

(I) Warm-up Exercises

Q1. [Review] Practice with colors

Download WS02_gettingReadyEx01.html from the course web and open it in Atom. (Figure 1)

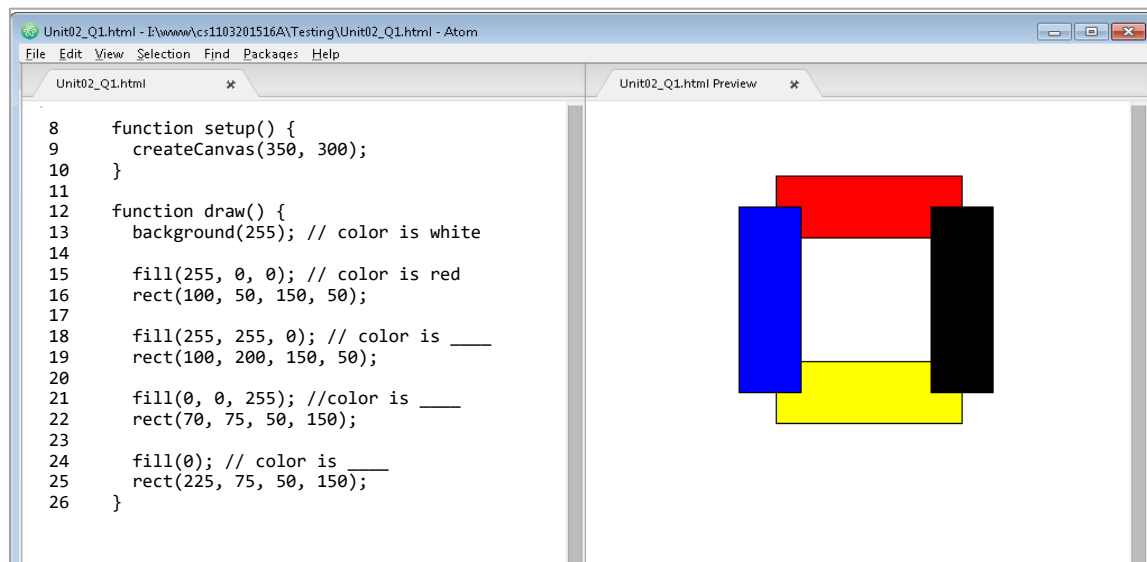


Figure 1 Four color bars

- (i) Study the code. Clarify about the work of each line. Discussions and questions are welcome!

Below is a note to help you:

Comments: Comments, like line 13 `// color is white`, are ignored and not executed by the computer. We often add comments to remind ourselves or inform others about what the code does.
[Hot key: Command - /]

Your task: Add comments to line 18, 21, 24 in Figure 1 to tell what the colors are set by the statements.

- (ii) The commands like `background()`, `fill()` etc. require color values, that can be in gray or red-green-blue.

Your task: Complete the blanks below with matching colors and values.

Note, you may modify the program in Atom to test your answers.

<code>fill(255):</code> <u>white</u>	<code>fill(255,255,255):</code> _____	<code>fill(255,0,0):</code> _____
<code>fill(200):</code> <u>light gray</u>	<code>fill(200,200,200):</code> _____	<code>fill(0,255,0):</code> _____
<code>fill(0) :</code> _____	<code>fill(0,0,0) :</code> _____	<code>fill(0,0,255):</code> _____
<code>fill(_____):</code> <u>yellow</u>		
<code>fill(_____):</code> <u>pink</u>		
<code>fill(_____):</code> <u>purple</u>		

- (iii) We can create a **transparency effect** ("see-through") by giving an *alpha value*.
0 means transparent, 255 means totally opaque

Your task: Apply the following *alpha values* as the additional parameters:

line 21 `fill(0, 0, 255, 150);`
line 24 `fill(0, 50);`

You should see the transparency effects as in Figure 2.

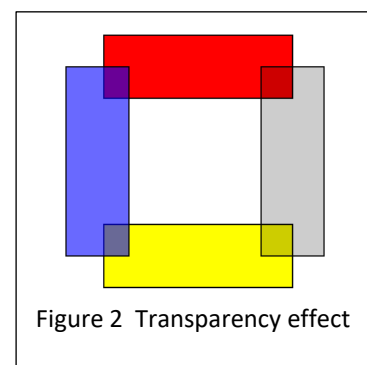


Figure 2 Transparency effect

Q2. [Review - random() and frames]

The random() command picks a random number bounded by (a) a maximum number or (b) a range.

Your task 1: Complete the blanks below:

(a) `random(255)` means to generate a random number within the range: 0 to _____

(b) `random(-300,300)` means to generate a random number within the range: _____ to _____

Your task 2: Draw green ellipses - Add the three lines of shaded code below into your Q1 program.

```
function setup() {
  createCanvas(300, 300);
  frameRate(10);
}

function draw() {
  background(255); // color is white

  fill(0,255,0); // color is green
  ellipse(random(0,300),random(0,300),random(50),random(50));

  ...
}
```

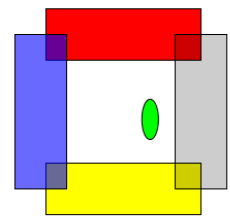


Figure 3 A random Green ellipse in each frame (random location, random size)

Study the code and learn from it. Discussions and questions are welcome!

Below is a note to help you:

frameRate: Set the frequency of running the draw function.
`frameRate(10);` means to draw 10 times per second.
 The default frame rate is 60 times per second. By changing it to 10 times per second, the animation slows down.

Q3. [Review - background() and more]

(i) The background() command within draw() clears the entire background when each frame starts.

Your task: Examine the effect of **not calling** `background()` in draw()
 -- Please simply comment it out like this:
`// background(255);`

[Hot key to comment-out a line: `Command - /`]

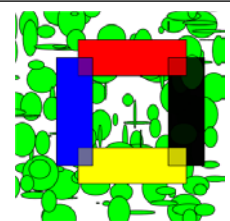


Figure 4 The green ellipse drawn in each frame retains

(ii) We can create a **fading-out** effect by giving an *alpha value* to background()

Your task: Apply the alpha value as `background(255, 50);`
 The result should look like Figure 5. Any problem? Ask for help!

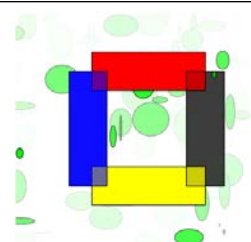


Figure 5 Each green ellipse drawn in a frame retains a while and fades out completely

(II) Working out Class Exercises**[Job – 1]** Move the ball inside the canvas

(a) horizontally from left to right

```

var x =      ;

function setup() {
  createCanvas(300, 300);
}

function draw() {
  background(100);
  fill(250, 255, 0);

  ellipse(____,____,50,50);

}

```

(b) vertically from bottom to top

```

function setup() {
  createCanvas(300, 300);
}

function draw() {
  background(100);
  fill(250, 255, 0);

  ellipse(____,____,50,50);

}

```

(c) diagonally to the bottom right

```

function setup() {
  createCanvas(300, 300);
}

function draw() {
  background(100);
  fill(250, 255, 0);

  ellipse(____,____,50,50);

}

```

[Hot key] To tidy up the **format of program code**: Ctrl-Alt-b (The Atom Beautify Package)

To open or close preview: Ctrl-Shift-H (The Atom HTML Preview Package).

Warning: may easily crash when you show preview while editing the code.

[Job – 2] Trace the changes of variable values in the following program? What will be the final output?

Values of variables:

```

1  function setup() {
2    createCanvas(200, 50);
3    background(230);
4
5    var a, b, c, d;
6    a = -10;
7    a = a + 1;
8    b = 30;
9    b = a;
10   b++;
11   a -= 10;
12   c = -b;
13   d = c % 3;
14   d + 1;
15
16   text(a, 0, 20);
17   text(b, 50, 20);
18   text(c, 100, 20);
19   text(d, 150, 20);
20 }

```

	<u>a</u>	<u>b</u>	<u>c</u>	<u>d</u> i.e., Undefined
after line 6. <u>a = -10;</u>	-10	--	--	-- ↙
after line 7. <u>a = a + 1;</u>				
after line 8. <u>b = 30;</u>				
after line 9. <u>b = a;</u>				
after line 10. <u>b++;</u>				
after line 11. <u>a -= 10;</u>				
after line 12. <u>c = -b;</u>				
after line 13. <u>d = c % 3;</u>				
after line 14. <u>d + 1;</u>				

[Final output of the program]
The contents painted by line 16-19 are:



Download WK02_Job02.html and check!

[Job –3] Displays a **vertical/Horizontal** line following cursor movement

(a) horizontal line following the mouse

```
function setup() {
  createCanvas(300, 300);
}

function draw() {
  background(200);
  stroke(255, 0, 0);

}
```

(b) vertical line following the mouse

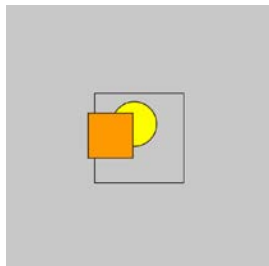
```
function setup() {
  createCanvas(300, 300);
}

function draw() {
  background(200);

}
```

[Discussion – Slide 20 Global & Local Variables]

Why the yellow ball can never move out of the small window?



```
var a = 150;
var b = 150;

function setup() {
  createCanvas(300, 300);
  frameRate(10);
}

function draw() {
  var x = 150;
  var y = 150;

  background(200);
  noFill();
  rect(100, 100, 100, 100);

  //flying ball
  fill(255, 255, 0);
  x += random(-20, 20);
  y += random(-20, 20);
  ellipse(x, y, 50, 50);

  //flying box
  fill(255, 153, 0);
  a += random(-20, 20);
  b += random(-20, 20);
  rect(a-25, b-25, 50, 50);
}
```

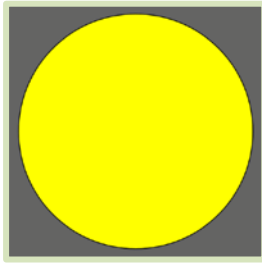
[Job –4] Move the ball inside the canvas, **vertically from bottom to top** again and again.

```
function setup() {
  createCanvas(300, 300);
}

function draw() {
  background(100);
  fill(250, 255, 0);

  ellipse(____,____,50,50);

}
```

[Job –5] Ball grows and shrinks

```
var radius = 25;

function setup() {
  createCanvas(300, 300);
}

function draw() {
  background(100);
  stroke(0);
  fill(255,255,0);
  ellipse(width/2, height/2, radius*2, radius*2);
}

}
```

[Job – 6]

Complete the code on the right:

Objectives:

The canvas gradually turns red ($r = r + 1$) when mouse cursor is on the left side of canvas. Otherwise, gradually turns dark ($r = r - 1$).

If the value of r is greater than 255, set the value to 255.

If the value of r is smaller than 0, set the value of r to 0.



```
var r = 150;    // red

function setup() {
  createCanvas(400,200);
}

function draw() {
  background(r,0,0);
  stroke(255);
  line(width/2,0,width/2,height);

  //Show mouse coordinates
  //Learn:  (1) Use double quotes for text literal: "...",
  //         (2) use + to put together the output contents
  fill(255); //choose text color
  text("mouse: " + mouseX + ", " + mouseY, 5, height-5);
}
```

[Job -7]

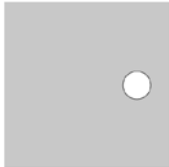
Complete the code on the right:

Objectives:

Split the canvas equally into four zones.

A bouncing ball moves horizontally across the canvas.

The ball shows different colors when moving across different zones.



```
function setup() {
  createCanvas(300, 300);
}
```

```
function draw() {
  background(100);
```

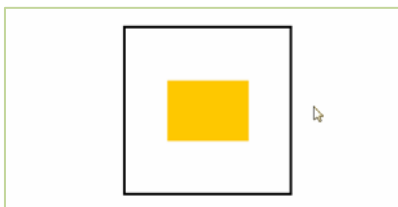
```
  ellipse(x, 150, 50, 50);
}
```

[Job – 8]

Complete the code on the right:

Objectives:

The background of canvas turns dark when the mouse cursor is inside the yellow rectangle.



```
function setup() {
  createCanvas(200, 200);
}
```

```
function draw() {
  //Fill background (Your task: rewrite this part)
  //background(200);
```

```
  stroke(255);
  fill(255, 200, 0);
  rect(50, 60, 100, 80);
```

```
  //Show mouse location
  fill(255,0,0); //choose text fill color
  stroke(0,0); // choose text outline color = transparent
  text("mouse: " + mouseX + ", " + mouseY, 5, height-5);
}
```