COMP 1010- Summer 2015 (A01)

Jim (James) Young

young@cs.umanitoba.ca

jimyoung.ca

Hello!

James (Jim) Young
young@cs.umanitoba.ca
jimyoung.ca

office hours T / Th: 17:00 – 18:00
EITC-E2-582
(or by appointment, arrange by email)

some more processing commands

Check these out in the reference

size()

ellipse()

… at home

point()

rect()

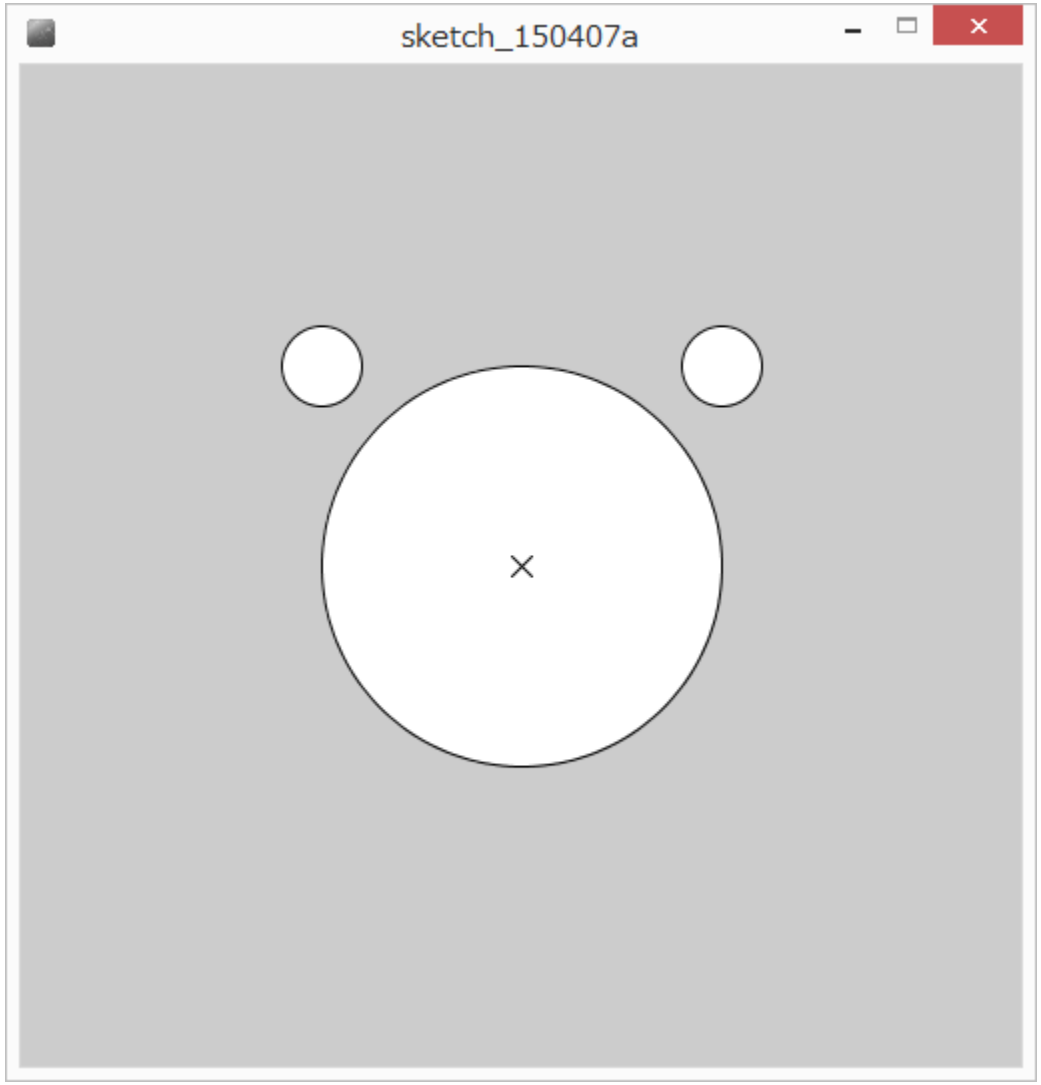
triangle()

What is the coordinate of the bottom right corner?

```
size(500,500);
line(0,0,499,499);
```

Let’s draw an image!

(500x500 canvas)



solution

```
size(500,500);
ellipse(250,250,200,200);
ellipse(150,150,40,40);
ellipse(350,150,40,40);
line(245,245,255,255);
line(255,245,245,255);
```

statements are run sequentially top to bottom

The order of commands to processing changes the result

```
size(500,500);
line(225,225,275,275);
ellipse(250,250,200,200);
```

Where is the line???

comments

Quick! What does this program do?

```
size(500,500);
ellipse(250,250,300,300);
triangle(375,80,300,150,400,200);
triangle(125,80,200,150,100,200);
ellipse(175,225,60,30);
ellipse(325,225,60,30);
ellipse(175,225,15,30);
ellipse(325,225,15,30);
line(250,300,200,275);
line(250,300,300,275);
line(250,300,190,300);
line(250,300,310,300);
line(250,300,200,325);
line(250,300,300,325);
ellipse(250,300,30,30);
```

comments

English language additions to a program to help a programmer understand what is going on.

The computer ignores comments – for humans only.

comments for readability

computer code is not always obvious and you may want to document your reasons:

what does a **statement** do?

if it looks odd, why are you doing it that way?

what big function does some block of **statements** do?

who wrote this code – who should I ~~fire~~ ask for help from?

all reasons for comments

overall making your program easy to read and understandable!

helps you understand! (fewer bugs!)

helps others understand (real-world!)

you lose marks if you don’t!

block comments

start with the characters /*

end with the characters */

EVERYTHING in between is ignored by processing

```
/*  eric the fish */
/*  eric the fruit bat, and,
    eric the cat, and,
    eric the kangaroo */
/**********************
 =^.^=   O_O   -_-  o_O
**********************/
```

Header block – common (and required) block comment

Put a block comment at the beginning of your program explaining what it does…

```
/**********************
* Cat Face! Draw a cat face on the screen
* author: Teo the dog
* version: try #awesome
* purpose: to show how a cat can be drawn
**********************/
size(500,500);
ellipse(250,250,300,300);
triangle(375,80,300,150,400,200);
triangle(125,80,200,150,100,200);
ellipse(175,225,60,30);
ellipse(325,225,60,30);
…
```

in-line comments

For shorter bits

use the characters: //

not \\

everything after // is a comment until the end of the line

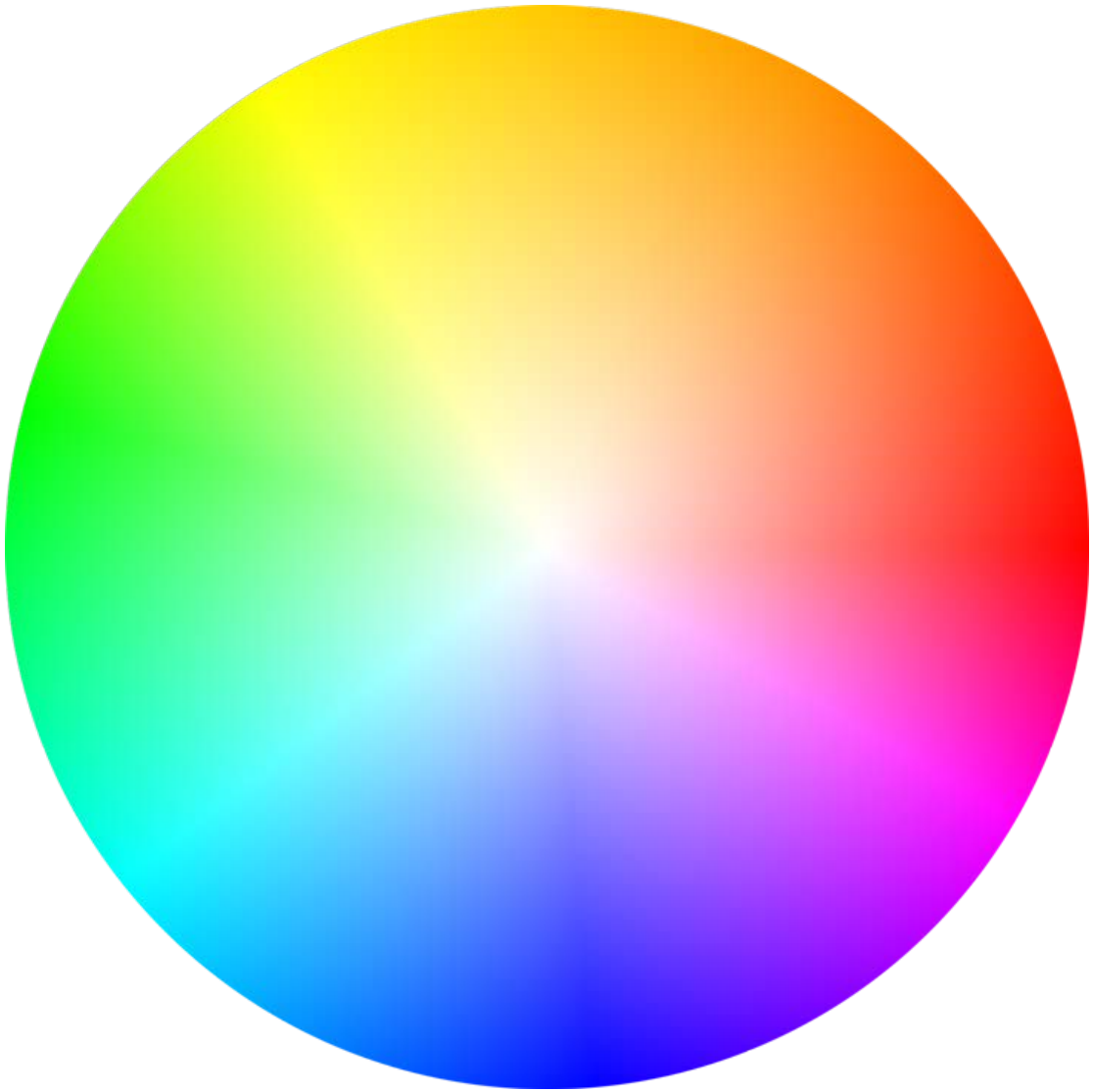
line(0,0,500,500); // a diagonal line

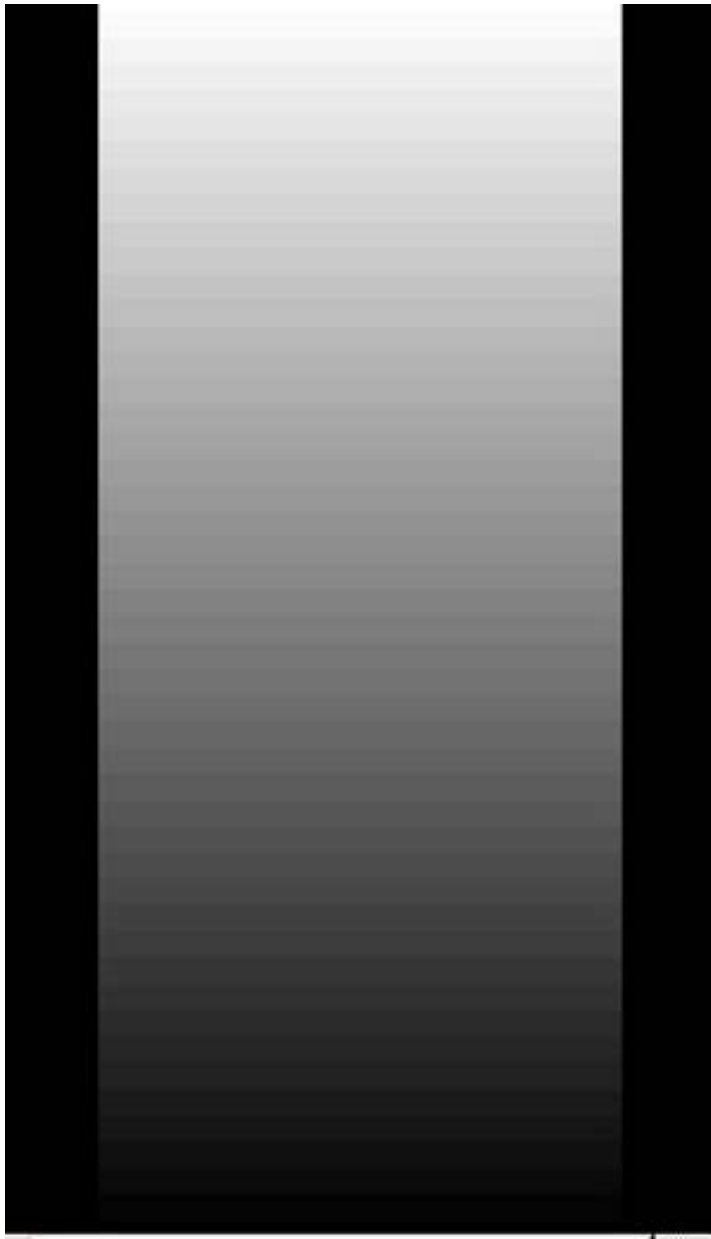
Add comments to the program

```
/*******************
* Cat Face! Draw a cat face on the screen
* author: Teo the dog
* version: try #awesome
* purpose: to show how a cat can be drawn
*********************/

size(500,500); // make a 500x500 canvas

//draw the head
ellipse(250,250,300,300);

//draw the ears
triangle(375,80,300,150,400,200);
triangle(125,80,200,150,100,200);

//draw the eyes
ellipse(175,225,60,30); // left eye
ellipse(175,225,15,30);
ellipse(325,225,60,30); // right eye
ellipse(325,225,15,30);

//whiskers!
line(250,300,200,275);
line(250,300,300,275);
line(250,300,190,300);
line(250,300,310,300);
line(250,300,200,325);
line(250,300,300,325);

// draw the nose. draw after whiskers for nice overlap effect
ellipse(250,300,30,30);
```

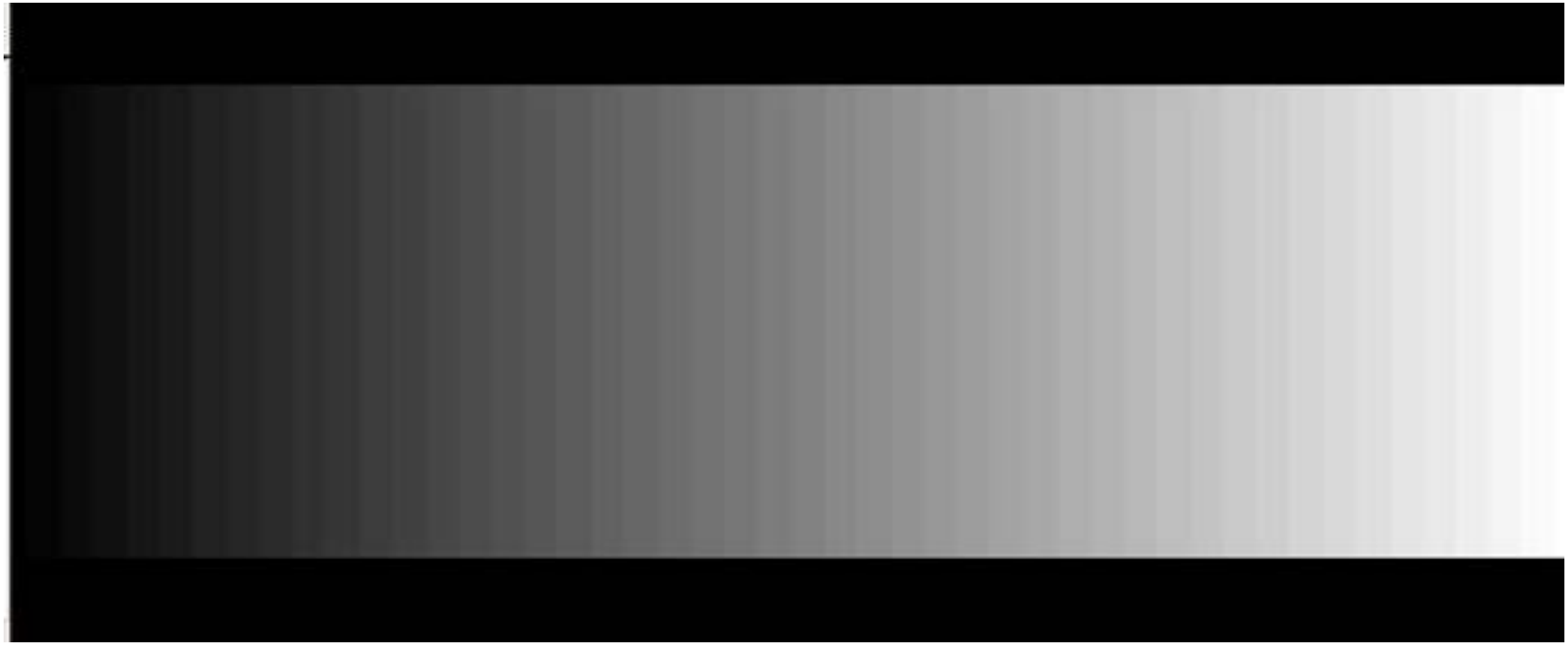
Choosing your paint!

Color!





Grey: 0..255



0……128…….255

Advanced!

Why 0..255?

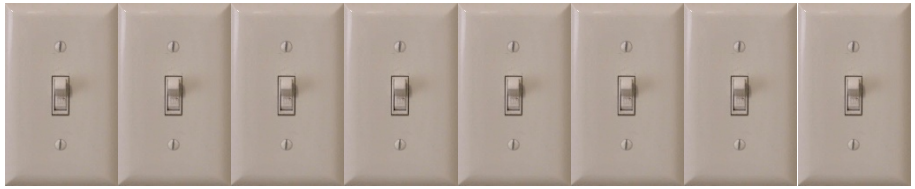
Computer uses 8 bits (switches) to store this color

1 switch – two possible combinations

2 switches – four possible combinations

n switches – 2^n possible combinations

8 switches – $2^8 = 256$ possible combinations



stroke – brush color

stroke(gray level) // sets the brush color

stroke(0) vs stroke(255)

Try with line

stroke – try with ellipse

stroke(0)

ellipse(250,250,50,50)

Why is the ellipse still filled with white?

fill color

Processing has two paint colors: stroke and fill

fill(gray) // 0..255 gray fill level

```
stroke(255);
fill(0);
ellipse(250,250,50,50);
```

how to change the color of the background?

Draw a rectangle over the whole screen!

Annoying…

Use the built in background command

background(grey)

Paints the entire canvas with the color (erases everything else)

Remember the cat?

```
/*******************
* Cat Face! Draw a cat face on the screen
* author: Teo the dog
* version: try #awesome
* purpose: to show how a cat can be drawn
*******************/

size(500,500); // make a 500x500 canvas

//draw the head
ellipse(250,250,300,300);

//draw the ears
triangle(375,80,300,150,400,200);
triangle(125,80,200,150,100,200);

//draw the eyes
ellipse(175,225,60,30); // left eye
ellipse(175,225,15,30);
ellipse(325,225,60,30); // right eye
ellipse(325,225,15,30);

//whiskers!
line(250,300,200,275);
line(250,300,300,275);
line(250,300,190,300);
line(250,300,310,300);
line(250,300,200,325);
line(250,300,300,325);

// draw the nose. draw after whiskers for nice overlap effect
ellipse(250,300,30,30);
```

What if we want to make the nose and whiskers little higher? Say, 100 pixels higher?

Let’s do it

That was a pain!!

There must be a better way..

Variables!

variables

variables

sometimes we need to store data to use it in various processing:

e.g., log into a website: it should remember who you are!

in this case, it would store your name in a **variable**!

e.g.,: Hi [name], welcome to our site! We see that you come from [city]. People in [city] have previously bought the following items from us

note: this company writes the above line ONCE. **[name]** and **[city]** are variables. When you login, your name and city are stored in those variables.

math example:

x = 1 ← **store** the #1 in the **variable** x

y = 15 ← **store** the #15 in the **variable** y

z = (x+1)*y ← **read** the values from **variables** x and y, do a calculation, and store the result in **z**

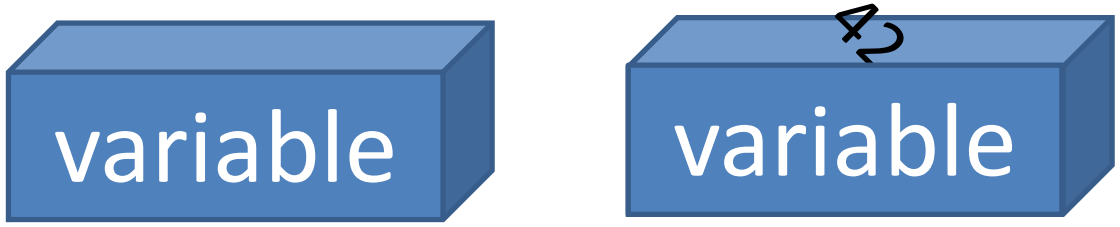
what does z =?

x, y, and z are variables because they can store a range of numbers depending on the circumstances

box analogy

variables can be seen as boxes that you put data into

42



sorting a mess into designated boxes….

books

papers

toys

processing is just as strict

When you create a variable (a box), you need to tell processing what kind of information you are going to store in it

A number

Some text

Some music!!

So how do you specify this in processing?

Your first data type: the integer

What is an integer? (highschool math?)

a whole number with no fractional part

42, -5002, 0, 24

what is the biggest integer?

theoretically there is none

Processing limits the amount of computer memory used by your integer so there is a maximum and a minimum

maximum: **2,147,483,647!**

(((huh!? $2^{32}/2-1$)))

minimum: **-2,147,483,648!**

there are **data types** available with more memory but we will talk about them later

back up a second.. No fractions??? why such a ridiculous limitation?

its easier and faster for computers to store and work with whole numbers, so we usually stick to them if its all that we need

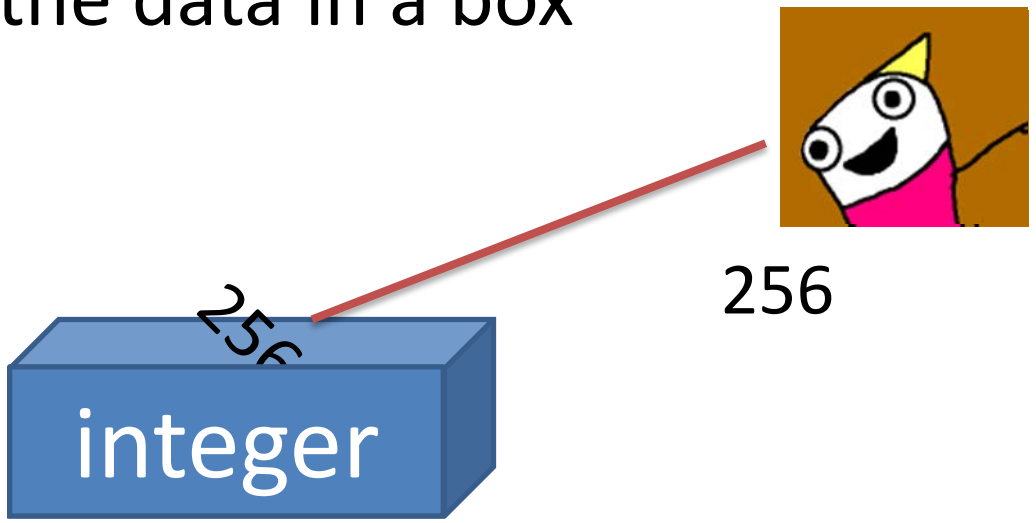
we‘ll learn a new data type for decimals later

three operations on **variables** (boxes)

a) make a new box to store some data in

b) put data into a box

c) look at the data in a box



variables in processing

a) make a new **variable** to store some data in

b) put data into a **variable**

c) look at the data in a **variable**

variableType variableName;

```
int noseCenterX;
int noseCenterY;
```

variables in processing

a) make a new **variable** to store some data in

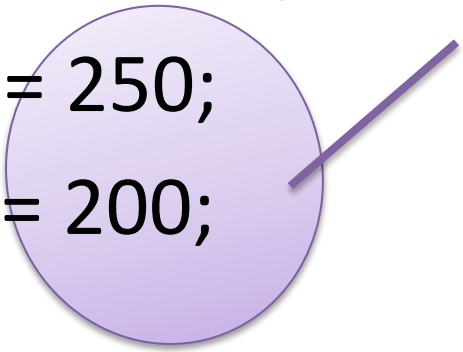
b) put data into a **variable**

c) look at the data in a **variable**

variableName = data;

noseCenterX = 250;

noseCenterY = 200;



literals

variables in processing

a) make a new **variable** to store some data in

b) put data into a **variable**

c) look at the data in a **variable**

use the **variableName** anywhere you would use raw data

```
ellipse(noseCenterX,noseCenterY,30,30);
noseCenterX = noseCenterY;
```

Let’s update our program

combined declaration and assignment

You can create a variable and assign it a value in one statement!

Instead of….

```
int noseCenterX;
noseCenterX = 250;
```

variableType variableName = data;

```
int noseCenterX = 250;
```

variables are.. well.. variable..

```
int circleSize = 50;
ellipse(100,100,circleSize,circleSize);
circleSize = 10;
ellipse(100,100,circleSize,circleSize);
```

Same command but different result!

Re-defining variables

```
int circleSize = 50;
ellipse(100,100,circleSize,circleSize);
int circleSize = 10;
ellipse(100,100,circleSize,circleSize
```

Variables can only be defined once, otherwise it’s like you’re trying to create two different ones with the same name. Like building a new house on top of an existing one!!

empty variables

```
int circleSize;
ellipse(100,100,circleSize,circleSize);
```

Well – what would you expect to reasonably happen?

Variables must be *initialized* – given a value – before they are used.

rules for variable names

no spaces!

no special characters !”#%&’()-=^[]{}

exception: underscore _

cannot start with a number but can contain one

no reserved words

reserved words…

are special words that already have a special meaning, so you cannot use them as your program or variable names.

e.g.,

int int

Reserved Words:

Strategy: be aware of the problem, but don’t memorize

Integer Operations

integer operations!

you can do a bunch of standard math operations on integers:

- addition
- subtraction
- multiplication
- division
- …?

integer “operators” (operations?)

addition!

the “+” symbol:

<integer> + <integer>

5+4, 6+10, 657+552, 5554543+2223232

Let’s look at one of the cat whiskers.

Remember the cat?

```
/*******************
* Cat Face! Draw a cat face on the screen
* author: Teo the dog
* version: try #awesome
* purpose: to show how a cat can be drawn
*******************/

size(500,500); // make a 500x500 canvas

//draw the head
ellipse(250,250,300,300);

//draw the ears
triangle(375,80,300,150,400,200);
triangle(125,80,200,150,100,200);

//draw the eyes
ellipse(175,225,60,30); // left eye
ellipse(175,225,15,30);
ellipse(325,225,60,30); // right eye
ellipse(325,225,15,30);

//whiskers!
line(250,300,200,275);
line(250,300,300,275);
line(250,300,190,300);
line(250,300,310,300);
line(250,300,200,325);
line(250,300,300,325);

// draw the nose. draw after whiskers for nice overlap effect
ellipse(250,300,30,30);
```

Cat whisker:

line(250,300,300,325);

→line(noseCenterX,noseCenterY,300,325);

Notice: the line end point is..

50 pixels to the right (250→300)

25 pixels below (300→325)

line(noseCenterX, noseCenterY,

noseCenterX+50, noseCenterY+25);