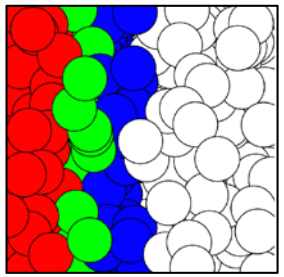
CS1103 Workshop 3

1. MODULO
2. DATA TYPES
3. LOOPS

CODING STYLE, DRAWING MODES AND TEXT CONTROL, %, NUMBER AND TEXT AND BOOLEAN VALUES, *WHILE-LOOP, FOR-LOOP*

```
function draw() {
    var x = random(300);
    var y = random(300);

    if (x < width/4) {
        fill(255, 0, 0);
    } else if (x < width/3) {
        fill(0, 255, 0);
    } else if (x < width/2) {
        fill(0, 0, 255);
    } else {
        fill(255);
    }

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



CODING STYLE

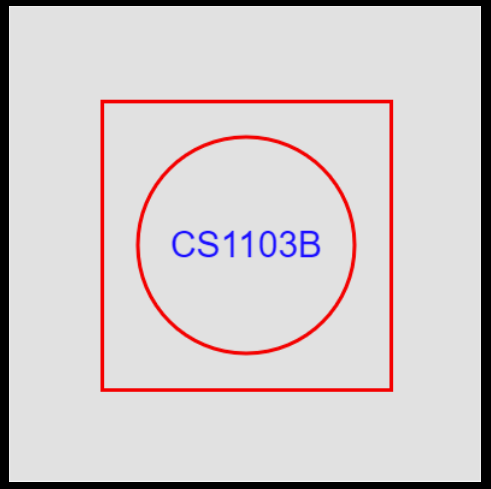
Its very *important* to use proper indentation and spacing in your code

Why?

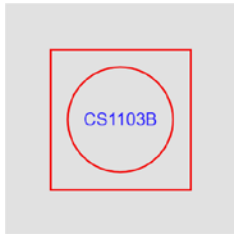
- It is easier for you to read
- It is easier for others to read
- It will help you to spot mistakes
- It shows you understand
  the craft of coding

DRAWING MODES

Draw a rectangle, an ellipse, and "CS1103B" at the center of the canvas.
We use the object center points in the drawing commands.



```
noFill();
strokeWeight(3);
stroke(255,0,0);

rectMode(CENTER);
rect(200,200,240,240);

ellipseMode(CENTER);
ellipse(200,200,180,180);

textAlign(CENTER,CENTER);
textSize(30);
noStroke();
fill(0,0,255);
text("CS1103B",200,200);
```



This is needed because the default rectangle mode is CORNER

This is optional since the default ellipse mode is already CENTER

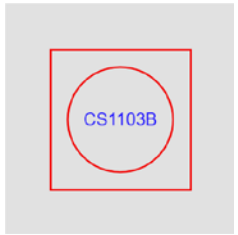
DRAWING MODES

We may alter the meaning of parameter values when drawings are made.

rectMode:
- affect rect(x, y, width, height)
- rectMode(CORNER) : (x, y) is the top-left
- rectMode(CENTER) : (x, y) is the center
- The default is *CORNER*

ellipseMode:
- affect ellipse(x, y, width, height)
- ellipseMode(CORNER) : (x, y) is the top-left
- ellipseMode(CENTER) : (x, y) is the center
- The default is *CENTER*

```
noFill();
strokeWeight(3);
stroke(255,0,0);

rectMode(CENTER);
rect(200,200,240,240);

ellipseMode(CENTER);
ellipse(200,200,180,180);

textAlign(CENTER,CENTER);
textSize(30);
noStroke();
fill(0,0,255);
text("CS1103B",200,200);
```



This is needed since the default text alignment mode is LEFT, BOTTOM

DRAWING MODES

We may alter the meaning of parameter values when drawings are made.

textAlign:

- affect text(content, x, y)
- textAlign(LEFT, BOTTOM) : (x, y) is bottom-left
- textAlign(CENTER, CENTER) : (x, y) is center
- The default is LEFT, BOTTOM

More:

textSize(*size_in_pixel*) – set the font size

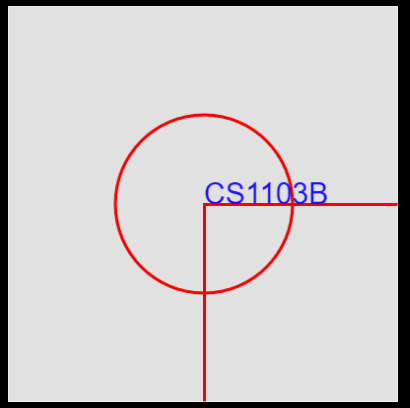
fill(*color*) – set the text color

noStroke() – *noStroke* to avoid the outline that may make the text not pretty.

QUESTION

If we do not set the drawing modes, we will get the following result. Why?

```
noFill();
strokeWeight(3);
stroke(255,0,0);

rect(200,200,240,240);

ellipse(200,200,180,180);

textSize(30);
noStroke();
fill(0,0,255);
text("CS1103B",200,200);
```



CS1103 Workshop 3

1. MODULO
2. DATA TYPES
3. LOOPS

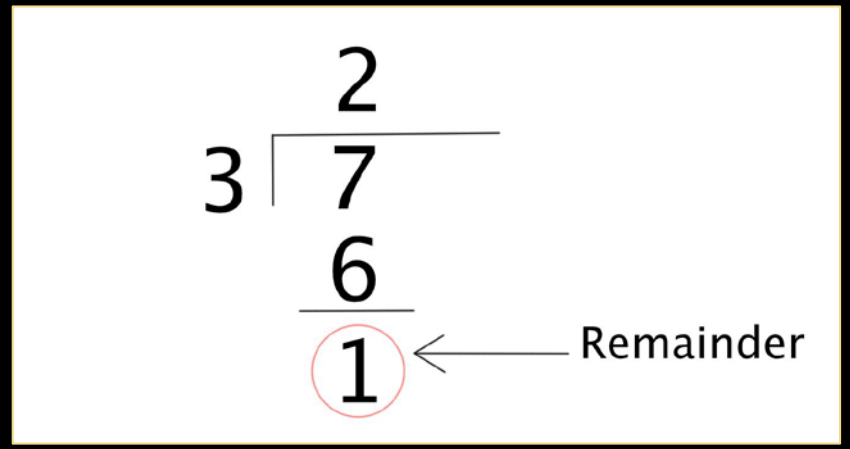
CODING STYLE, DRAWING MODES AND TEXT CONTROL,
%, NUMBER AND TEXT AND BOOLEAN VALUES,
WHILE-LOOP, FOR-LOOP

MODULO: %

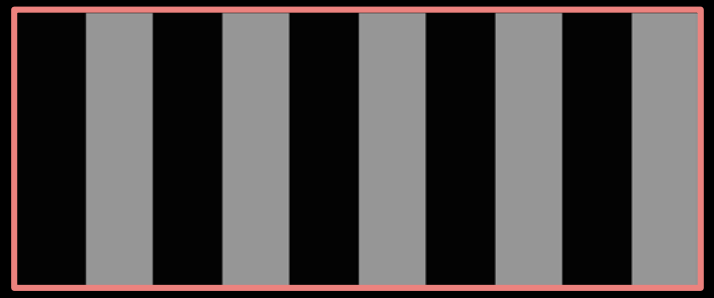
- This is not about percentage!
- MODULO: Give the remainder after division
- 7 % 3 =???

  Divide 7/3. Ignore the result, just take the remainder

- Modulo can be very useful

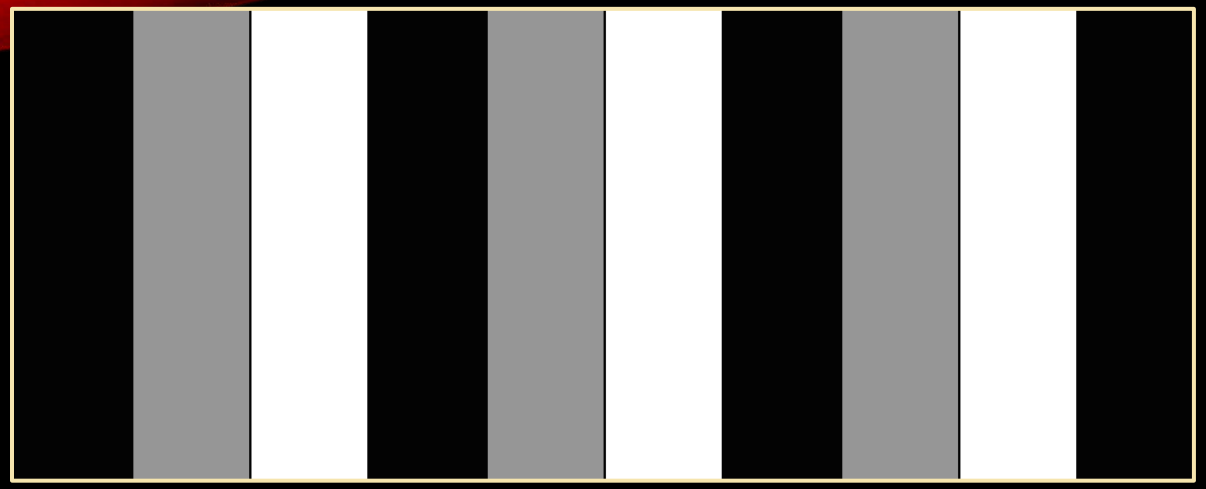


EXAMPLE

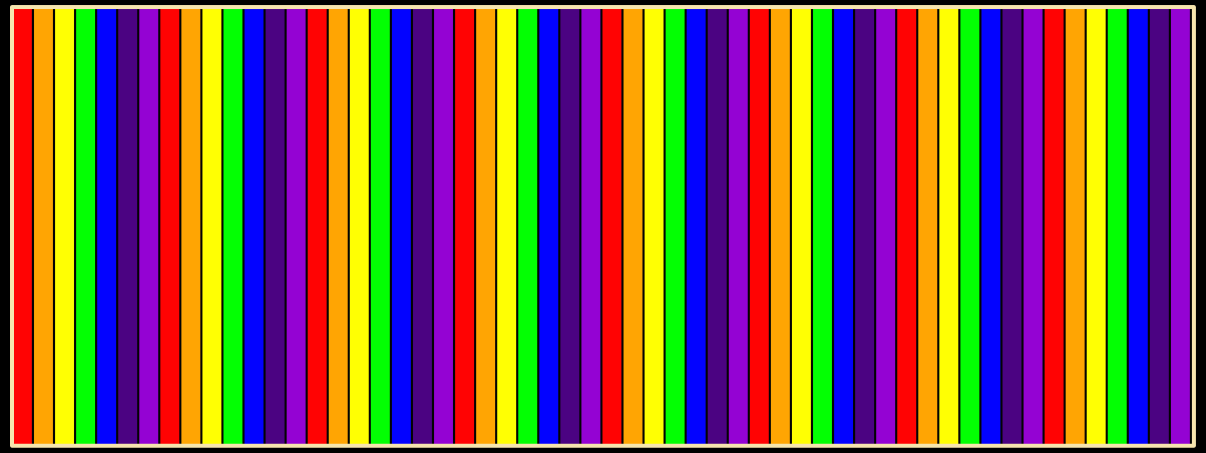


```
var x = 0;

function setup() {
  createCanvas(500, 200);
  background(220);
  frameRate(6); // <== lower frameRate to visualize the effect
}

function draw() {

  if (x%2 == 1) {
    fill(150);
  } else {
    fill(0);
  }

  rect(50*x, 0, 50, 200);
  x++;
}
```

JOB 1a



JOB 1b



JOB 1c

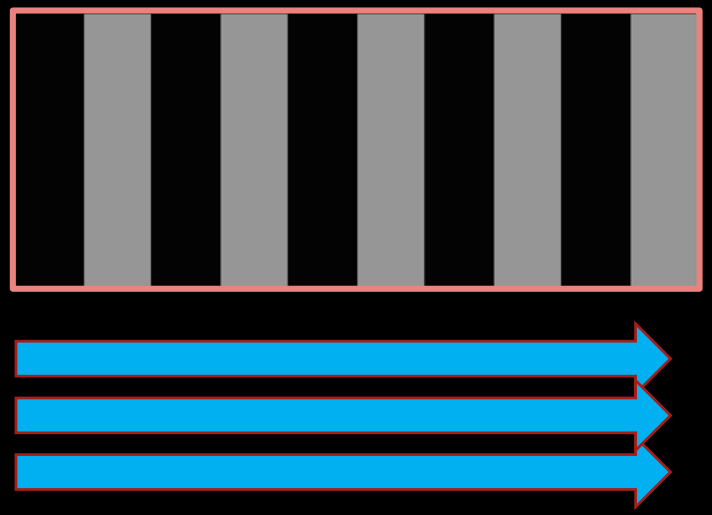
CYCLES

We can use frameCount and % to control animation cycles.

E.g. cycles of 10 frames:

Observe: When a cycle starts, we have `frameCount%10` equals `1`

EXAMPLE



Cycles repeated

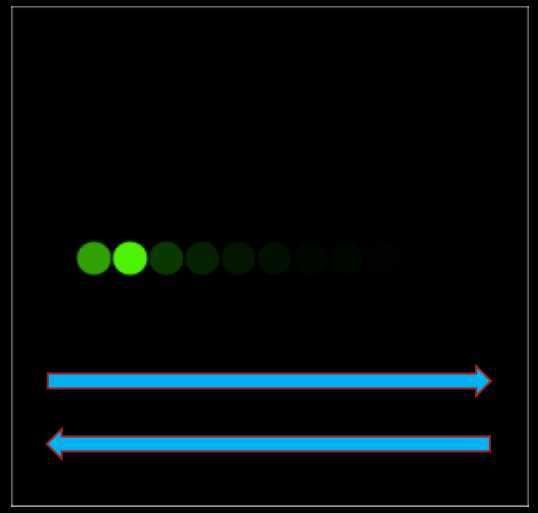
[Self-test exercise]
Modify the programs of Job 1 a-c so that they run in repeated cycles.

```
var x = 0;

function setup() {
  createCanvas(500, 200);
  background(220);
  frameRate(6);
}

function draw() {

  var f = frameCount % 10;

  if (f == 1) {
    x = 0;
    background(220);
  }

  if (x % 2 == 1) {
    fill(150);
  } else {
    fill(0);
  }
  rect(50 * x, 0, 50, 200);
  x++;
}
```

It is a new cycle, so we restore x and background for the new cycle

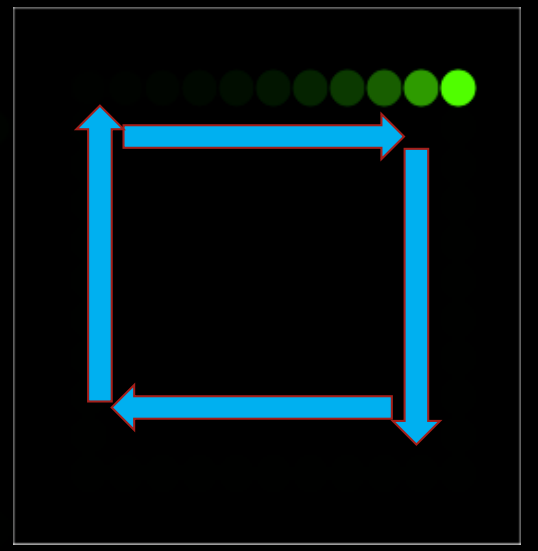
JOB 2

Given program:
Draw balls in cycles of 20 frames.
(left-to-right in 10 frames, followed by right-to-left in 10 frames)



Animation:
https://courses.cs.cityu.edu.hk/cs1103/public/Workshop03_Code/WS03_Slide13_Job2(Given)RollingBalls2Ways.gif

Your task:
Draw balls in cycles of 40 frames;
towards right, down, left, up, each in 10 frames.



Animation:
https://courses.cs.cityu.edu.hk/cs1103/public/Workshop03_Code/WS03_Slide13_Job2(Done)RollingBalls4Ways.gif

CS1103 Workshop 3

1. MODULO
2. DATA TYPES
3. LOOPS

CODING STYLE, DRAWING MODES AND TEXT CONTROL, %, **NUMBER AND TEXT AND BOOLEAN VALUES**, *WHILE-LOOP, FOR-LOOP*

DATA TYPES

We use data everywhere in our program.
The followings are the most basic types of data in JavaScript:

3 JavaScript primitive types:	Operations
(1) **numbers** : whole integers: ... -3, -2, -1, 0, 1, 2, 3, ...	+, -, *, /, % : results are numbers
floating-point values: eg. -0.001, 3.1416	>, >=, <, <=, ==, != : results are true/false
(2) **strings of text** - "…"	+ (i.e., join text together)
(3) **Booleans**: only 2 values: **true** and **false**	&&, ||, ! (i.e., not)

DATA TYPES

- The computer stores data using **binary bits** like 00110100..
- Each bit corresponds to the on/off state of a tiny electrical switch in the computer circuitry.
- Data, like **numbers**, **strings**, and the *Boolean* **true/false** values, are all coded in terms of 0's and 1's.
- For instance:
  129 is coded as 10000001
  Letter 'A' is coded as 01000001
  true is 1, false is 0

DATA TYPES

JOB 3 [Please finish in the handout]

What are the output of the following statements?

```
(a)   text(1234 + 20, 50, 50);      //output _____________________
(b)   text(1234 - 20, 50, 80);      //output _____________________
(c)   text("cs" + "1103", 50, 110); //output _____________________
(d)   text("1234" + "20", 50, 140); //output _____________________
(e)   text("cs" + 1103, 50, 170);   //output _____________________
(f)   text(1234 + "20", 50, 200);   //output _____________________
```

JOB 4 [Please finish in the handout]

Read the given programs and study the operations on Boolean variables

CS1103 Workshop 3

1. MODULO
2. DATA TYPES
3. LOOPS

CODING STYLE, DRAWING MODES AND TEXT CONTROL, %, NUMBER AND TEXT AND BOOLEAN VALUES, ***WHILE-LOOP, FOR-LOOP***

LOOP: WHILE

- What is the pattern?

Something-true-or-false

```
while(condition) {


}
```

PATTERN OF WHILE LOOP

- The pattern is similar to `if`

```
while(condition) {

    // do actions multiple times
}
```

```
if(condition) {

   // do actions once
}
```

PATTERN OF WHILE LOOP

- The pattern is similar to **`if`**
- When to stop?

```
while(condition) {

    // do actions multiple times
}   // stop when the condition is false
```

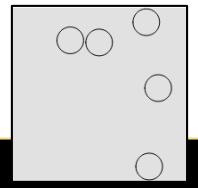
```
if(condition) {

   // do actions once
}
```

WHILE VS IF

The while-loop repeats looping

The loop can draw 5 ellipses

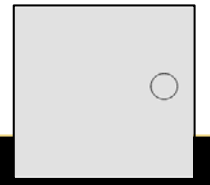
```
var i = 0;
while (i < 5) {
  ellipse(random(300), random(300), 30, 30);
  i = i + 1;
}
```



The if-statement does not loop

Can draw 1 ellipse only

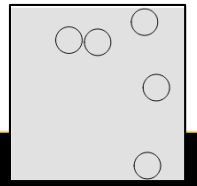
```
var i = 0;
if (i < 5) {
  ellipse(random(300), random(300), 30, 30);
  i = i + 1;
}
```

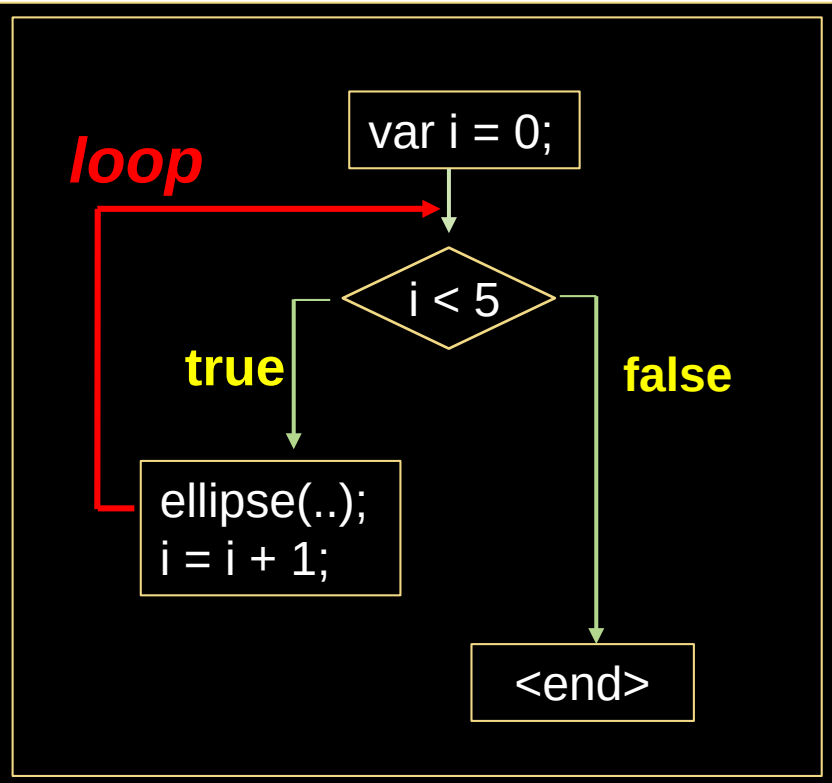


The while-loop repeats looping

The loop draws 5 ellipses

```
var i = 0;
while (i < 5) {
  ellipse(random(300), random(300), 30, 30);
  i = i + 1;
}
```

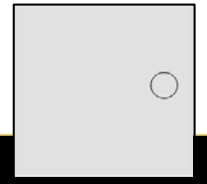


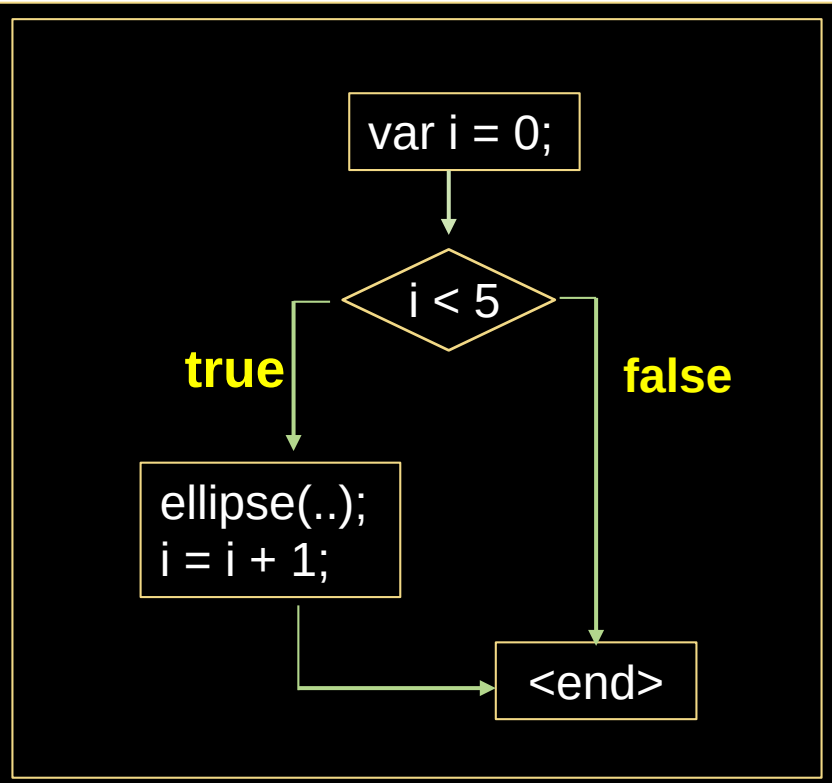


The if-statement does not loop

Draw 1 ellipse only

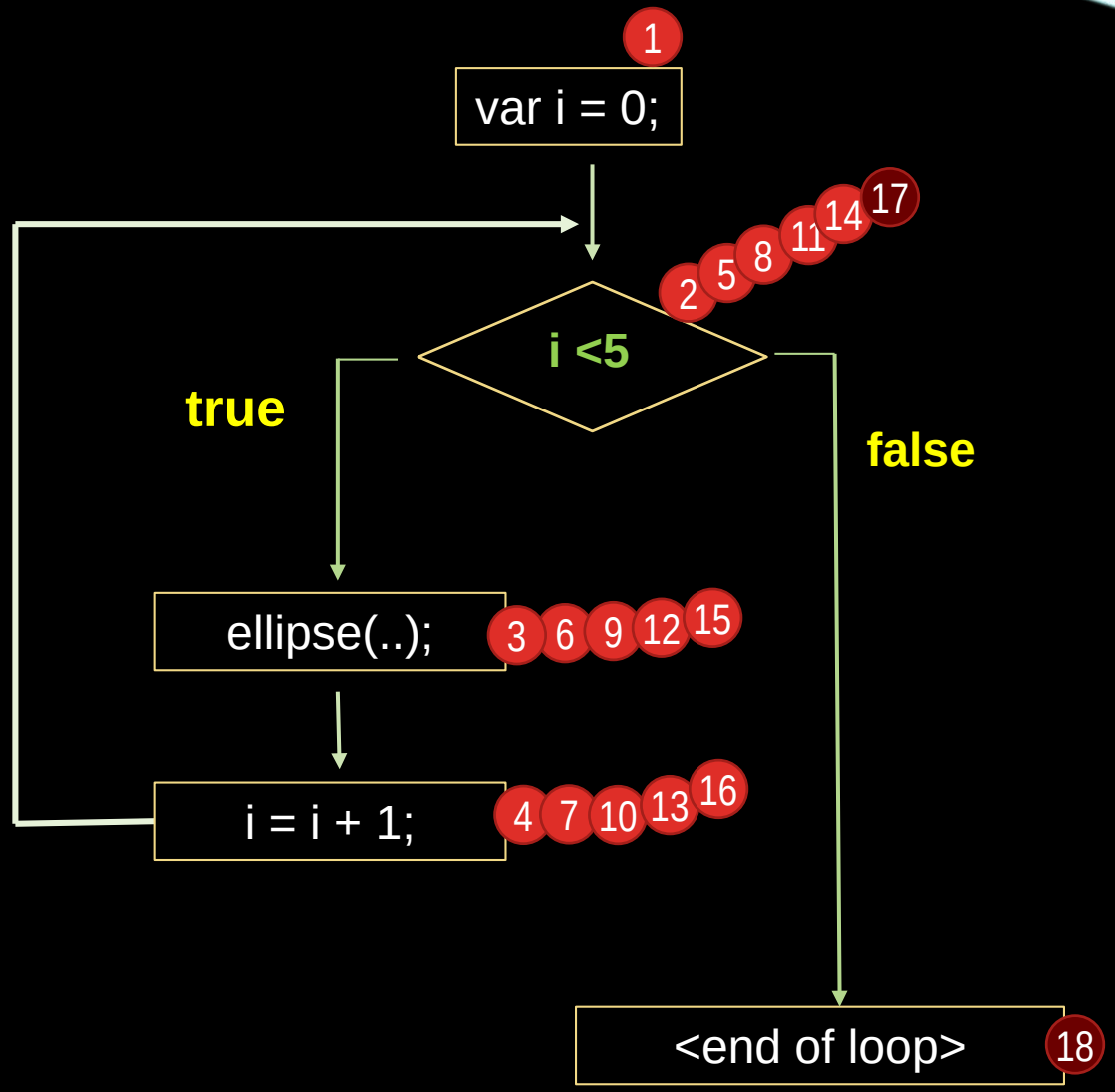
```
var i = 0;
if (i < 5) {
  ellipse(random(300), random(300), 30, 30);
  i = i + 1;
}
```





WHILE LOOP

```
var i = 0;
while (i < 5) {
  ellipse(random(300), random(300), 30, 30);
  i = i + 1;
}
```



Changes of `i`:

First `i` is initialized with 0. 1

Then `i` is increased by 1 five times. 4 7 10 13 16

Finally, when the value of `i` is 5. => The condition `i<5` is false 17 => The loop is ended. 18

WHILE LOOP

Let's trace the execution again:

initialization
Loop condition
Post action

```
var i = 0;
while (i < 5) {
  ellipse(random(300), random(300), 30, 30);
  i = i + 1;
}
```

Changes of `i`:

First `i` is initialized with 0.

Then `i` is increased by 1 five times.

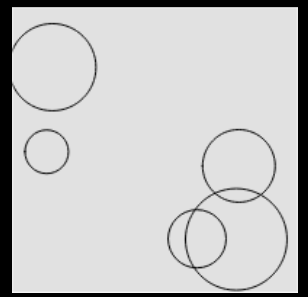
Finally, when the value of `i` is 5. => The condition `i<5` is false => The loop is ended.

LOOP: WHILE

Loop control variable

```
var i = 0;
while (i < 5) {
  ellipse(random(300), random(300), 30+i*10, 30+i*10);
  i++;  // same as i = i + 1;
}
```

Increasing sizes



Loop Control Variable

- The variable to control the execution of a loop is called a loop control variable.
- The variable name *i* is very often used by programmers for this purpose.
- To control more complicated or nested structures, *j* and *k* are also used.

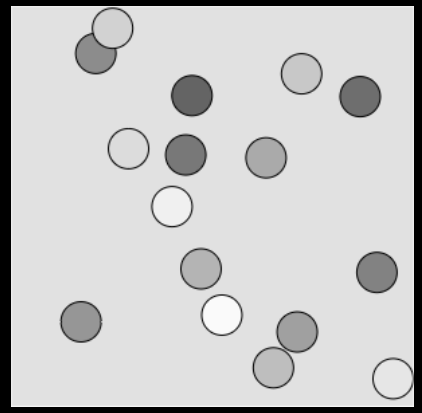
Writing i++

- To increase i by one:
  - ✓ `i = i+1;`
  - ✓ `i += 1;`
  - ✓ `i++;`

The loop control variable also provides a regularly changing value in the loop action.

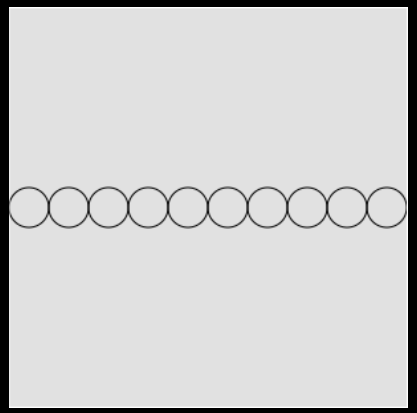
LOOP: WHILE

JOB 5



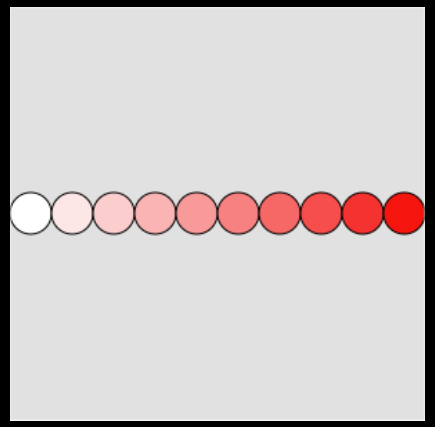
16 circles filled with increasing gray colors 100, 110, 120, and so on.

JOB 6



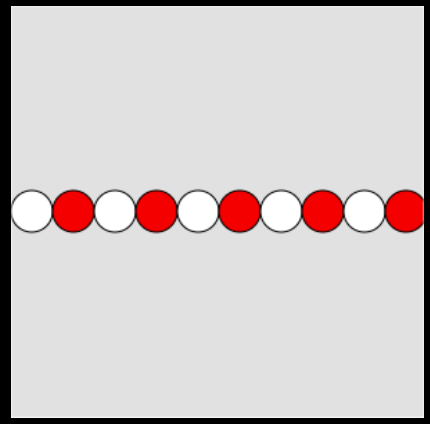
A row of 10 circles (30x30) at the middle of the canvas

JOB 6 EXTENSION



Filled with colors from white to red

JOB 7



Filled with white and red alternatively

Inside the while-loop, use if-else and % to determine the filling color!

CS1103 Workshop 3

1. MODULO
2. DATA TYPES
3. LOOPS

CODING STYLE, DRAWING MODES AND TEXT CONTROL, %, NUMBER AND TEXT AND BOOLEAN VALUES, *WHILE-LOOP*, ***FOR-LOOP***

FOR LOOP

- The function of for loop is same as while loop, but with different syntax
- The for loop is more compact and consists of
  - Initial condition
  - Loop condition
  - Post action after each iteration

```
var i = 0;
while (i < 5) {
  ellipse(random(300), random(300), 30, 30);
  i++;
}
```

Initial condition

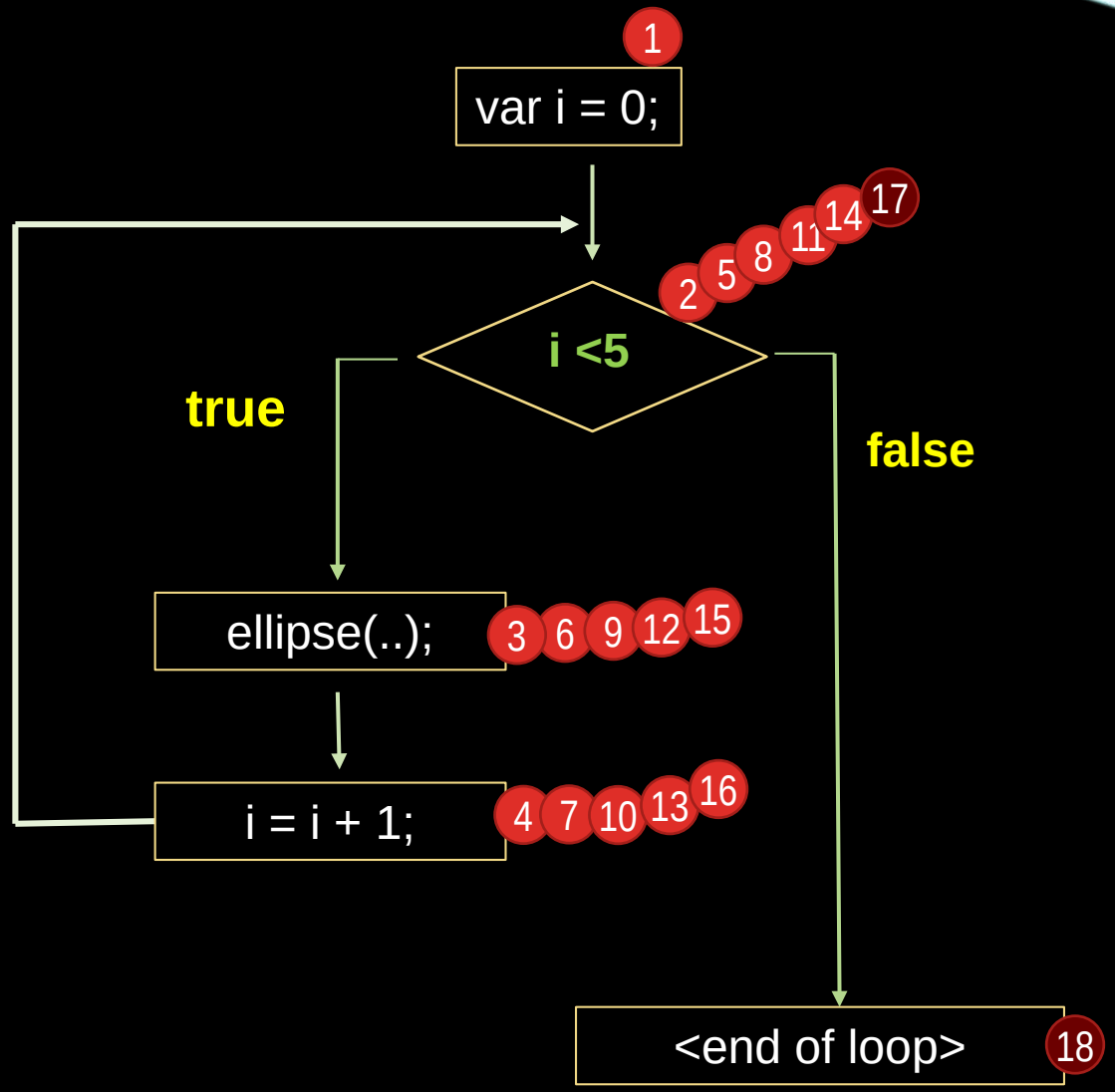
Loop condition

Post action

```
for (var i = 0; i<5; i++) {
  ellipse(random(300), random(300), 30, 30);
}
```

FOR LOOP

```
for (var i = 0; i<5; i++) {
  ellipse(random(300), random(300), 30, 30);
}
```



Changes of `i`:

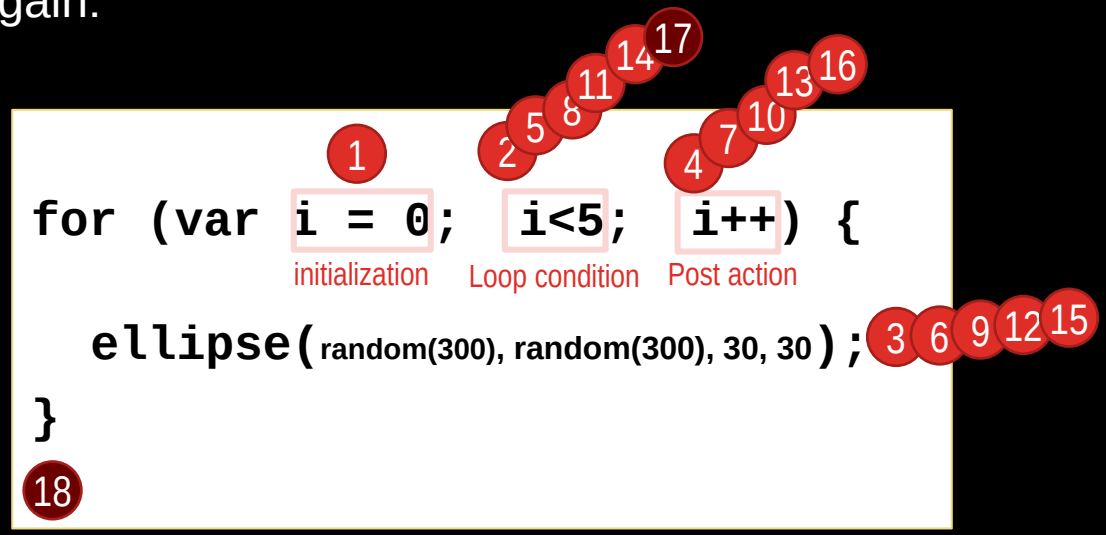
First `i` is initialized with 0. 1

Then `i` is increased by 1 five times. 4 7 10 13 16

Finally, when the value of `i` is 5. => The condition `i<5` is false 17 => The loop is ended. 18

FOR LOOP

Let's trace the execution again:



Changes of `i`:

First `i` is initialized with 0.

Then `i` is increased by 1 five times.

Finally, when the value of `i` is 5. => The condition `i<5` is false => The loop is ended.

FOR LOOP

```
function setup() {
  createCanvas(200, 150);
  background(225);

  stroke(0);
  line(50, 60, 50, 80);
  line(60, 60, 60, 80);
  line(70, 60, 70, 80);
  line(80, 60, 80, 80);
  line(90, 60, 90, 80);
  line(100, 60, 100, 80);
  line(110, 60, 110, 80);
  line(120, 60, 120, 80);
  line(130, 60, 130, 80);
  line(140, 60, 140, 80);
}
```

Job 8(a)

Rewrite the program using while loop

```
var x = 50;
while (x <= ______) {
  line(x, 60, x, 80);
  x _________;
}
```

Job 8(b)

Rewrite the program using **for loop**

[Self-test exercise]
Redo job 5 – 7 using for loop.

Given animation:
Three water drops falling down.
Each drop is a triangle + an ellipse.

triangle(x1,y1,x2,y2,x3,y3)

Job 9

Your task: rewrite using
(a) **while-loop**
(b) **for-loop**

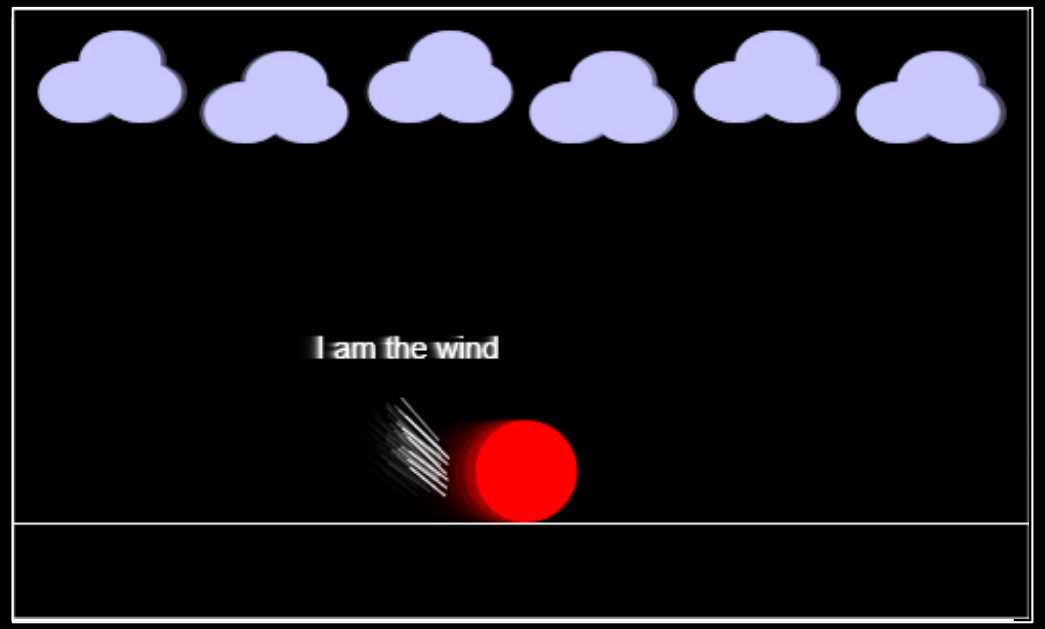
Further improvement – to repeat in cycles:
Change y +=20; to y = (y+20) % height;

```
var y=0;

function setup() {
   createCanvas(300, 300);
   fill(100, 100, 255);
   noStroke();
   frameRate(10);
}

function draw() {
   background(0);
   y += 20;

   var x = 100;
   triangle(x, y - 20, x - 10, y - 5, x + 10, y - 3);
   ellipse(x, y, 20, 20);

   x = x + 50;
   triangle(x, y - 20, x - 10, y - 5, x + 10, y - 3);
   ellipse(x, y, 20, 20);

   x = x + 50;
   triangle(x, y - 20, x - 10, y - 5, x + 10, y - 3);
   ellipse(x, y, 20, 20);
}
```

Animation:
https://courses.cs.cityu.edu.hk/cs1103/public/Workshop03_Code/WS03_Slide32_Job9_WaterDrops.gif

DISCUSSION

How is this animation created?



Animation:
https://courses.cs.cityu.edu.hk/cs1103/public/Workshop03_Code/WS03_Slide34_WindBlowingBall.gif

Assignment of this lesson:

Create a program according to the requirement below.

Submit it on Canvas.

Due: Before the start of your next lesson (Except that for LA1 (Mon) classes: On/Before Sep 26)

Create the program as shown in the animation.

Details:

1. Draw 5 Zoogs using a loop. (1.5 marks)
2. Draw 20 circle slots in a row using a loop. (1 marks)
3. Roll a ball in the circle slots and bounce on left and right margins. (1 mark)
4. Make Zoog's eyes follow the rolling ball (0.5 mark)
5. Add one or more creative features / animations. (1 mark)

Animation:
https://courses.cs.cityu.edu.hk/cs1103/public/Workshop03_AsgAnimation