

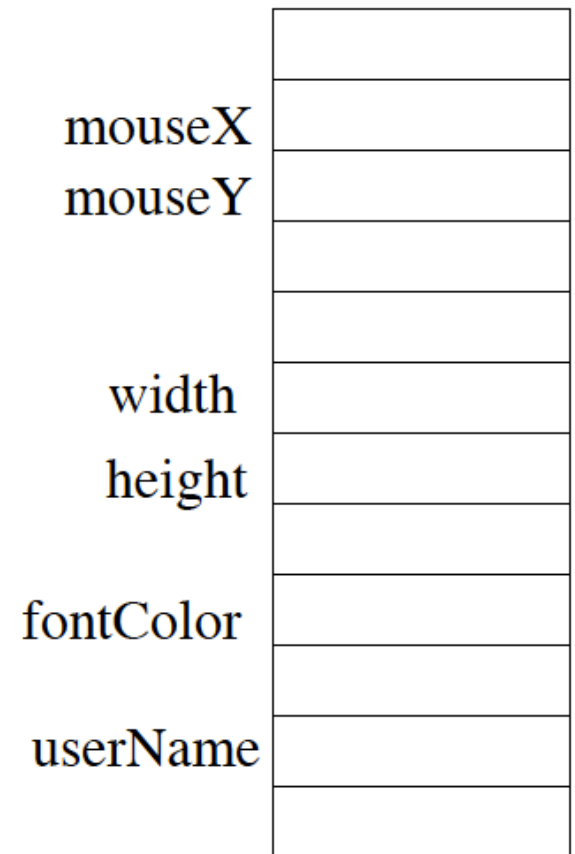
# Chapter 4

# Summary/review

- coordinate system
- basic drawing commands and their parameters (rect, line, ellipse, background, stroke, fill)
- color model - RGB + alpha
- Processing IDE - entering/saving/running
- top to bottom statement execution - order matters
- function call syntax (e.g. `rect(10,20,30,40);` )
- `println()` for debugging
- comments
- `setup` & `draw`
- system variables - `mouseX`, `pmouseY`, `width`, `height`, ...
- `translate`
- Did I leave off something important?

# What is a variable?

- A named location in the computer's memory.
- A variable stores values



# Variables

- store/remember values
- can be changed
- must be declared to store a particular kind of value (e.g. whole number, fraction, character, color, image, boolean)
- should have a descriptive name
- start with letter
- then also include numbers
- no spaces

```
void setup() {  
    size(500,500);  
    noFill();  
}  
  
int diameter = 0;  
  
void draw() {  
    ellipse(width/2, height/2, diameter,diameter);  
    diameter = diameter + 10;  
}
```

---

What does this display?

- A) many nested circles (like a target) growing in size
- B) a white circle growing in size
- C) a black circle growing in size
- D) a pulsing images of circles growing and shrinking
- E) nothing (a circle with diameter 0)

```
// draw a simple house
int houseX = 50;
int houseY = 100;
int houseWidth = 100;
rect(houseX, houseY, houseWidth, houseWidth/2);
triangle(houseX, houseY, houseX+houseWidth/2,
houseY-houseWidth/2, houseX+houseWidth, houseY);
```

```
// draw a simple house
int houseX = 50;
int houseY = 100;
int houseWidth = 100;
rect(houseX, houseY, houseWidth, houseWidth/2);
triangle(houseX, houseY, houseX+houseWidth/2,
houseY-houseWidth/2, houseX+houseWidth, houseY);

// draw another house - lines are copied from above
houseX = 200;
houseWidth = 50;
rect(houseX, houseY, houseWidth, houseWidth/2);
triangle(houseX, houseY, houseX+houseWidth/2,
houseY-houseWidth/2, houseX+houseWidth, houseY);

// draw another house - lines are copied from above
houseX = 300;
houseY = 200;
rect(houseX, houseY, houseWidth, houseWidth/2);
triangle(houseX, houseY, houseX+houseWidth/2,
houseY-houseWidth/2, houseX+houseWidth, houseY);
```

```
void setup() {
    size(400,400);
}
// draw a simple house
float houseX = 50;
float houseY = 100;
float houseWidth = 100;
void draw() {
    rect(houseX, houseY, houseWidth, houseWidth/2);
    triangle(houseX, houseY, houseX+houseWidth/2,
houseY-houseWidth/2, houseX+houseWidth, houseY);
    houseX = houseX + 1.5*houseWidth;
    houseY = houseY + 0.75*houseWidth;
    houseWidth = houseWidth / 2;
}
```



```
/* Write a program that shows a 10x10 circle moving from  
the top center of the display down to the bottom center.  
*/
```

```
/* Write a program that shows a 10x10 circle moving from  
the top center of the display down to the bottom center.  
*/  
//Can you make it go slower or faster?
```

```
void setup() {  
    size(200, 200);  
}  
void draw() {  
    background(255);  
    int xPos = 0;  
    ellipse(xPos, height/2, 20, 20);  
    xPos = xPos + 1;  
}
```

---

What does this draw?

- A) circle moving across the screen left to right
- B) circle moving across the screen right to left
- C) circle moving across the screen top to bottom
- D) circle moving across the screen bottom to top
- E) half-circle on the left edge not moving

# Declaring v. Initializing v. Assigning

- `int carFront; // declare`
- `int carFront = 100; // declare and initialize`
- `carFront = carFront - 1; // assign`

# Modulus (%)

- $17 / 3$  in Java is 5 (goes in evenly 5 times)
- $17 \% 3$  in Java is 2 ( $17 - 3 * 5 = 2$ )
- What is  $8 \% 5$ ?
- What is  $150 \% 10$ ?
- What is  $17 \% 2$ ?

```
int counter = 0;
void draw() {
    background(255);
    int xPos = width - counter*width;
    int yPos = counter*height;
    fill(255,0,0);
    ellipse(xPos, yPos, 5, 5);
    fill(0,255,0);
    ellipse((2*counter)%width, (2*counter)%height, 5, 5);
    counter = counter + 1;
}
```

- 
- A. The red one is moving faster.
- B. One is moving top to bottom following a diagonal, the other is moving bottom to top following a diagonal.
- C. Both A and B
- D. Neither A nor B

```
int frontOfCar = 100;
int topOfCar = 100;
int lengthOfCar = 200;
int carBodyHeight = lengthOfCar/4;
int wheelDiameter = lengthOfCar/4;

// draw the body
rect(frontOfCar,topOfCar,lengthOfCar,carBodyHeight);

// draw the wheels
ellipse(frontOfCar+wheelDiameter,topOfCar+carBodyHeight,
wheelDiameter,wheelDiameter);
ellipse(frontOfCar+lengthOfCar-wheelDiameter,
topOfCar+carBodyHeight, wheelDiameter,wheelDiameter);

// draw the windshield
// use carBodyHeight to also control the size of the windshield
line(frontOfCar+carBodyHeight,topOfCar,
frontOfCar+2*carBodyHeight, carBodyHeight);
```

```
int frontOfCar = 100;
int topOfCar = 100;
int lengthOfCar = 200;
int carBodyHeight = lengthOfCar/4;
int wheelDiameter = lengthOfCar/4;
void draw() {
    background(255); // 255 means white
    fill(255); // the inside of the body should be white
    // draw the body
    rect(frontOfCar,topOfCar,lengthOfCar,carBodyHeight);
    // draw the wheels
    ellipse(frontOfCar+wheelDiameter,topOfCar+carBodyHeight,
wheelDiameter,wheelDiameter);
    ellipse(frontOfCar+lengthOfCar-wheelDiameter,
topOfCar+carBodyHeight,wheelDiameter,wheelDiameter);
    line(frontOfCar+carBodyHeight, topOfCar,
frontOfCar+2*carBodyHeight, topOfCar-carBodyHeight);
frontOfCar = frontOfCar - 1; // move car to the left
}
```



```

int frontOfCar = 100;
int topOfCar = 100;
int lengthOfCar = 100;
int carBodyHeight = lengthOfCar/4;
int wheelDiameter = lengthOfCar/4;
void draw() {
    println(lengthOfCar);
    println(carBodyHeight);
    println(wheelDiameter);
    background(255); // 255 means white
    fill(255); // the inside of the body should be white
    // draw the body
    . . .
    lengthOfCar = lengthOfCar - 1; // make the car shrink
}

```

---

What is printed when the 2nd frame is drawn?

- |       |       |       |        |
|-------|-------|-------|--------|
| A) 99 | B) 99 | C) 99 | D) 100 |
| 25    | 24    | 24.75 | 25     |
| 25    | 24    | 24.75 | 25     |

# Primitive Types

- boolean - true or false
- char - 'a', 'b', 'c', ...
- byte - small integer -128 to 127
- short - bigger integer -32768 to 32767
- int - even bigger integer +/- 2 billion
- long - really big integer
- float - numbers with fractional parts 3.1415
- double - like float but more precision

# System Variables (Processing)

- mouseX, mouseY
- pmouseX, pmouseY
- width, height
- frameCount

```
void setup() {  
    size(400, 400);  
}  
void draw() {  
    fill(255-abs(mouseX-pmouseX));  
    rect(pmouseX, pmouseY, mouseX, mouseY);  
}
```

```
void setup() {
    size(400, 400);
    _____ // position A
}
_____ // position B
void draw() {
    _____ // position C
    fill(255-abs(mouseX-pmouseX));
    rect(pmouseX, pmouseY, mouseX, mouseY);
    _____ // position D
}
```

Where should the line "background(255);" be placed so that the sketch shows just a single moving rectangle?

Choose option E if it would work with either C or D.

# Type Conversion/Casting

```
int houseWidth;
```

```
houseWidth = houseWidth * .75;
```

# What is the value of x in...

- `int x, y = 7;`
- `x = y / 8;`
- `x = (-y) / 8;`
- `x = y / 3;`
- `x = y % 3;`
- `x = int(y*1.5);`
- `x = (int)y*1.5;`

# What is the value of z in...

```
int x = 6, y = 10;
```

```
float z;
```

```
z = x / y;
```

A) 0

B) .6

C) 1

D) none of the above - it is a syntax error



```
void setup() {  
    size(500,500);  
    fill(255,0, 0);  
}  
  
void draw() {  
    char letter = (char)('A' +random(26));  
    text(letter, mouseX, mouseY);  
}
```

```
/*Author: Dustin Adams
```

```
This program draws a square that changes color every frame and  
follows the mouse*/
```

```
float r;
```

```
float g;
```

```
float b;
```

```
void setup() {
```

```
    size(400,400);
```

```
    background(150);
```

```
}
```

```
void draw() {
```

```
    background(150);
```

```
    r = random(255);
```

```
    g = random(255);
```

```
    b = random(255);
```

```
    fill(r,g,b);
```

```
    rect(mouseX,mouseY,50,50);
```

```
}
```

# Variable Names

- The rules
  - start with a letter, \_ or \$
  - 2nd and on can also be a digit
- The guidelines
  - descriptive
  - start with lower case
  - use camelCaseToSetOffMultipleWords

# Variables – Recap

- declare globally if used to remember values from one call of `draw()` to the next
- declare within `draw()` (or `setup()`) if needed **ONLY** for that one execution of method
- give the type only once, in the declaration (which is different from an assignment)

```
void setup() {  
    size(400,400);  
    yPos = height;  
    xPos = width/2;  
}  
int xPos, yPos;  
void draw() {  
    background(255);  
    ellipse(xPos, yPos, 15, 15);  
    yPos = yPos - 1;  
    xPos = xPos - frameCount/(height/2);  
}
```

- 
- A. The circle moves from bottom center to top center.
- B. The circle moves from bottom center to the middle, then diagonally to the upper left corner.
- C. The circle moves from the bottom center to the middle, then diagonally to the upper right corner.
- D. The circle moves from the bottom center to the middle then stops.