(mid);
stroke(col2);
for(int rows = 0; rows < 25; rows++) {
  for(int cols = 0; cols < 15; cols++) {
    pushMatrix();
    translate(pgwidth * 0.6 + 60*cols, pgheight * 0.2 + 60*rows);
```

```
    rect(0,0,60,60);
    popMatrix();

  }
}


popMatrix();
```