Update variable
hyesung.park authored 1 week ago

variable.md 5.12 KiB

Variables

Getting started

Think about a variable like a container. Just like you use a container to hold things. You use a variable to hold information in a program. For example, you can have a container called "age" and you can put your age inside it.

In Java, when we want to create a container, we have to give it a name and tell a computer what kind of information it will hold. For example, you can create a variable named **age** and tell the computer that it will hold a number.

Here's an example of how you would write that in Java.

```
int age;
```

The word "int" is short for "integer," which is a type of number. So we are telling the computer that this variable will hold an integer.

Once you have a container, you can put information inside it. Here is how you would put your age inside the **age** container.

```
age = 20;
```

So now the "age" container holds the value 20. And you can use this container anywhere in your program, just like you can use any other container to hold things.

Alreay got it? Then, open the processing, a programming language. The processing is a development environment for creating images, animations, and interactive graphics. First, let's draw a line. To draw a line, you will need to set the properties, such as the size of the canvas, the background color, stroke color, and stroke weight. In this program, the code declares three integer variables named **a,b, and c**. They are assigned value **200,200, and 200**.

The first line is drawn using the line() function, which draws a line between two points. The **x and y** coordinates of the start and end points are specified using the variables **a,b,c**.

```
size(640, 360);
background(0);
stroke(153);
strokeWeight(4);
strokeCap(SQUARE);

int a = 200;
int b = 200;
int c = 200;

//line(x-coordinate of the start, y-coordinate of the start, x-coordinate of end points, y-coordinate of
line(a, b, a + c, b);
```

The result of this program is a single line drawn from the point (200, 200) to the point (400, 200).

Then,please add the following statement.

```
line(a, b+10, a+c, b+10);
```

And run the program. How many lines do you see now?

Then, follow the direction and see if you can complete this program accordingly.

Once you complete this program, you will have the first set of two lines has a start point at the coordinate **(a,b)** and **(a, b+10)**, and an end point at the coordinates **(a + c, b)** and **(a + c, b + 10)**. The x coordinate of the end point is **a+c**, which is equal to **(200 + 200 = 400)**, and the **y** coordinate of each end point is **b** and **(b + 10)**.

The second set of lines are similar to the first set, but with an updated value of **b**. The value of is updated by subtracting its current value from the height of the canvas, effectively flipping it across the x-axis. The new value of **b** is equal to **(height - b)**, which is equal to **(360- 200 = 160)**. The start point of the second set of lines at the coordinates **(a + c, b)** and **(a + c , b + 10)**, and the end point of each line is at **(a + 2c , b)** and **(a + 2c, b + 10)**

Can you solve this program?

Answer:

```
size(640, 360);
background(0);
stroke(153);
strokeWeight(4);
strokeCap(SQUARE);

int a = 200;
int b = 200;
int c = 200;

line(a, b, a+c, b);
line(a, b+10, a+c, b+10);

 a = a + c;
 b = height-b;

 line(a, b, a+c, b);
 line(a, b+10, a+c, b+10);
```

Exercise 1: Write a program named passing a Wall.pde procesing program.

1. set the **size of the canvas** to **(500 by 500)** pixels.

2. In the **setup()** function, set the background to black.

3. In the **draw()** function, set the stroke color to **153**, with a weight of **1**

4. Declare three variables **a,b,c,d**, which are used for specifying the position and size of the objects to be drawn. Initialize **four** variables with following values. assign **width/3 to a, assign height/2 to b, assign (2 * a) to c, and assign b to d**

5. Then, copy the following code.

```

if(mouseX >= 100 && mouseY < 100) {
  fill(242, 99, 42, 100);
}

if(mouseX < 100 && mouseY >= 100) {

    fill(146, 21, 175, 100);
  }

```

This tells you that if the mouse cursor is within the top-left **100X100** pixels of the canvas, the fill color of the object is set to an orange color with transparency **100**. Then, if the moust cursor is outside of the top-left **100X100** pixels, the fill color is set to purple color with transparency **100**.

6. Then, write a statement to draw a line from the point **(a,b) to the point (c,d) with a stroke weight of 3**.

7. Then, write a statement to draw an **ellipse(circle)** at the position of the mouse cursor, with a **diameter of 50 pixels**.

Answer:

```
//color bg = color(250, 100, 200);
void setup() {

 // background(bg);
  size(500, 500);
}
```

```
 void draw(){

background(0);
stroke(153);
strokeWeight(1);

int a  = width/3;
int b = height/2;
int c = 2 * a;
int d = b;


if(mouseX >= 100 && mouseY < 100) {
  fill(242, 99, 42, 100);
}

if(mouseX < 100 && mouseY >= 100) {

    fill(146, 21, 175, 100);
  }
strokeWeight(3);
line(a, b, c, d);
ellipse(mouseX, mouseY, 50,50);

}
```