Ruby Programming and Shoes

1. Draw Shapes

Draw a yellow rectangle with a blue outline:

```
Shoes.app do
  fill yellow
  stroke blue
  rect :left => 10, :top => 10, :width => 40
end
```

Try other shapes, positions and colors:

```
Shoes.app do
  fill yellow
  stroke blue
  rect :left => 10, :top => 10,
      :width => 40, :height => 20
  oval :left => 60, :top => 10,
      :width => 50, height => 200

  fill green
  arrow :left => 40, :top => 100, :width => 60

  fill purple
  stroke red
  star :left => 200, :top => 200, :points => 5
end
```

You can find a complete list of Shoes built in colors on your computer in My Documents/shoes

2. Background

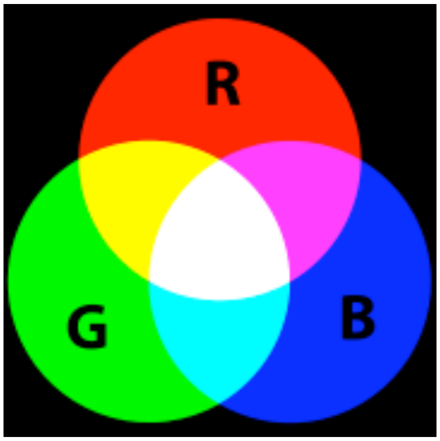
Create a solid black background:

```
Shoes.app do
   background black
 end
```

Create a gradient pattern:

```
Shoes.app do
   background yellow .. red
 end
```

3. Make Your Own Colors



On the computer screen, colors are made of light. Each pixel is created from a red, green and blue light. Light mixes very differently from paint. When paint is mixed, we call that subtractive. When we mix light, we call it additive.

For most computers these days, each red, green, and blue light in a pixel can have a value of 0 to 255, where 0 is black and 255 is the brightest color.

In Shoes, the code looks like this:

```
rgb(red_number, green_number, blue_number)
```

Colors and Code	
red	rgb(255, 0, 0)
green	rgb(0, 255, 0)
blue	rgb(0, 0, 255)
red+green = yellow	rgb(255, 255, 0)
green + blue = cyan (aqua)	rgb(0, 255, 255)
blue + red = magenta	rgb(255, 0, 255)

So, another way to tell Shoes to draw a black background:

```
Shoes.app do
   background rgb(0,0,0)
end
```

When all colors are at their brightest (255), then the pixel turns white. To draw a white background:

```
Shoes.app do
   background rgb(255,255,255)
end
```

Most of the time, you can just use the color names, but sometimes it is handy to use numbers instead.

4. Computer Generated Art

Let's make the computer draw. We can tell it to make up the number for where to put a shape, using its built-in random number generator. To tell it to pick a number between 1 and 100, we write it like this:

```
(0..100).rand
```

We can also tell it to do something over and over again. We call that a loop.

```
Shoes.app(:width => 400, :height => 600) do
  fill blue
  stroke white

  10.times do
    oval :left => (0..400).rand,
      :top => (0..600).rand,
      :radius => (25..50).rand
  end
end
```

To make it so that we can see through the overlapping shapes, we add a number to the end of 'rgb' to make the color transparent.

```
Shoes.app do
  fill rgb(0, 200, 200, 0.1)
  100.times do
     oval :left => (-10..self.width).rand,
      :top => (--10..self.height).rand,
      :radius => (25..50).rand
  end
end
```

Can you think of how to tell Shoes to make up a random color?