CS1103 Workshop 9

MATH And SOUND

Probability, map, sin, cos, SOUND

PROBABILITY

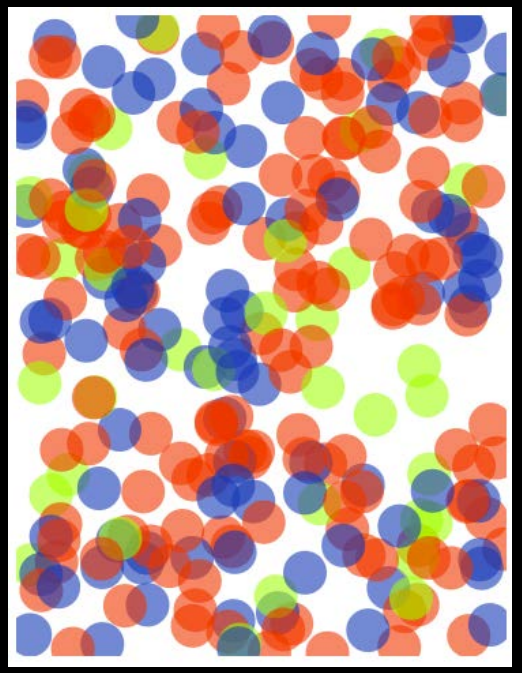
Probability is the measure of the likeliness that an event will occur. Probability is quantified as a number between 0 and 1. The higher the probability of an event, the more certain we are that the event will occur.

A simple example is the toss of a (unbiased) coin. Since the two outcomes are equally probable, the probability of "heads" equals the probability of "tails", so the probability is 1/2 (or 50%) chance of either "heads" or "tails".

Sample Program 1:

Draw a ball of either red, green or blue, each with a probability.

```
function setup() {
  createCanvas(360, 470);
  background(255);
  noStroke();
}

function draw() {

  // Probabilities for 3 different cases
  // These need to add up to 100%!
  var red_prob = 0.60;     // 60% chance of red color
  var green_prob = 0.10;   // 10% chance of green color
  var blue_prob = 0.30;    // 30% chance of blue color

  // Pick a random number between 0 and 1
  var num = random(1);

  if (num < red_prob) {
    fill(255, 53, 2, 150);
  } else if (num < green_prob + red_prob) {
    fill(156, 255, 28, 150);
  } else {
    fill(10, 52, 178, 150);
  }

  ellipse(random(width), random(height), 32, 32);
}
```



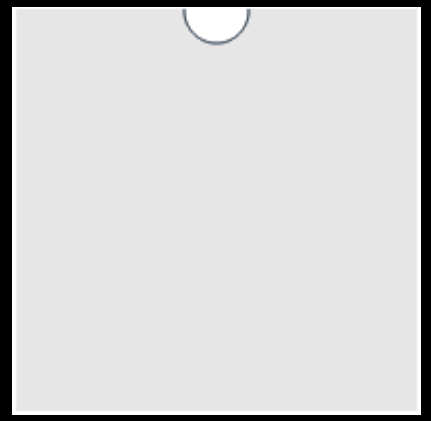
Three cases:

- num is less than .6
- num is between .6 and .7
- All other cases (between .7 and 1.0)

Job 1:

Write a program so that a ball has 10% chance of moving up, a 20% of chance moving down, and a 70% chance of doing nothing.

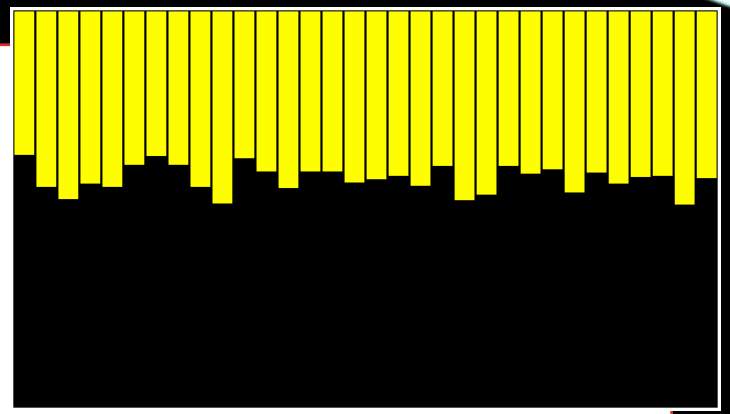
By common sense, guess what will happen to the ball eventually?



Sample Program 2:

Suppose we repeatedly throw a dice for 100 times and count the occurrences of each number. Then, we can have a histogram or "distribution" showing the probability that each number will occur.

Sample program 2 shows an example of generating a histogram of 32 numbers randomly. You will notice that even though these numbers are *randomly* generated, each number has more or less equal probability of generation. In other words, the random() function is unbiased, basically.



```
var randomCounts = [];
var numPoints;

function setup() {
  createCanvas(640, 360);
  background(0);

  numPoints = width / 20;

  for (var x = 0; x < numPoints; x++) {
    randomCounts[x] = 0;
  }
}

function draw() {

  // Pick a random number and increase the count
  // floor() is applied to obtain the largest whole number not larger than the random number.
  var index = floor(random(numPoints));
  randomCounts[index]++;

  // Draw a rectangle to graph results
  stroke(0);
  fill(255, 255, 0);
  rect(index * 20, 0, 19, randomCounts[index]);
}
```

MAP

The map() function, which converts one range of values to another range, can be very useful

map(theVariable, actualMin, actualMax, targetMin, targetMax)

Parameters:

theVariable : the incoming number to be converted
actualMin : the lower bound of the incoming number's current range
actualMax : the upper bound of the incoming number's current range
targetMin : the lower bound of the target range
targetMax : the upper bound of the target range

Return value: remapped number

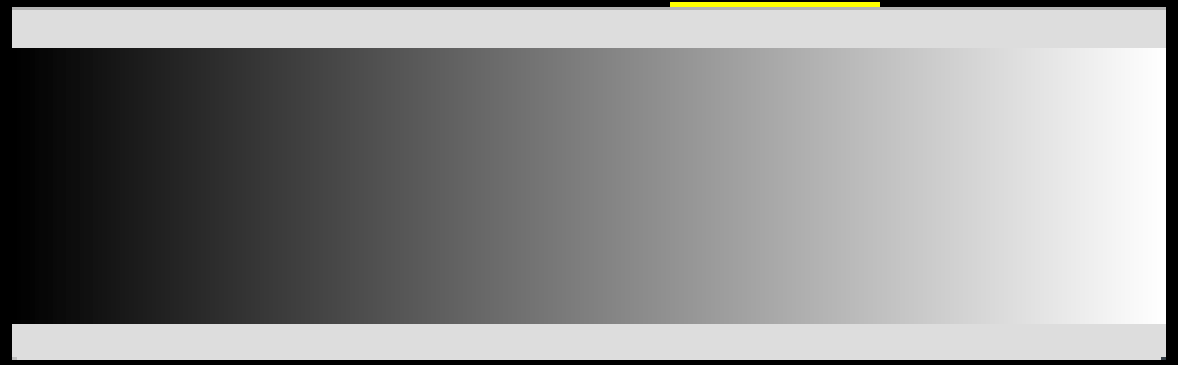
Sample Program 3:

MAP

Given:
The canvas size is 256x100
We paint gray colors in 256 levels:

```
createCanvas(256, 100);

var x = 0;

while (x < 256) {
  stroke(x);
  line(x, 0, x, height);
  x++;
}
```

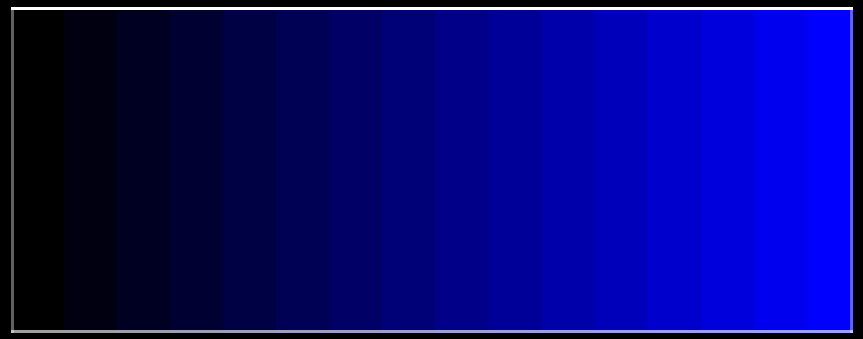
How about if canvas size is **400x100**?



```
createCanvas(400, 100);

var x = 0;

while (x < width) {
  var c = map(x, 0, width, 0, 256);
  stroke(c);
  line(x, 0, x, height);
  x++;
}
```

MAP

Job 2: Generate 16 stripes in gradients of blue

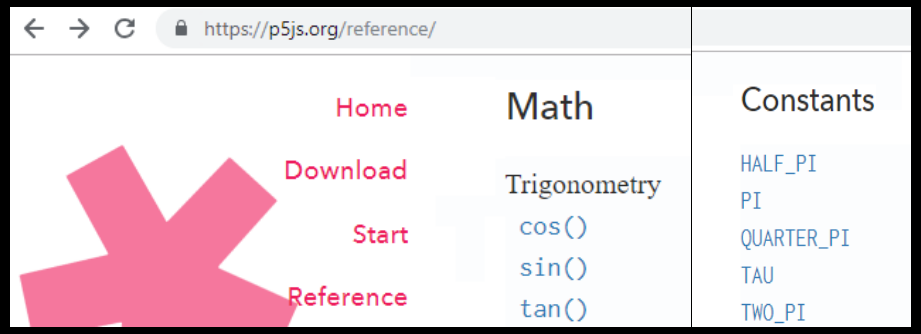


```
createCanvas(256, 100);
noStroke();

var s = 0;

while (s < 16) {
  var c = map(s, ___, ___, ___, ___);

  fill(0, 0, c);
  rect(s * 16, 0, 16, 100);

  s++;
}
```

SIN AND COSINE WAVES

Trigonometry is a branch of mathematics that studies relationships between the sides and angles of triangles.

Examples of trigonometry functions are sin, cosine, tangent.



https://p5js.org/reference/

```
text(sin(0).toFixed(4), 10, 50);
text(sin(3.14).toFixed(4), 10, 100); //sin(180 degrees)
text(sin(3.14159).toFixed(4), 10, 150);
text(sin(3.1416).toFixed(4), 10, 200);
text(sin(1.57).toFixed(4), 10, 250); //sin(90 degrees)
text(sin(0.7854).toFixed(4), 10, 300); //sin(45 degrees)

text(PI.toFixed(8), 10, 400);

text(sin(PI).toFixed(4), 10, 500);
text(sin(TWO_PI).toFixed(4), 10, 550);
text(sin(HALF_PI).toFixed(4), 10, 600); //sin(90 degrees)
text(sin(QUARTER_PI).toFixed(4), 10, 650); //sin(45 degrees)
```

WS09_Slide09_PI_sin_

```
0.0000
0.0016
0.0000
-0.0000
1.0000
0.7071

3.14159265

0.0000
-0.0000
1.0000
0.7071
```

SIN AND COSINE WAVES

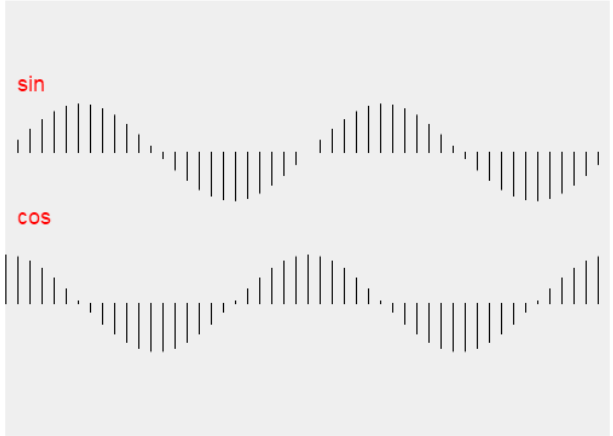
Trigonometry is a branch of mathematics that studies relationships between the sides and angles of triangles.

Examples of trigonometry functions are sin, cosine, tangent.

Your task:
Study and try *sample program* 4 to visualize the difference between sin and cosine waves.

Sample Program 4:

```
function setup() {
  createCanvas(500, 360);
  background(240);
  fill(255, 0, 0);
  textSize(18);
  text("sin", 10, 75);
  text("cos", 10, 185);

  var a = 0.0;
  var inc = TWO_PI / 25;

  for (var i = 0; i < 50; i++) {
    line(i*10, 125, i*10, 125 - sin(a)*40);
    line(i*10, 250, i*10, 250 - cos(a)*40);
    a = a + inc;
  }
}
```

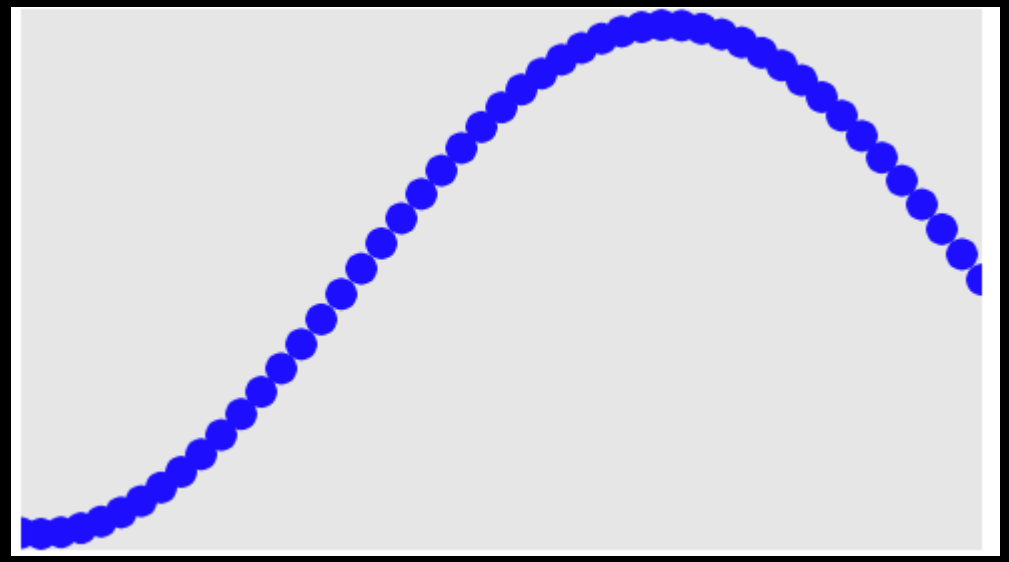


TWO_PI is a constant defined in p5.js. It is about 2 x 3.1416.

Sample Program 5:

Trigonometric function is useful in simulating *smooth* motion.

Sample program 5 uses sin to simulate the movement of snake.



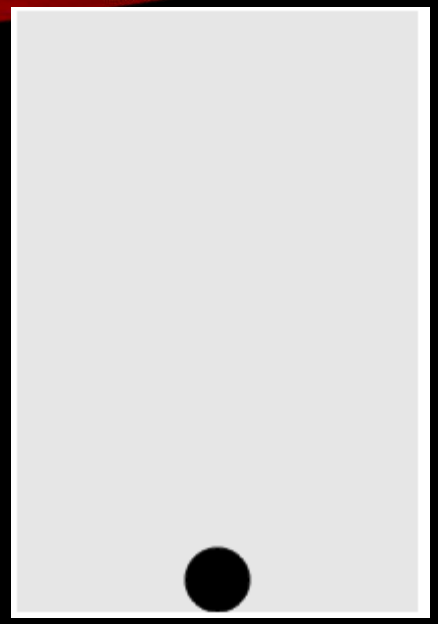
```
//The first theta whose sine value is drawn as a dot on the canvas
var firstTheta = 0.0;

function setup() {
  createCanvas(480, 270);
  noStroke();
  fill(0, 0, 255);
}

function draw() {
  background(230);

  // Increment firstTheta for every frame
  // Todo: try other values of "angular velocity"
  firstTheta += 0.02;

  var theta = firstTheta;

  // a for-loop draws all points along a sine wave (scaled to canvas size).
  for (var x = 0; x <= width; x += 10) {

    // Calculate y using the sine function
    var y = map(sin(theta), -1, 1, 8, height-8);
    ellipse(x, y, 16, 16);
    theta += 0.1;
  }
}
```

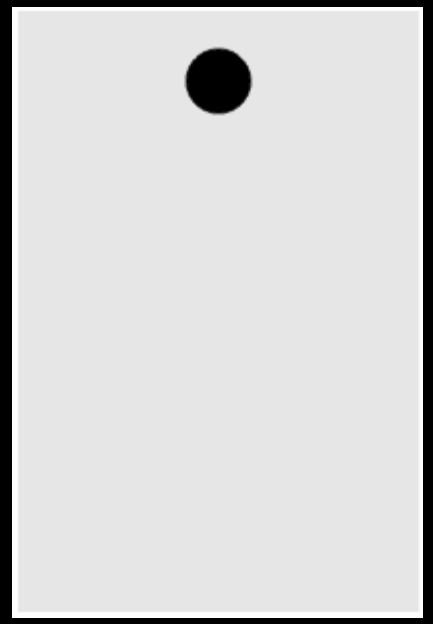
Sample Program 6:

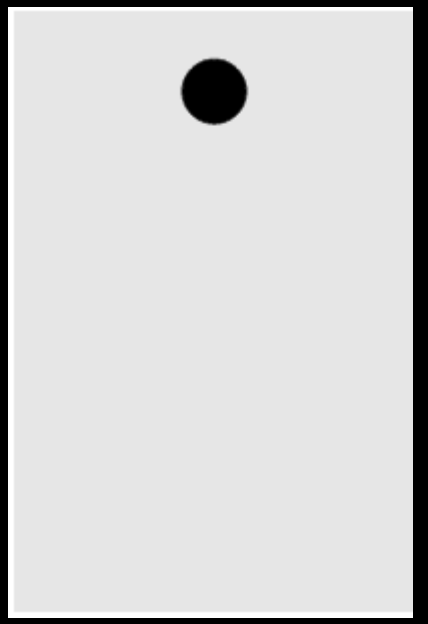
This example swings a ball using sin function. The movement is smooth and natural, basically.

```
var theta = 0.0;

function setup() {
  createCanvas(480, 270);
  fill(255, 0, 0);
  stroke(0);
}

function draw() {
  background(240);

  // The output of the sin() function oscillates
  // smoothly between -1 and 1.  We map it
  // to an x coordinate between 16 and width-16
  var x = map(sin(theta), -1, 1, 16, width - 16);

  // With each cycle, increment theta
  theta += 0.05;

  // Draw the ellipse at the value produced by sine
  line(width / 2, 0, x, height / 2);
  ellipse(x, height / 2, 32, 32);
}
```

Job 3: Modify *sample program* 6 for the animations of bouncing ping pong ball.







(a) The ping pong ball bounces back when hitting the ground, and drops when reaching certain height. Using sin wave, you should see that the ball moves more naturally.
[Approach: Map the sin values of -1 to +1 to vertical positions between top and bottom]

(b) The ping pong ball rises and drops slower at high positions, then hits and bounces the ground quickly.
[Approach: Using theta from 0 to PI only, map the sin values of 0 to +1 to vertical positions between bottom and top]

(c) The speed of the ball decays and the ball eventually stops. *Hint*: initialize a path-height and reduce it gradually.

SOUND IN P5.JS

p5.sound is an additional library in p5.js

That allows us to:

- play/loop background sounds
- trigger sounds on events
- analyze sound in real-time

etc..

https://p5js.org/reference/#/libraries/p5.sound

SOUND IN P5.JS

The sound commands:

```
var sound = loadSound(filepath);
```

- loads the sound file
- returns a sound object
- Similar to `loadImage`

  only works locally in FireFox(10.4 and above)

```
sound.play();
```

- play the sound once

```
sound.loop();
```

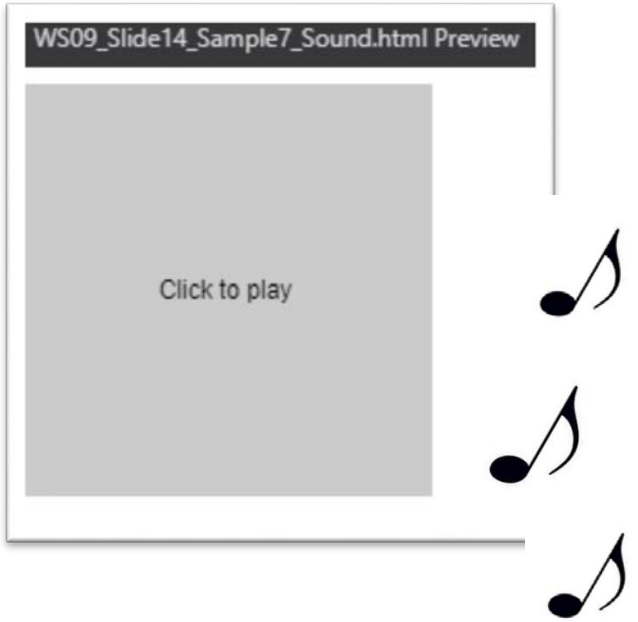
- play the sound repeatedly

Sample Program 7:

This program loads a sound track from a sound file. It plays the sound track when the mouse button is pressed.

If we change this to `sound.loop();` The sound track will repeat playing

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.5.14/p5.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.5.14/addons/p5.sound.js"></script>

<script>
  var sound;

  function preload() {
    sound = loadSound("Canon5s.mp3");
  }

  function setup() {
    createCanvas(200, 200);
    background(200);

    textAlign(CENTER,CENTER);
    text("Click to play", width/2,height/2);
  }

  function mousePressed() {
    sound.play();
  }

</script>
```



Sample Program 8:

More functions:

`isPlaying()`

- Return true if the sound is currently playing; otherwise return false.

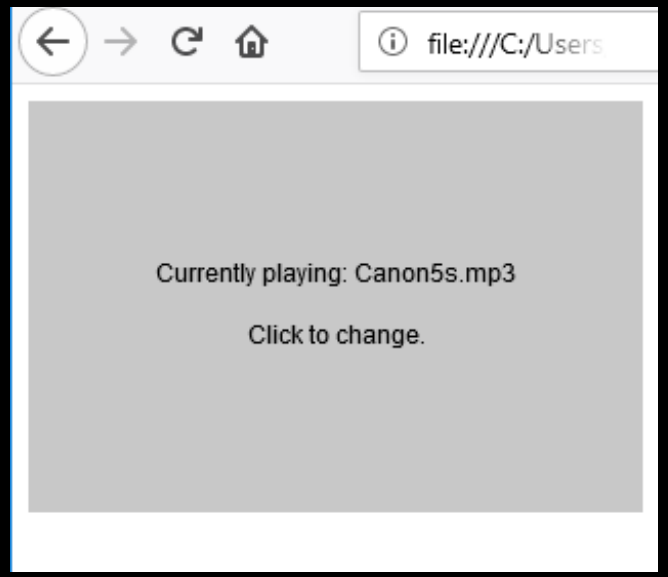
`stop()`

- Stop the playing sound.

```
var sound;

function preload() {
  sound = loadSound("Canon5s.mp3");
}

function setup() {
  createCanvas(200, 200);
  textAlign(CENTER, CENTER);
}

function draw() {
  background(200);
  if (sound.isPlaying()) {
    text("Song is playing.  Click to stop.", width / 2, height / 2);
  } else {
    text("Click to play", width / 2, height / 2);
  }
}

function mousePressed() {
  if (sound.isPlaying()) {
    sound.stop();
  } else {
    sound.play();
  }
}
```

Job 4a:

- Create an array of sound files
- Loop over each to load them
- Select a random sound to loop
- Click mouse to select a new one to loop



```
var files = ["Sheep.mp3", "Canon5s.mp3", "DeerHunter.mp3", "Popeye.mp3"];
var sound = [];
var iCurrent = 0; //The index of the sound that is playing

function preload() {
  //Load the sounds in a loop


}

function setup() {
  createCanvas(300, 200);
  textAlign(CENTER, CENTER);

  //Pick a random sound and loop it


}

function draw() {
  background(200);
  //Display the file name of the current sound
  text("Currently playing: "       , width / 2, height / 2 - 15);
  text("Click to change.", width / 2, height / 2 + 15);
}

function mousePressed() {
  //Stop the current sound


  //Pick a random sound and loop it

}
```

SOUND IN P5.JS

Analyzing sound

```
var am = new p5.Amplitude();
```

- p5.Sound allows us to analyze sound in real-time
- We create an amplitude object to get the amplitude ~volume of the sound at each moment.
- We can sync motion to the beat…

```
am.setInput(sound);
```

- link up with a sound object

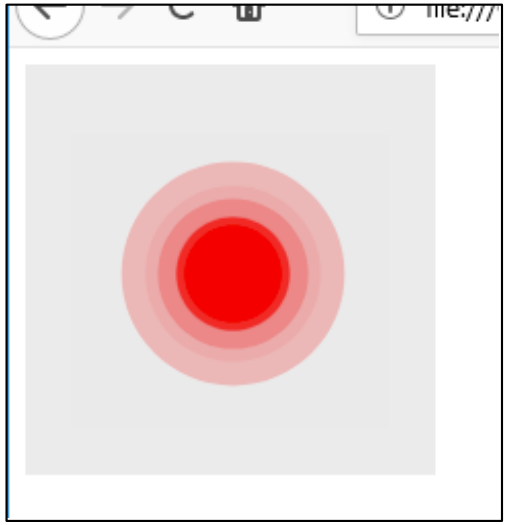
```
am.getLevel();
```

- get the amplitude (0 to 1.0)

Sample Program 9:

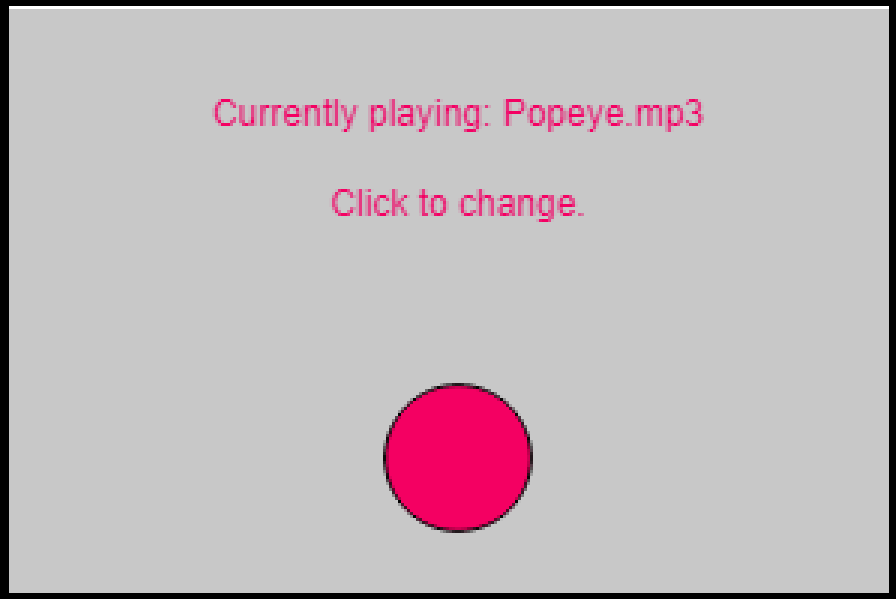
This program displays a ellipse of changing size to reflect the beat of the sound.

```
var sound;
var am;

function preload() {
  sound = loadSound("Popeye.mp3");
}

function setup() {
  createCanvas(200, 200);
  textAlign(CENTER, CENTER);
  am = new p5.Amplitude();
  am.setInput(sound);

  sound.play();
}

function draw() {
  background(235,50);
  fill(255,0,0);
  noStroke();

  var diameter = map(am.getLevel(), 0, 1, 0, width);
  ellipse(100, 100, diameter, diameter);
}
```



Job 4b:

Modify your program of Job 4a so that the beat of the playing sound is shown.



Assignment of this lesson:

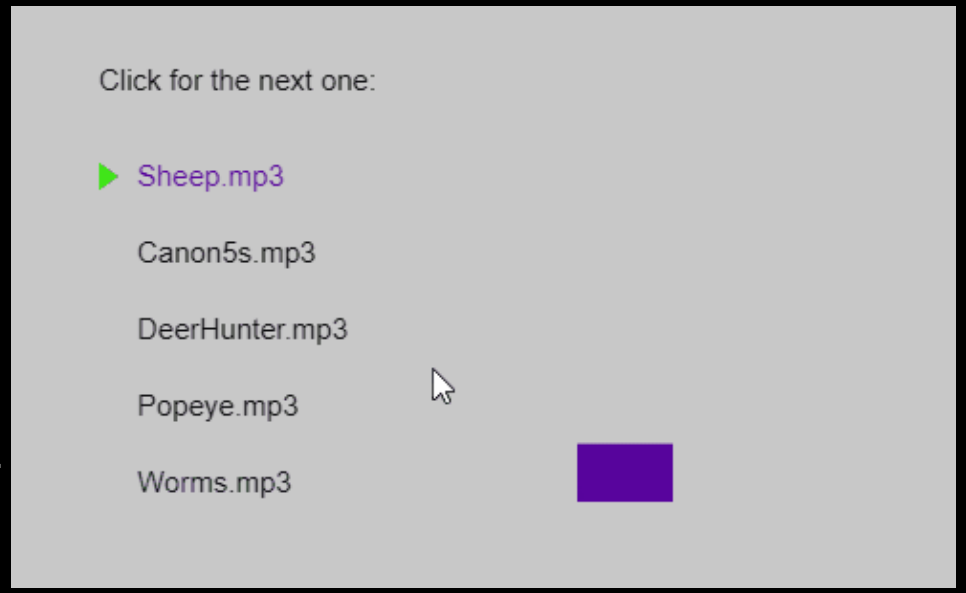
Create a program according to the requirement below.
Submit it on Canvas.
Due: Please check the deadline on Canvas.

Create the program according to the requirements below (5 marks)

Basic requirements – Please implement the program as shown in the animation [3 marks]

Details:

1. Extend your program of Job4b to list the names of the sound files.
2. Play the first sound initially.
3. Change to the next sound when the mouse button is pressed.
   In case the current one is the last sound in the list, go back to play the first sound.
4. Illustrate the sound that is currently playing
   (In the given animation, a green triangle is painted next to the sound file name.)
5. Analyze the amplitude level of the playing sound and animate according to the beat.
   (In the given animation, a vertical bar of changing size is shown.)



Creative component [2 mark]
Implement creative features for point 4 and 5 of the above.

Note: You may use your own sound files. But the total size of your submission must be ***under 10 MB****.*

Animation: https://courses.cs.cityu.edu.hk/cs1103/public/Workshop09_Animations