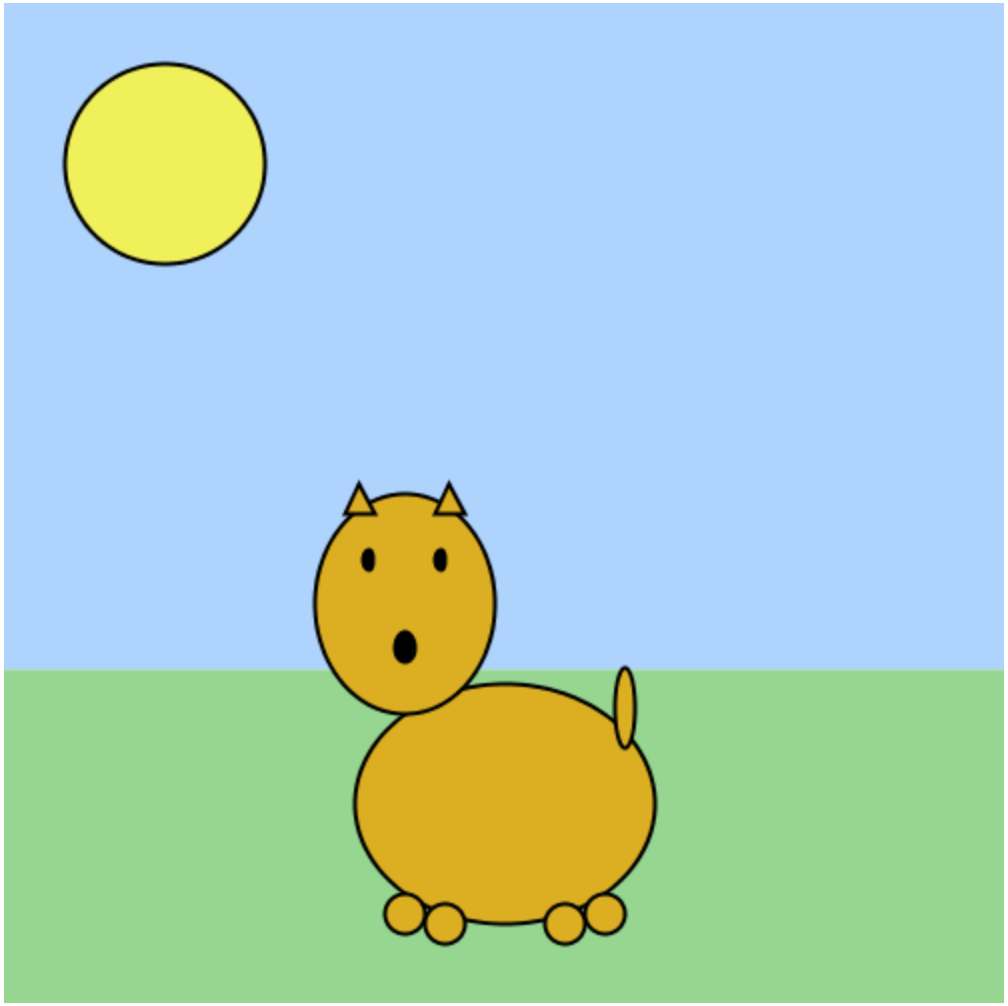
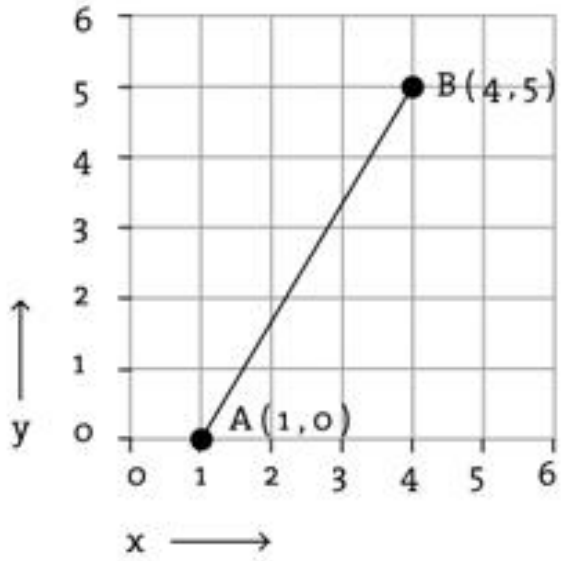
Drawing Pictures with Processing

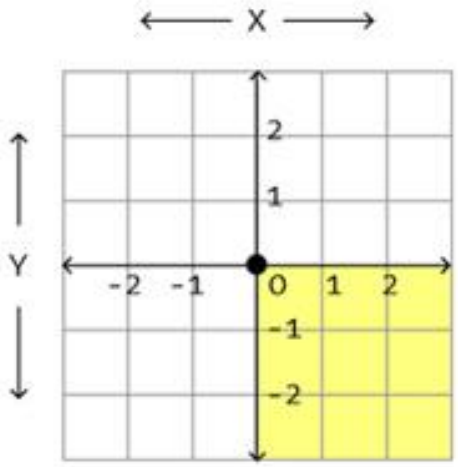
Shapes, Colours, Numeric Data Types, Variables

Admin Stuff

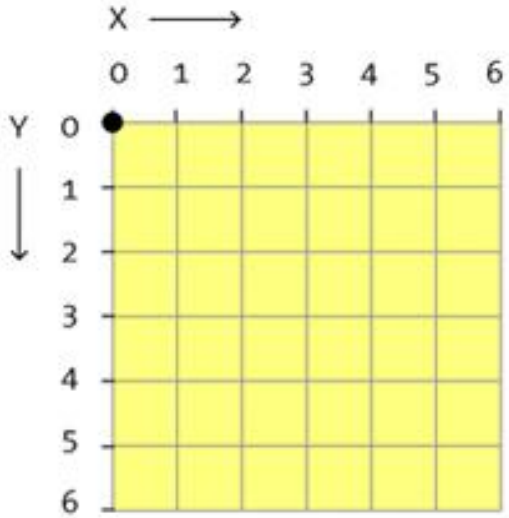
- Tutorials start next week
- TA office hours start next week
- Assignment 1 will be released by Friday, due the following Friday at 11:55pm



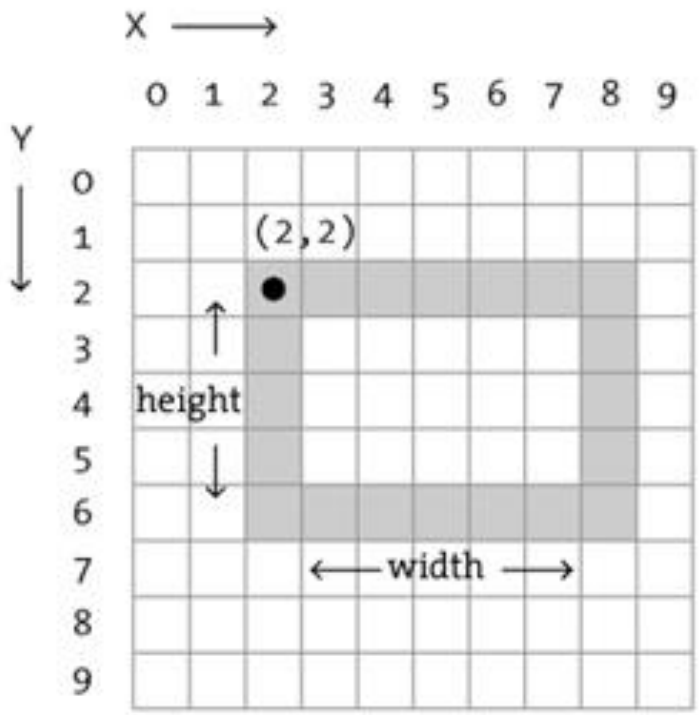

6
5
4
3
2
1
0
y
B(4,5)
A(1,0)
0 1 2 3 4 5 6
x



Eighth Grade



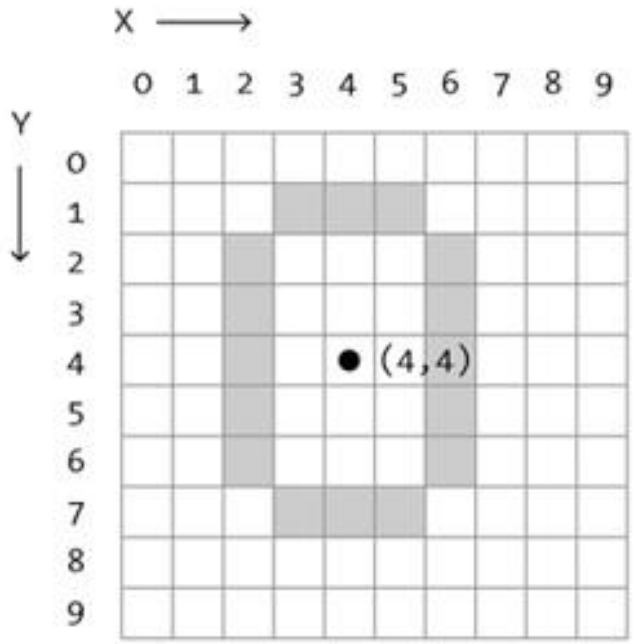
Computer




```
rect(x,y,width,height);
```

Example:

```
rect(2,2,7,5);
```




```
ellipseMode(CENTER);
ellipse(x,y,width,height);
```

Example:

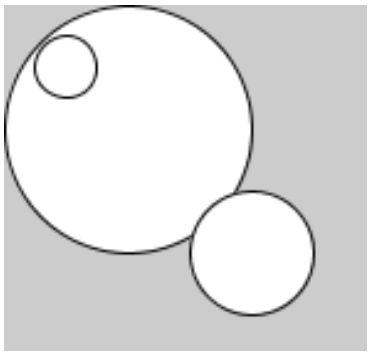
```
ellipseMode(CENTER);
ellipse(4,4,5,7);
```

Poll Everywhere Question:

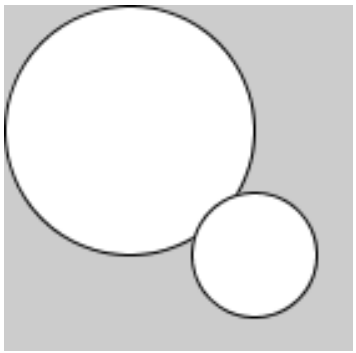
What picture will the following code make?

```
size(130,130);
ellipseMode(CENTER);
ellipse(25, 25, 25, 25);
ellipse(50, 50, 100, 100);
ellipse(100, 100, 50, 50);
```

Text: 37607



623633

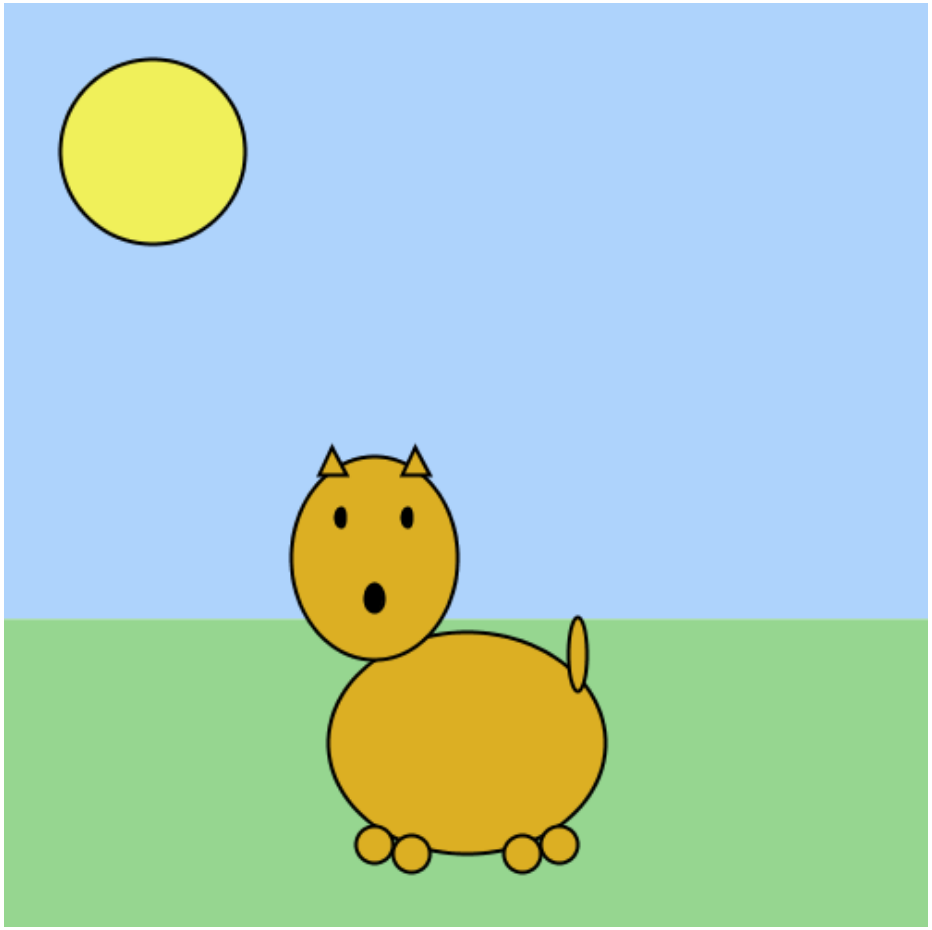


623633

Neither

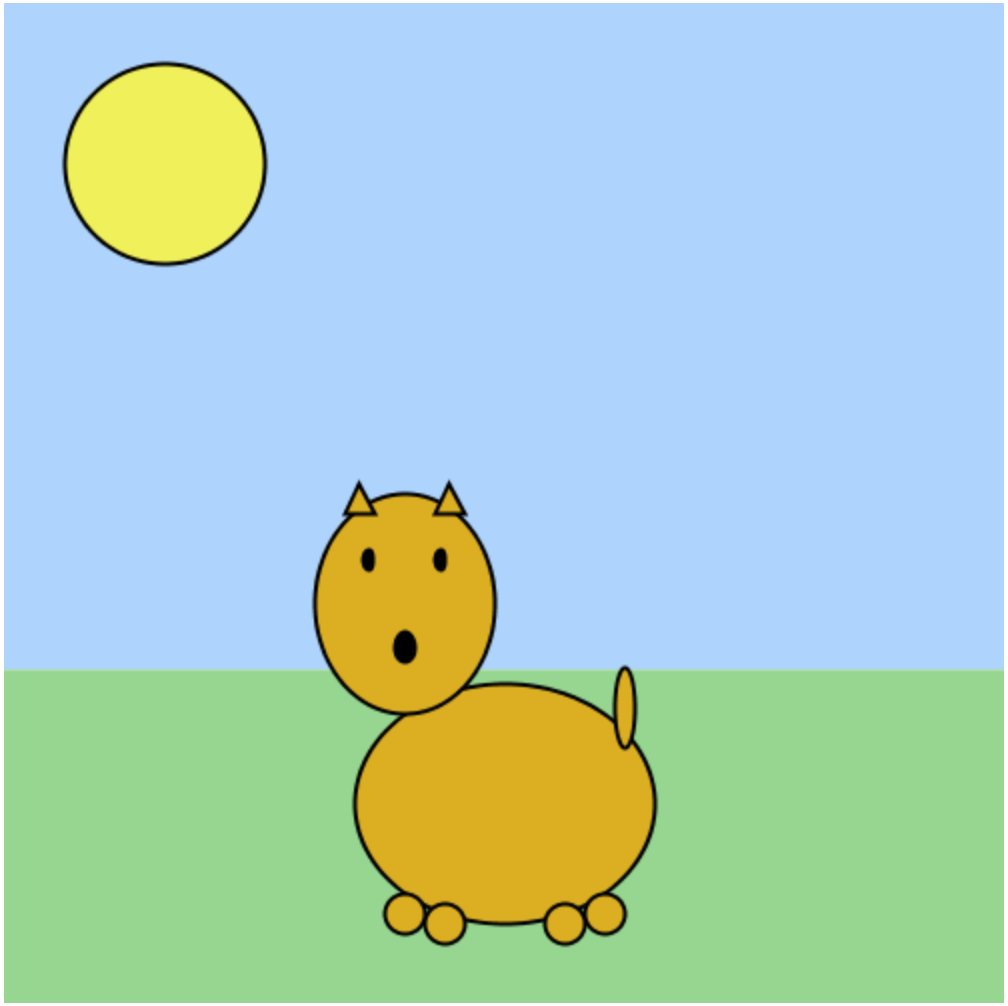
623633

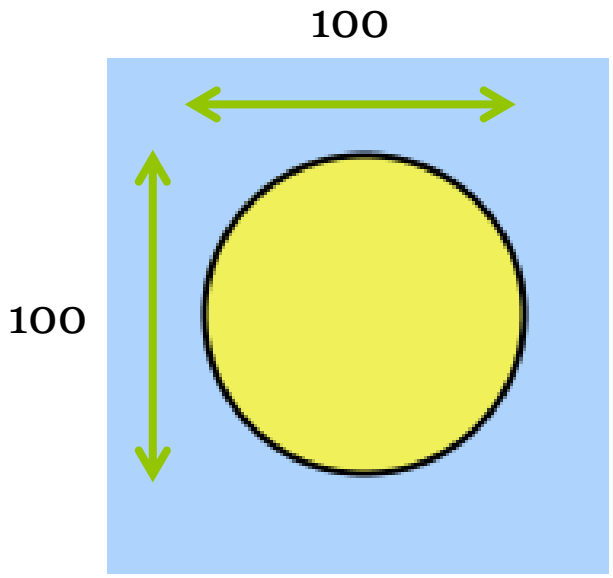
First Step: Make a Plan

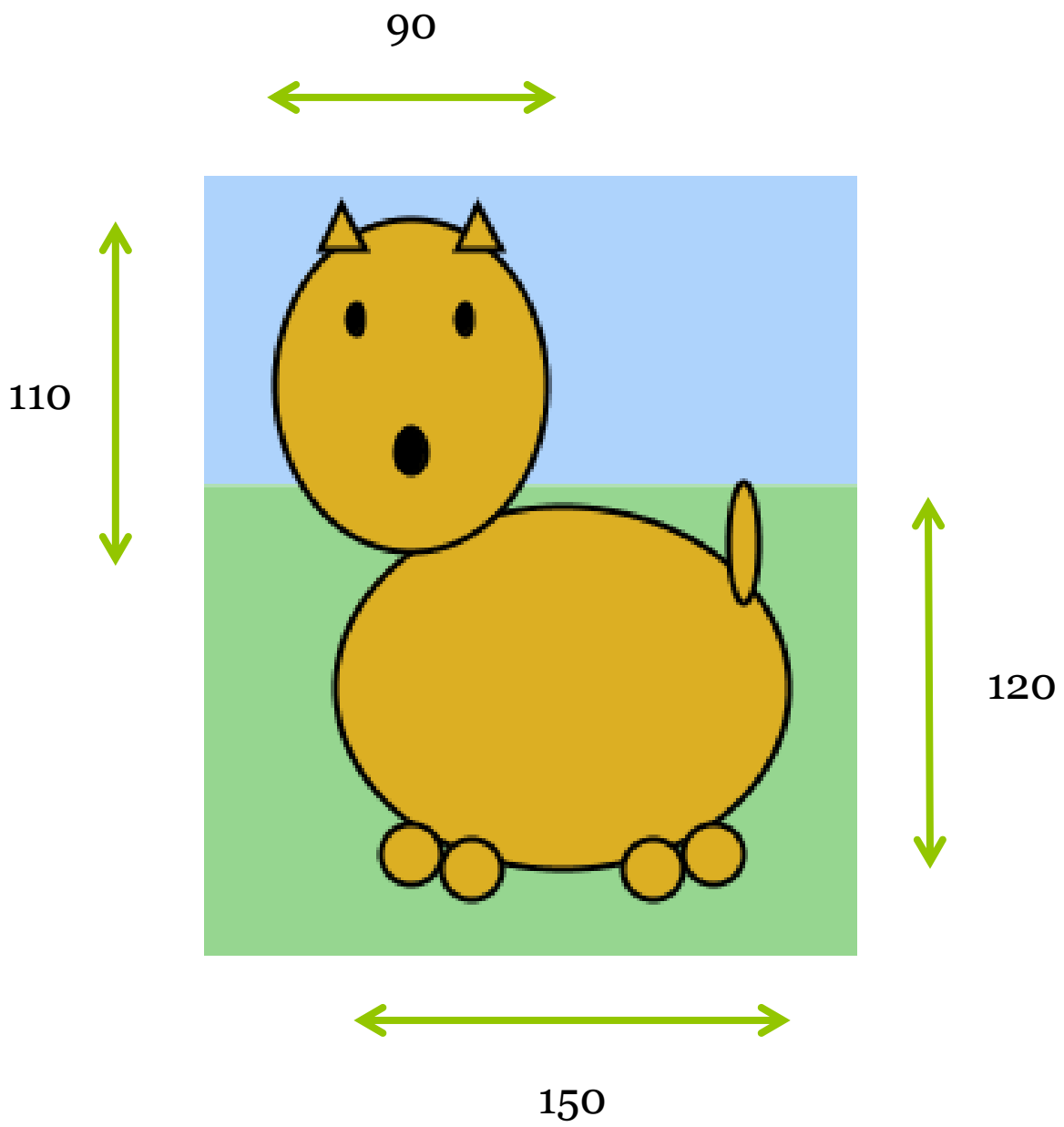


500

500

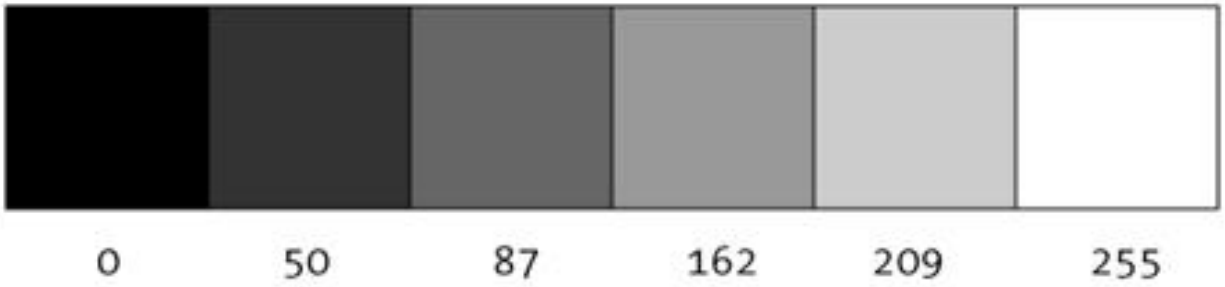



100
100

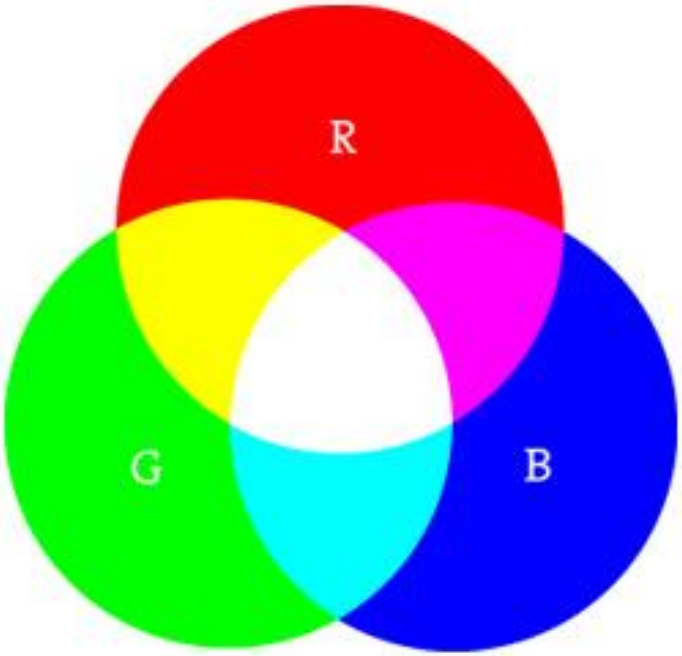

90
110
120
150

Let's get started...

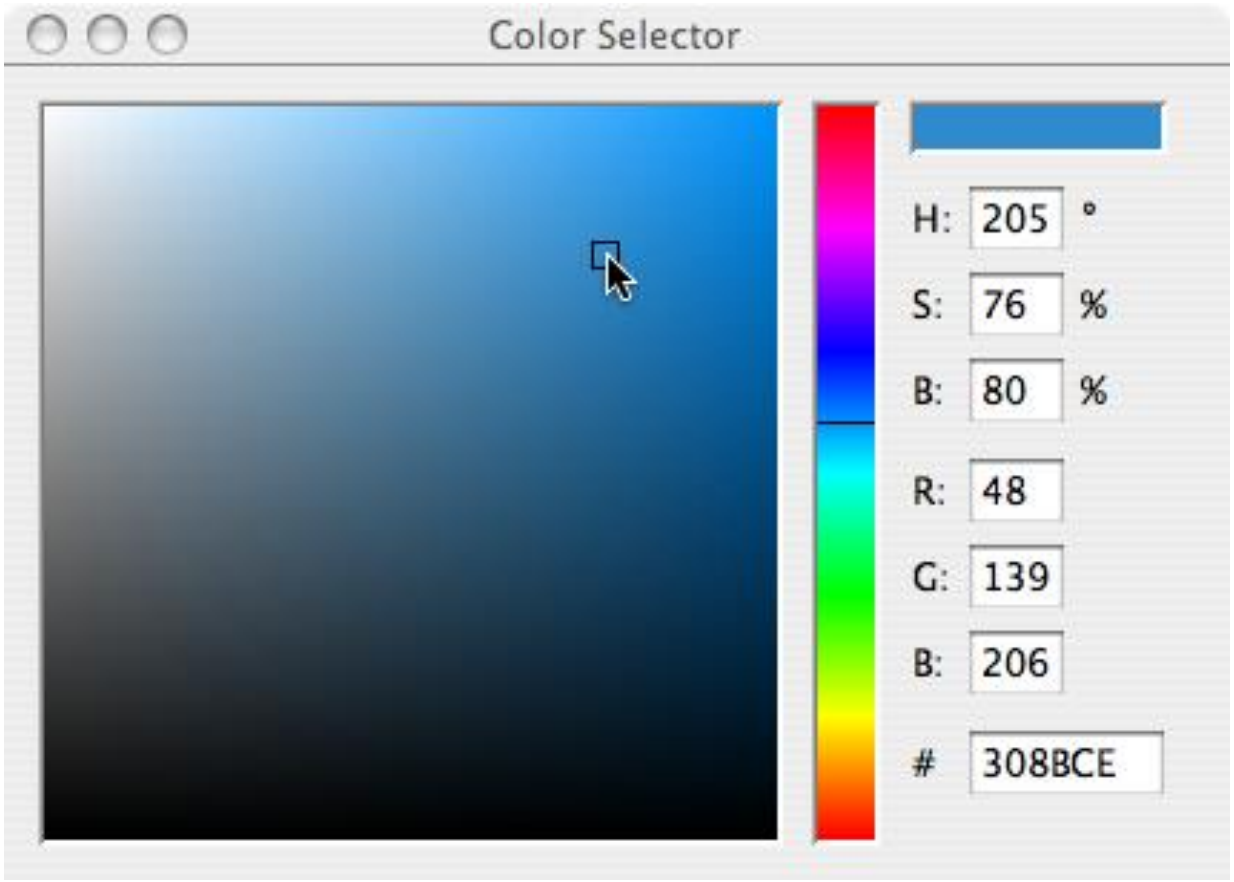
What about color?



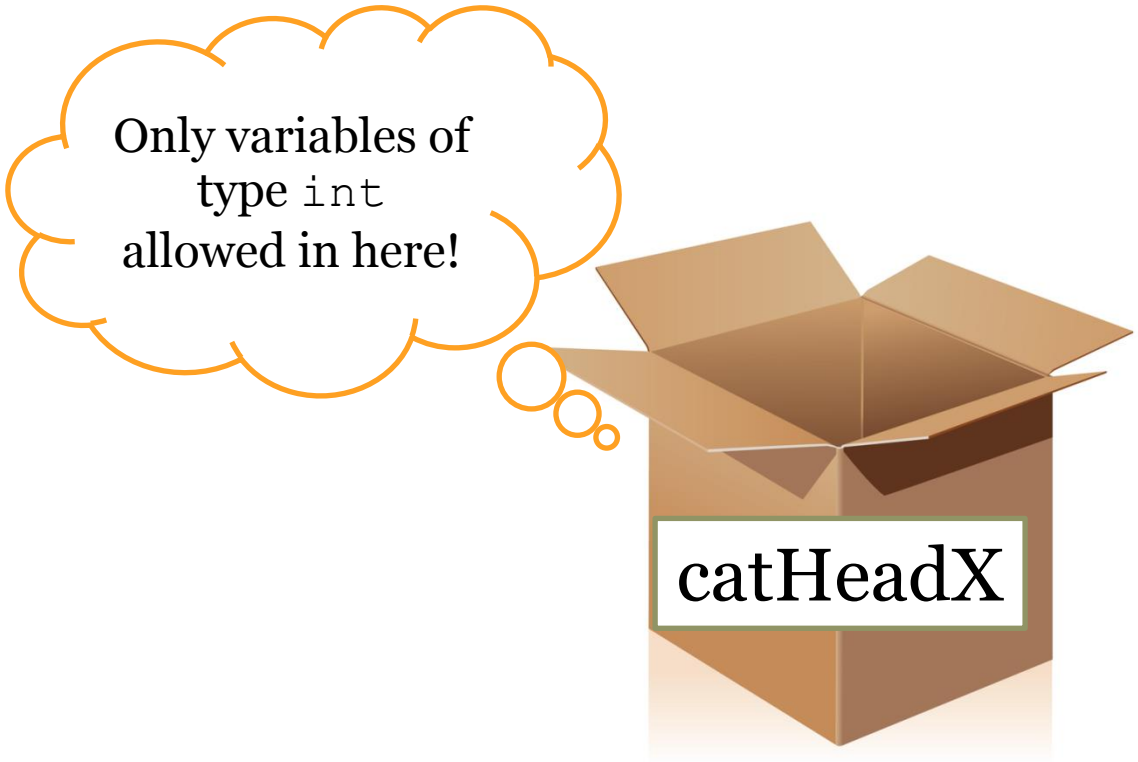
What about color?



What about color?



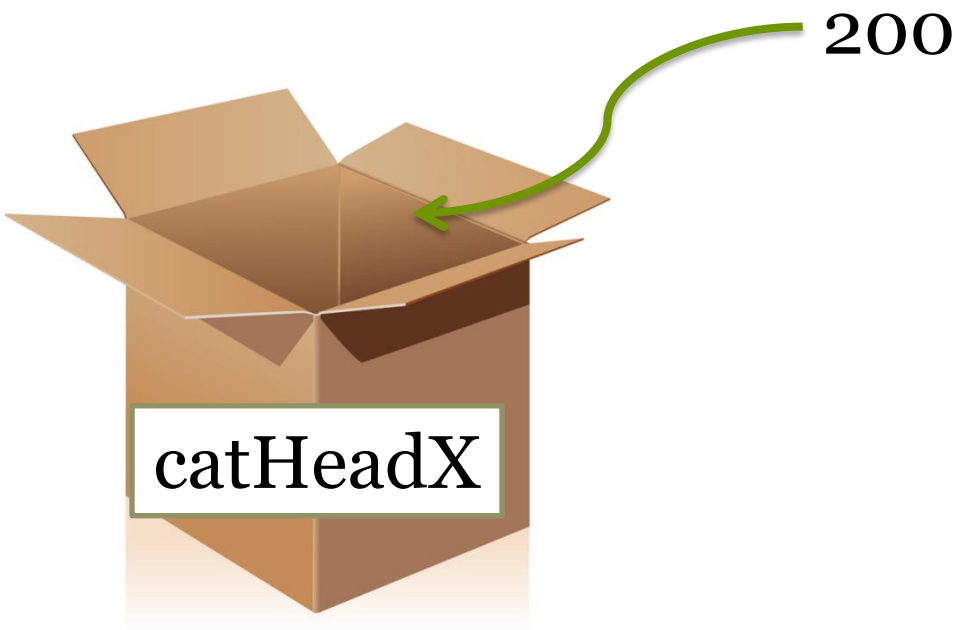
Variables



Variable type → `int catHeadX;`

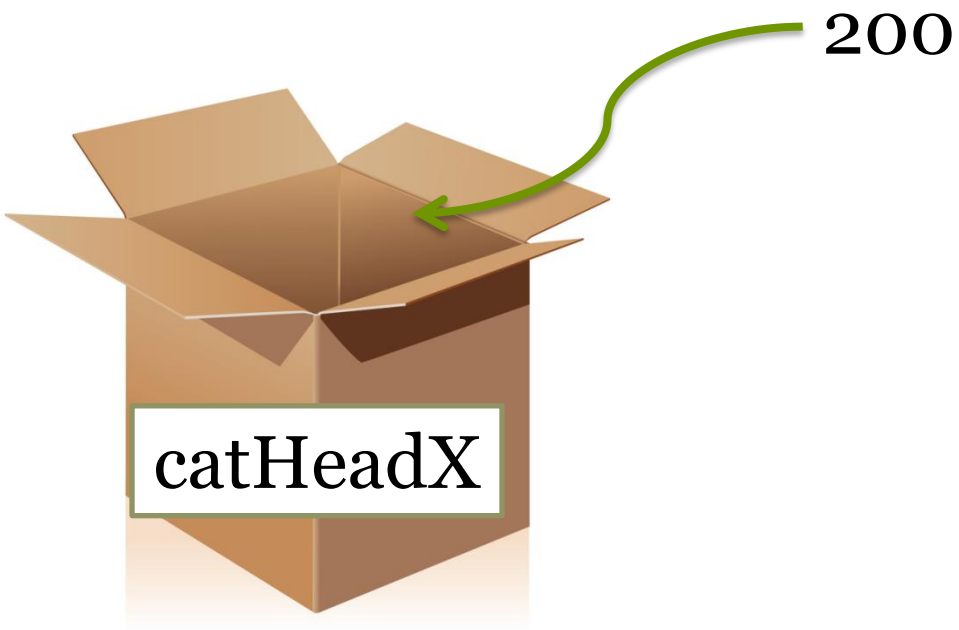
Variable declaration!

Data Type	Values
boolean	true/false
byte	generic 8 bits of data
char	character (‘a’, ‘b’, …)
color	a grayscale or RGB color
double	floating point with double precision
float	floating point (number with a decimal point)
int	integer (whole number)
long	really big integer



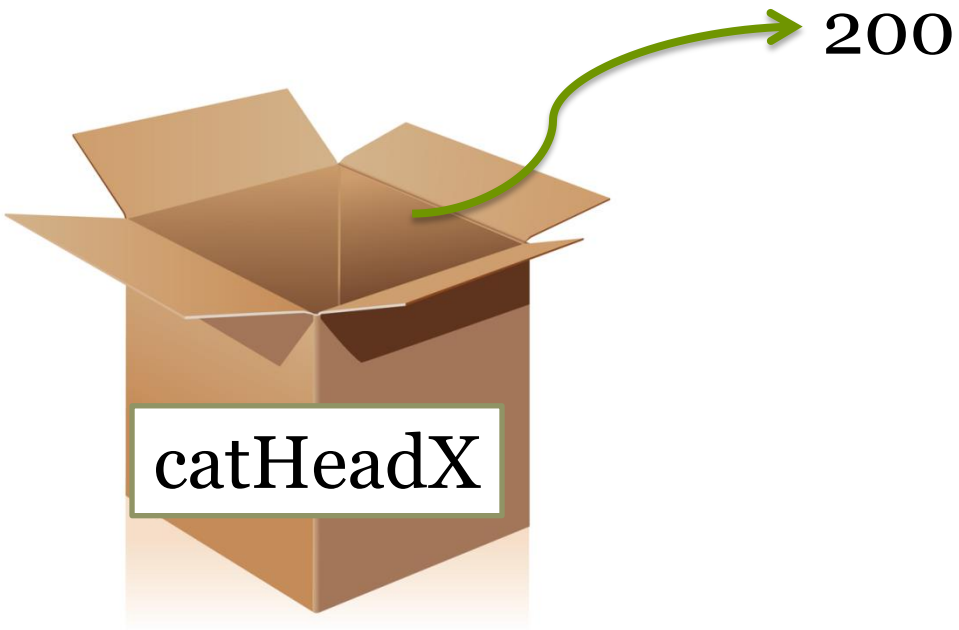

```
catHeadX = 200;
```

Variable assignment!




```
int catHeadX = 200;
```

Variable declaration AND assignment!




```
ellipse(catHeadX, catHeadY, 90, 100);
```

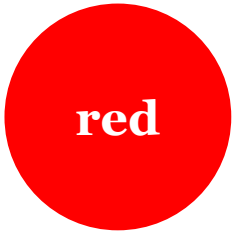
Using the variable's value

Poll Everywhere Question:

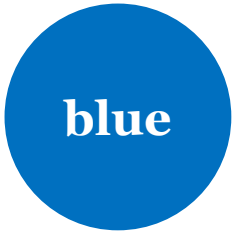
What color will the circle be?

```
color blueColor = color(0,0,255);
color redColor = color(255,0,0);
blueColor = redColor;
redColor = blueColor;
fill(redColor);
ellipse(50,50,75,75);
```

Text: 37607



649726



649727

Neither (syntax error)

649799

Why use variables?

To avoid repetition.
To make code easier to read.

Why Variable Names Matter

The Story of the Magician Marcus and His Frustrating Shopping Trip

http://computationaltales.blogspot.ca/2011/11/names-for-ingredients-and-variables.html

RP3p

Live coding continued…

Identifying Parts of the Code

```
//////
// Draw cat's ears

// These variables are only used here, so keep them
//
int                = 15;
int catEarOffset = 10;

fill(220, 175, 35);

triangle(catHeadX - catHeadWidth/4 + catEarWidth/2, //x1
         catHeadY - catHeadHeight/2 + catEarOffset, // y1
         catHeadX - catHeadWidth/4, //x2
         catHeadY - catHeadHeight/2 - catEarWidth + catEarOffset, // y2
         catHeadX - catHeadWidth/4 - catEarWidth/2, //x3
         catHeadY - catHeadHeight/2 + catEarOffset); //y3
```

comment

Identifying Parts of the Code

```
//////
// Draw cat's ears

// These variables are only used here, so keep them
// close by
int catEarWidth = 15;
int catEarOffset = 10;

                175, 35);

triangle(catHeadX - catHeadWidth/4 + catEarWidth/2, //x1
         catHeadY - catHeadHeight/2 + catEarOffset, // y1
         catHeadX - catHeadWidth/4, //x2
         catHeadY - catHeadHeight/2 - catEarWidth + catEarOffset, // y2
         catHeadX - catHeadWidth/4 - catEarWidth/2, //x3
         catHeadY - catHeadHeight/2 + catEarOffset); //y3
```

data type

Identifying Parts of the Code

```
//////
// Draw cat's ears

// These variables are only used here, so keep them
// close by
int catEarWidth = 15;
int catEarOffset = 10;

fil                ;

tri                - catHeadWidth/4 + catEarWidth/2, //x1
                   - catHeadHeight/2 + catEarOffset, // y1
         catHeadX - catHeadWidth/4, //x2
         catHeadY - catHeadHeight/2 - catEarWidth + catEarOffset, // y2
         catHeadX - catHeadWidth/4 - catEarWidth/2, //x3
         catHeadY - catHeadHeight/2 + catEarOffset); //y3
```

variable name

Identifying Parts of the Code

```
//////
// Draw cat's ears

// These variables are only used here, so keep them
// close by
int catEarWidth = 15;
int catEarOffset = 10;

fill(220, 175

triangle(catH                th/4 + catEarWidth/2, //x1
         catH                ght/2 + catEarOffset, // y1
         catHeadX - catHeadWidth/4, //x2
         catHeadY - catHeadHeight/2 - catEarWidth + catEarOffset, // y2
         catHeadX - catHeadWidth/4 - catEarWidth/2, //x3
         catHeadY - catHeadHeight/2 + catEarOffset); //y3
```

integer value

Identifying Parts of the Code

```
//////
// Draw cat's ears

// These variables are only used here, so keep them
// close by
int catEarWidth = 15;
int catEarOffset = 10;
```

variable declaration and assignment

```
eadWidth/4 + catEarWidth/2, //x1
eadHeight/2 + catEarOffset, // y1
eadWidth/4, //x2
eadHeight/2 - catEarWidth + catEarOffset, // y2
        catHeadX - catHeadWidth/4 - catEarWidth/2, //x3
        catHeadY - catHeadHeight/2 + catEarOffset); //y3
```

Identifying Parts of the Code

```
//////
// Draw cat's ears

// These variables are only used here, so keep them
// close by
int catEarWidth = 15;
int catEarOffset = 10;

fill(220, 175, 35);

triangle(catHeadX - catHeadWidth/4 + catEarWidth/2, //x1
         catHeadY - catHeadHeight/2 + catEarOffset, // y1
         catHead                                    //x2
         catHead                    - catEarWidth + catEarOffset, // y2
         catHead                      catEarWidth/2, //x3
         catHead                    + catEarOffset); //y3
```

using a variable's value

Identifying Parts of the Code

```
//////
// Draw cat's ears

// These variables are only used here, so keep them
// close by
int catEarWidth = 15;
int catEarOffset = 10;

fill(220, 175, 35);

triangle(catHeadX - catHeadWidth/4 + catEarWidth/2, //x1
         catHeadY - catHeadHeight/2 + catEarOffset, // y1
         catHead                                    //x2
         catHead                    - catEarWidth + catEarOffset, // y2
         catHeadX - catHeadWidth/4 - catEarWidth/2, //x3
         catHeadY - catHeadHeight/2 + catEarOffset); //y3
```

expression

Identifying Parts of the Code

```
//////
// Draw cat's ears

// These variables are only used here, so keep them
// close by
int catEarWidth = 15;
int catEarOffset = 10;

fill(220, 175, 35);

triangle(catHeadX - catHeadWidth/4 + catEarWidth/2, //x1
         catHeadY - catHeadHeight/2 + catEarOffset, // y1
         catHead                                   //x2
         catHead                                   - catEarWidth + catEarOffset, // y2
         catHeadX - catHeadWidth/4 - catEarWidth/2, //x3
         catHeadY - catHeadHeight/2 + catEarOffset); //y3
```

expression

Identifying Parts of the Code

```
//////
// Draw cat's ears

// These variables are only used here, so keep them
// close by
int catEarWidth = 15;
int catEarOffset = 10;

fill(220, 175, 35);

triangle(catHeadX - catHeadWidth/4 + catEarWidth/2, //x1
         catHeadY - catHeadHeight/2 + catEarOffset, // y1
         catHeadX -                                 2
         catHeadY -                                 atEarWidth + catEarOffset, // y2
         catHeadX - catHeadWidth/4 - catEarWidth/2, //x3
         catHeadY - catHeadHeight/2 + catEarOffset); //y3
```

expression

Identifying Parts of the Code

```
//////
// Draw cat's ears

// These variables are only used here, so keep them
// close by
int catEarWidth = 15;
int catEarOffset = 10;

fill(220, 175, 35);

triangle(catHeadX - catHeadWidth/4 + catEarWidth/2, //x1
         catHeadY - catHeadHeight/2 + catEarOffset, // y1
         catHeadX - catHeadWidth/4, //x2
         catHeadY - catHeadHeight/2 - catEarWidth + catEarOffset, // y2
         catHeadX - catHeadWidth/4 - catEarWidth/2, //x3
         catHeadY - catHeadHeight/2 + catEarOffset); //y3
```

expression