

Program Structure Recap (Active Sketch)

```
// declare variables
void setup () {           // ...stuff to do once at the beginning goes here...
  // open window
  // initialize variables
}
void draw () {           // ...stuff to do for each frame goes here...
  // draw frame
  // update variables
}
```

Variable Recap

- declaring variables

```
type name;           // where type is the type of the variable and
                      // name is its name
```

- types

```
int                   // whole number
float                 // numbers with decimal points
boolean              // true or false
```

- assignment statements – for initialization and update

```
name = value;       // where name is the name of the variable whose
                      // value is to be set, and value is the new value
                      // (value can be a literal or an expression)
```

Animation and Variable Questions

- what changes from one frame to the next? → one variable for each thing
- for each variable –
 - what kind of value? → type for declaration
 - where does the thing start? → initialization
 - how does the value change from one frame to the next? → update

At the End of Class

Hand in whatever you have done during class, even if a sketch is incomplete.

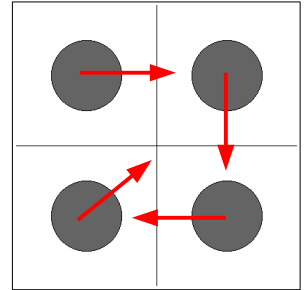
- Make sure each sketch is named as directed and has a comment with the names of your group. Also be sure to save your sketches! (in Linux, this should be in your sketchbook **~/cs120/sketchbook**)
- Copy the entire directory for each sketch (not only the .pde file) into your handin directory (**/classes/cs120/handin/username**). You only need to hand in one copy for the group. (If you are running Processing on your computer instead of using the Linux virtual desktop, you will need to use FileZilla to copy the sketches.)

For all exercises, include a comment with the names of your group at the beginning of the sketch.

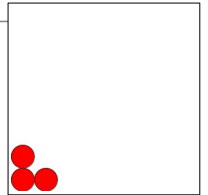
Exercises

Do **either** #1-3 or #4 – choose #1-3 if you didn't complete those exercises on Monday and #4 if you did – and then continue on to #5.

1. Create a new sketch named **sketch_240911a** where a circle starts in the lower left corner of the window and moves diagonally up and right.
2. Modify the sketch so the circle starts with a width and height of 1, and grows as it moves.
3. Modify the sketch so the circle starts filled in black (with a black outline) and then gradually gets redder so that by the time the circle reaches the upper right corner of the window, it is completely red.
4. Create a new sketch named **sketch_240911b** which has the pattern of circles and lines shown, then animate the circles so that they move in different directions as shown by the arrows. (You don't need to draw the arrows or the box around the outside. Also, each circle only moves in one direction.) Make the drawing window 600x600.



5. Create a new sketch named **sketch_240911c** where three circles start as shown in the lower left corner of the window and the whole pattern moves diagonally up and right. (How many variables do you need? Can you use only two?)
6. Modify the sketch so that the circle in the corner changes color from red to white over time. (How many variables do you need?)
7. Modify the sketch so it works no matter what size the drawing window is – that is, the circles always start in the lower left corner without having to change more than the size of the window. You will need one or both of the following system variables:
 - `width, height` - dimensions of the drawing window (only valid after `size(..., ...)` has been executed)



If you have time, try the following –

The reading introduced the `random()` function which can be used to generate random values: `random(n)` returns a value between 0 and n (including n). You can substitute `random(n)` (with something substituted for n) anywhere a number (specifically, a float value) is expected. For example

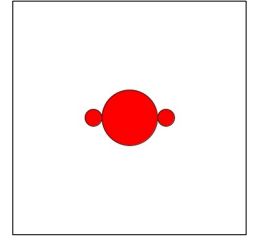
```
fill(random(255));
```

sets the fill color to a random shade of gray between 0 and 255.

- Create a new sketch called **sketch_240911d** which has a single circle centered in the drawing window. (Write your sketch so that it works with any size window.) Use a variation on the `fill` example given above to choose a random color (rather than a random shade of gray) for the ellipse – that is, the red and the green and the blue components of the color should each be separate random numbers. Run the sketch – what do you see?
- Modify the sketch so that a random color is chosen for the circle each time the sketch is run, but then it stays a fixed color – no flickering. Hint: this is still animation because something changes over time – in this case, the circle's color changes from “not yet initialized” (the circle's color goes from “not yet chosen” to a particular color at the very beginning of the sketch even though it doesn't change again after that) – and animation calls for variables!

If you still have time –

- Create a new sketch called **sketch_240911e** which contains a three-circle pattern as shown – the pattern should be centered in the drawing window (write the sketch so that it works with any size window) and the center circle should start with a width and height of 1 and then grow over time, with the other two circles staying the same size and always just touching the center circle.



- The following statements draw the car shown:

```
rectMode(CENTER);  
ellipseMode(CENTER);  
  
fill(255,0,0);  
stroke(255,0,0);  
rect(100,285,60,20);  
rect(100,270,30,10);  
fill(0);  
stroke(0);  
ellipse(80,295,15,15);  
ellipse(120,295,15,15);
```



Create a new sketch named **sketch_240911f** where the car starts at the right side of the window and then drives off towards the left side of the window. (What variable(s) do you need? Can you do it with just one variable?)

- Modify the sketch so that the car gains speed as it moves.
- Modify the sketch so that the car also shrinks as it moves.