

(I) Review: Rules and conventions for naming of parameters, variables, and functions

- Only allow uppercase and lowercase letters, digits, and underscore (_). Examples: **result1**, **result2**, **x**, **bonus_Eng**
- CANNOT start with 0 - 9. CANNOT be a keyword. Eg. **9To5** and **var** are wrong
- The names should be descriptive. Do not use names like **temp1**, **temp2**.
Good examples of short names: **c** for *coursework*; **e** for *exam*.

Reminder: **better readability = easy to maintain and less careless mistakes**

- Often started in lowercase.
- Recall: *camelCase*  - A common naming style, like **strokeWeight** (First word starts with lowercase, uppercase for others)

(II) Warm-up Exercises

Q1. Recall:

- =** : "=" is known as the *Assignment operator* - It changes the LHS to the given value provided on the right.
e.g. `x = x + 1;` | `x = random(300);`
- +=** : Apply an increment to a variable.
e.g. `x += 10;` means `x = x + 10;`
Similar short-hands: `-=`, `*=`, `/=`, `%=`
- ++**, **--** : Add or subtract by one.
e.g. `x++;` means `x = x + 1;`, `x--;` means `x = x - 1;`

Your tasks: (i) Rewrite `x = x + 5;` using **+=**

(ii) Rewrite `x = x + 5;` using **++**

Q2. (a) What will be shown after running the following program for some seconds?

```
function setup() {
  createCanvas(300, 300);
  background(200); noFill(); stroke(255,0,0);
}

function draw() {
  var x = random(width);
  var y = random(height);

  if (x < 100 || x > 200) {
    triangle(x, y, x - 10, y + 10, x + 10, y + 10); //tiny triangle
  }

  if (x > 100) {
    if (x < 200) {
      ellipse(x, y, 10, 10); //tiny ellipse
    }
  }
}
```

Draw the answer:

Download Q2.html and check!

(b) Rewrite the ellipse part using one if statement

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

  if (x < 100 || x > 200) {
    triangle(x, y, x-10, y+10, x+10, y+10);
  }

  if (_____) {
    ellipse(x, y, 10, 10);
  }
}
```

(c) Rewrite all drawings using one if-else statement

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

  _____ {
    triangle(x, y, x-10, y+10, x+10, y+10);
  } _____ {
    ellipse(x, y, 10, 10);
  }
}
```

Q3. (a) What will be the output of the following program?

Your answer:

```
function setup() {  
  createCanvas(250, 80);  
  background(230);  
  noFill(); stroke(255, 0, 0);  
  
  var gap = 20;  
  var x = 0;  
  for (var i = 1; i <= 10; i++) {  
    x = x + gap;  
    ellipse(x, 40, 10, 10);  
  }  
}
```

(b) Programmers often start counting from zero.

Now, rewrite the code of part (a) to make `i` start from zero.

```
function setup() {  
  createCanvas(250, 80);  
  background(230);  
  noFill(); stroke(255, 0, 0);  
  
  var gap = 20;  
  var x = 0;  
  for (var i = 0; i < 10; i++) {  
    x = x + gap;  
    ellipse(x, 40, 10, 10); //tiny ellipse  
  }  
}
```

(c) The declaration of the variable `x` has moved as follows. What will be the output? Why?

Your answer:

Try and check!

```
function setup() {  
  createCanvas(250, 80);  
  background(230);  
  noFill(); stroke(255, 0, 0);  
  
  var gap = 20;  
  for (var i = 1; i <= 10; i++) {  
    var x = 0;  
    x = x + gap;  
    ellipse(x, 40, 10, 10); //tiny ellipse  
  }  
}
```

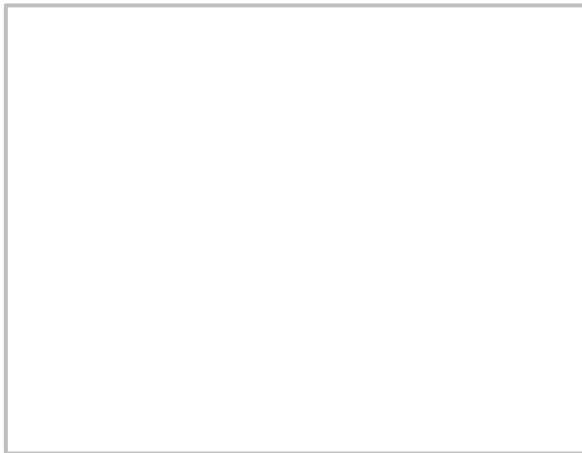
Q4. Download Q4.html and open it. The program draws a vertical line using **points of random colors**:

```
function setup() {
  createCanvas(200, 200);
  background(230);
  strokeWeight(3); //control the size of each point.

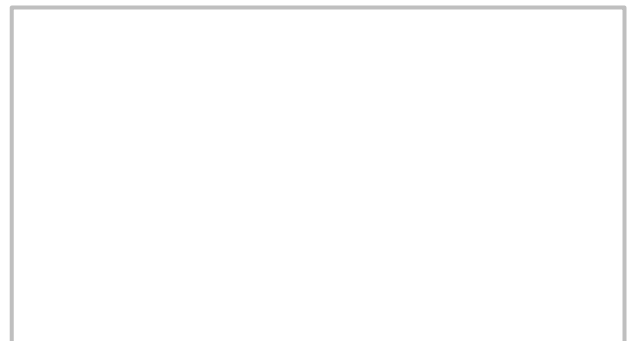
  for (var y = 20; y < 180; y++) {
    stroke(random(255));
    point(100, y);
  }
}
```

(a) Use 2 different approaches to draw one more horizontal line, like this: .

Approach (i) Add one more loop.

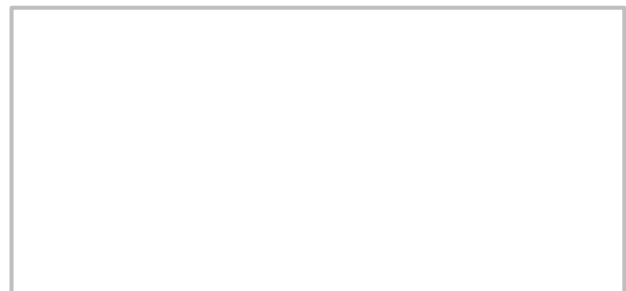
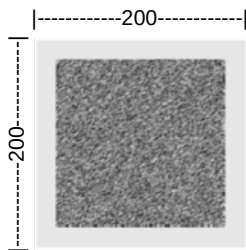


Approach (ii) Modify the current loop.



(b) Use nested for-loops to fill a big square inside the canvas of 200x200:

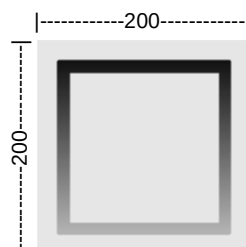
top-left = (20,20),
bottom-right = (180,180)



Guideline for beginners

- Use y to control **outer loop**: to paint row by row
- Use x to control **inner loop**: to paint each dot in a row

(c) Modify your program in part (b) to draw a hollow square.
The rim is 10 pixels wide and colored in gradient.



Q5 [Self-test Homework] *When Frame Animation works with Loop*

Your task: [First] **Tracing** - By following the logic of the programs below, tell what their differences.
[Next] Run the programs to check your answer.

//WS05_GettingReadyQ2a.html

```
function setup() {  
  createCanvas(300, 300);  
  noFill();  
  stroke(255, 0, 0);  
}  
  
function draw() {  
  background(225);  
  
  var k = 0;  
  
  while (k <= width) {  
    ellipse(k, height/2, frameCount%width, frameCount%height);  
    k += 50;  
  }  
}
```

//WS05_GettingReadyQ2b.html

```
function setup() {  
  createCanvas(300, 300);  
  noFill();  
  stroke(255, 0, 0);  
}  
  
function draw() {  
  background(225);  
  
  var k = 0;  
  
  while (k <= width) {  
    ellipse(frameCount%width, height/2, k, k);  
    k += 50;  
  }  
}
```

(III) Working out Class Exercises

Job 1 Given the function `drawZoog()` with 5 parameters, write a program that draws the picture below.

```
function setup() {
  createCanvas(500, 300);
  background(200);

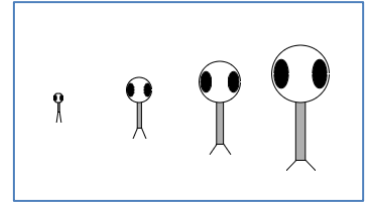
}

// x and y specify location
// w and h specify width and height of zoog
// eyeSize specify the size of zoog's eyes
function drawZoog(x, y, w, h, eyeSize) {
  // Draw Zoog's body
  stroke(0);
  fill(175);
  rectMode(CENTER);
  rect(x, y, w/6, h*2);

  // Draw Zoog's head
  fill(255);
  ellipse(x, y-h/2, w, h);

  // Draw Zoog's eyes
  fill(0);
  ellipse(x-w/3, y-h/2, eyeSize, eyeSize*2);
  ellipse(x+w/3, y-h/2, eyeSize, eyeSize*2);

  // Draw Zoog's legs
  stroke(0);
  line(x-w/12, y+h, x-w/4, y+h+10);
  line(x+w/12, y+h, x+w/4, y+h+10);
}
```



Job02 Given program: Bouncing left and right

```
var circleX = 25;
var circleY = 100;
var direction = 1;

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

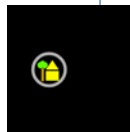
function draw() {
  //paint
  background(0);
  stroke(175);
  strokeWeight(5);
  fill(0);
  ellipse(circleX, circleY, 50, 50);
  drawHouse(circleX + 5, circleY);
  drawTree(circleX - 10, circleY);

  //move
  circleX = circleX + direction;

  //bounce
  if (circleX > width - 25 || circleX < 25) {
    direction = direction * -1;
  }
}

function drawHouse(x, y) {
  ..
}

function drawTree(x, y) {
  ..
}
```



New structure - 3 functions: `paint()`, `move()`, `bounce()`



Discussion: How does the program work? (Slide #21) How to work out Job 03?

```

var x, y, isPressed;

function setup() {
  createCanvas(240, 360);
  ellipseMode(CENTER);
  rectMode(CENTER);

  isPressed = false;
}

function draw() {
  background(225);

  if (isPressed == false) {
    x = mouseX;
    y = mouseY;
  }

  drawEyes();
  drawLegs();

  placeLego();
  drawEyes(); //redraw eyes on lego

  textSize(10);
  noStroke();
  fill(255, 0, 0);
  text("x: " + x + ", y: " + y + ", isPressed: " + isPressed, 5, 355);
}

function drawEyes() {
  fill(0);
  ellipse(101, 160, 16, 32);
  ellipse(139, 160, 16, 32);
}

function drawLegs() {
  stroke(0);
  line(110, 240, 100, 250);
  line(130, 240, 140, 250);
}

function placeLego() {
  // Draw body and head (mouse location)

  stroke(0);
  fill(175);
  rect(x, y, 20, 100);

  stroke(0);
  fill(255);
  ellipse(x, y - 30, 60, 60);
}

function mousePressed() {
  isPressed = true;
}

```

