

# Variables & Datatypes

CSE 120 Winter 2018

## Instructor:

Justin Hsia

## Teaching Assistants:

Anupam Gupta, Cheng Ni,  
Sam Wolfson, Sophie Tian,

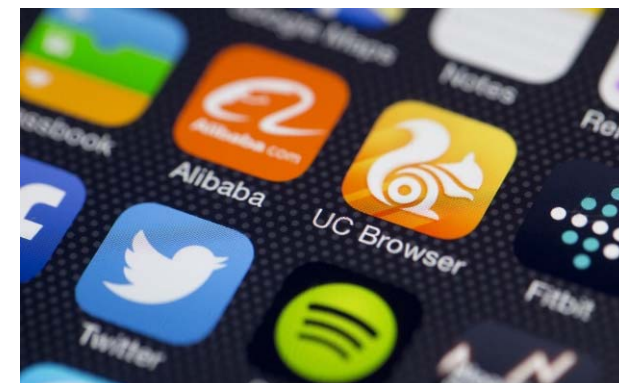
Eugene Oh,  
Teagan Horkan

## A Browser You've Never Heard of Is Dethroning Google in Asia

"A mobile browser rarely used in the West has outflanked Google's Chrome in some of Asia's fastest-growing markets, giving owner Alibaba Group Holding Ltd. an advantage in the race among technology giants to capture the next generation of internet users.

"Hundreds of millions of people in India, Indonesia and other emerging markets getting online for the first time are picking UC Browser, owned by Chinese e-commerce giant Alibaba, over ones made by U.S. rivals. Users say UC Browser works better in countries dominated by low-end smartphones and spotty mobile service.

- <https://www.wsj.com/amp/articles/a-browser-youve-never-heard-of-is-dethroning-google-in-asia-1514808002>

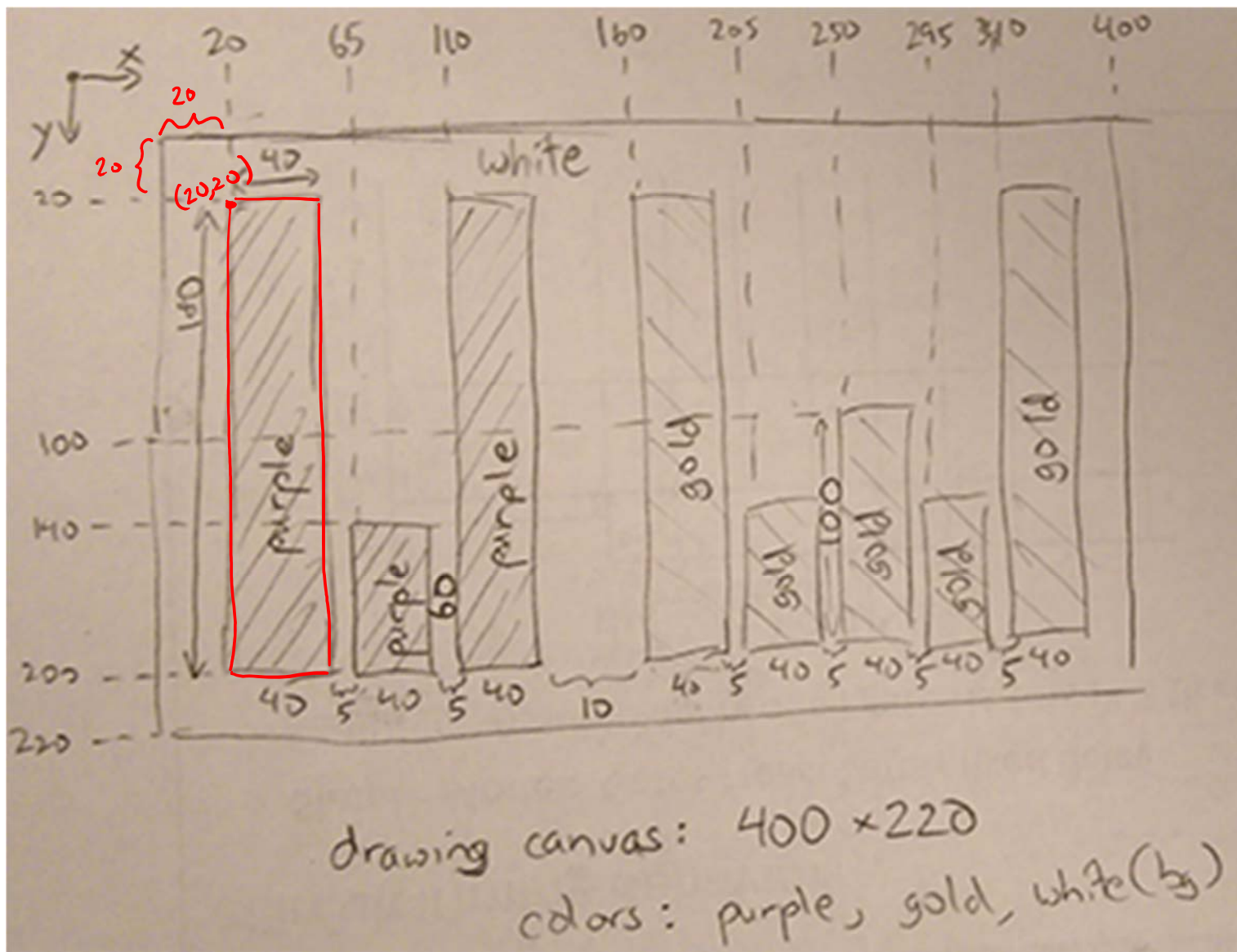


# Administrivia

- ❖ Assignments:
  - Taijitu due before lab tomorrow (1/11)
  - Reading Check 1 due tomorrow (1/11)
  - Logo Design due Friday (1/12)
  
- ❖ Signing up for Piazza [demo]

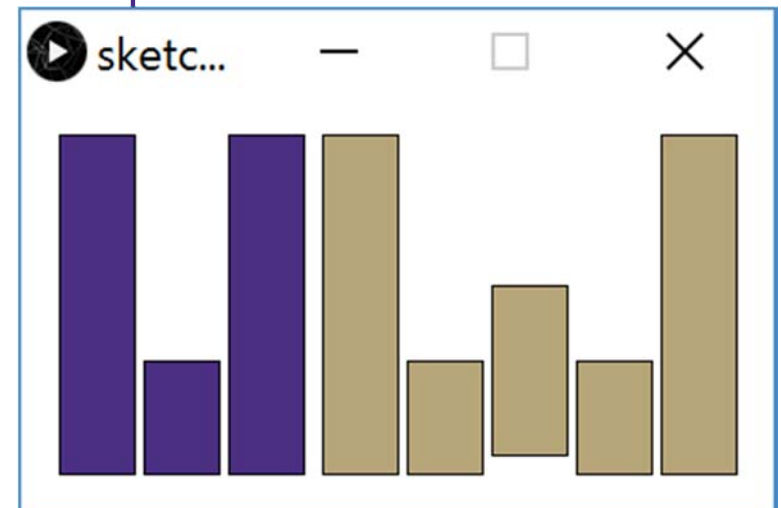
# Homework: Logo Design

Coordinate system reminder



# Homework: Logo Design

```
uw_logo ▾
1 /* uw_logo.pde
2   Created by Justin Hsia
3
4   UW logo made out of rectangles in school colors.
5 */
6
7 size(400,220); // drawing canvas of 400x220
8 background(255); // white background
9
10 // The letter 'U' in purple
11 fill(75, 47, 131); // purple fill
12 rect(20, 20, 40, 180); // left side of U
13 rect(65, 140, 40, 60); // middle base of U
14 rect(110, 20, 40, 180); // right side of U
15
16 // The letter 'W' in gold
17 fill(183, 165, 122); // gold fill
18 rect(160, 20, 40, 180); // left segment of W
19 rect(205, 140, 40, 60); // left base of W
20 rect(250, 100, 40, 90); // middle segment of W
21 rect(295, 140, 40, 60); // right base of W
22 rect(340, 20, 40, 180); // right segment of W
```

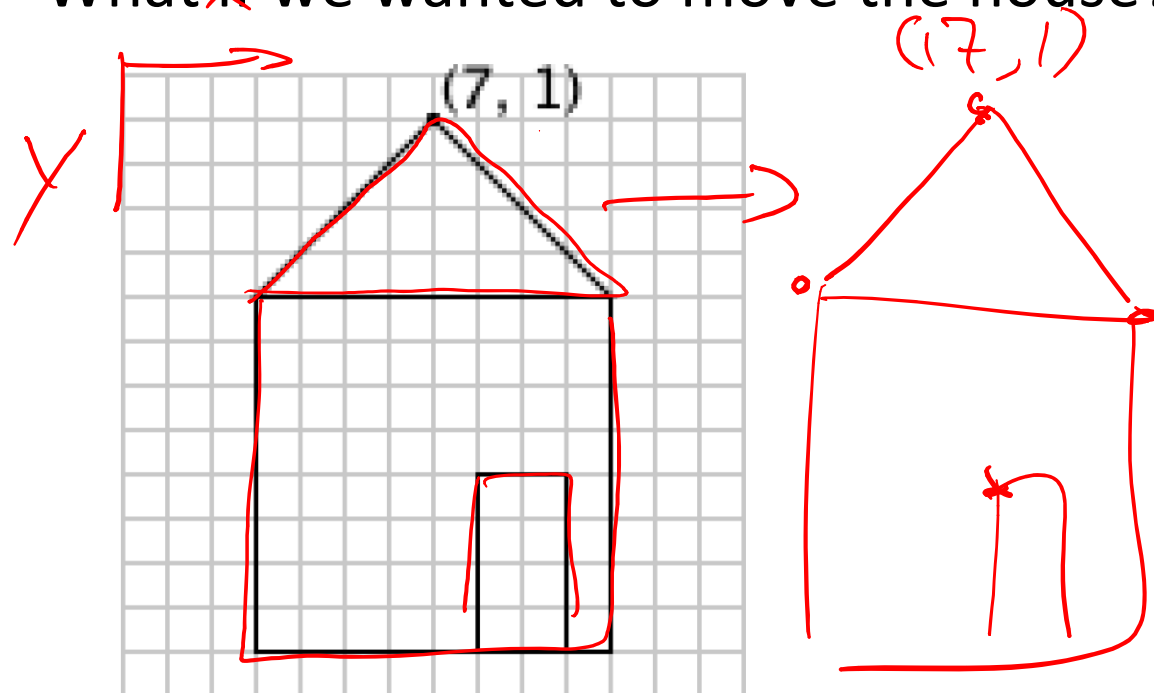


# Drawing a House

- ❖ One solution from worksheet:

```
triangle(717, 1, 313, 5, 1121, 5);  
rect(313, 5, 8, 8);  
rect(818, 9, 2, 4);
```

- What ~~x~~ if we wanted to move the house?



# Variables

- ❖ Piece of your program that holds the value of something *storage space, like a box*
  - Every variable must be given a name and a data type
    - identifier (which box?)* →
    - "shape" of box* →
- ❖ The values of these **variables** can change (*i.e.* vary) during the execution of your program
  - Warning: Not like a variable in Algebra (*i.e.* an unknown)
    - ↳  $5x=20$  implies  $x=4$*
- ❖ **Assignment/Write**: give a variable a specific value
  - *e.g.*  $x \leftarrow 12;$  *"put value 12 into the box x"*

# Variables

- ❖ Piece of your program that holds the value of something *storage space, like a box*
  - Every variable must be given a name and a data type
    - identifier (which box?)* (points to name)
    - "shape" of box* (points to data type)
- ❖ The values of these **variables** can change (*i.e.* vary) during the execution of your program
  - Warning: Not like a variable in Algebra (*i.e.* an unknown)
- ❖ **Read**: use the current value of a variable
  - e.g. `ellipse(x+1, 50, 20, 20);`
    - 12+1* (above x)
    - look in box x and use its current value* (points to x)
    - draws an ellipse with center at (13,50)* (points to the entire expression)

# Datatypes

- ❖ `int` integers (e.g. 12)
- ❖ `float` decimal/real numbers (e.g. 3.14)
- ❖ `color` RGB (e.g. `color(0)`)
- ❖ `char` characters (e.g. 'J')
- ❖ `boolean` true or false (e.g. `true`)

- ❖ Many more exist and can be found in the Processing Reference:

Primitive

`boolean`

`byte`

`char`

`color`

`double`

`float`

`int`

`long`



# Declarations

- ❖ We **declare** a variable by telling Processing the variable's datatype, followed by the variable's name:

```
1 int x;  
2 float half;  
3 color yellow;
```

*x = 4;*

*Separate declaration  
and initialization*

- ❖ You can also give a variable a starting value (**initialization**) in the same line as the declaration:

```
1 int x <=> 4; // combined declaration and initialization  
2 float half <=> 0.5;  
3 color yellow <=> color(255, 255, 0);
```

# Variable Manipulation

- ❖ Executed sequentially, just like other statements
- ❖ For variable assignments, compute right-hand side *first*, then store result in variable

❖ Example:

```
int x = 4;  
x = x + 1;
```

- 1) Read the current value of  $x$  (4) for the right-hand side
- 2) Add 1 to the current value of  $x$
- 3) Store the result (5) back into  $x$

# Variable Practice

```
1) int x = 1;
   int y = 2;
   int z = 3;
```

$$\begin{aligned} 2 &= 1 + 1; \\ 1 &= 2 - 1; \\ 5 &= 3 + 2; \end{aligned}$$

x	y	z
2	1	5

```
2) int x = 7;
   int y = 2;
   int z = 0;
```

$$\begin{aligned} 10 &= 7 + 3; \\ 0 &= 2 - 2; \\ &= 10 + 0; \end{aligned}$$

x	y	z
10	0	10

```
3) int x = -1;
   int y = 0;
   int z = 5;
```

$$\begin{aligned} 4 &= -1 + 5; \\ 4 &= 0 - (-1); \\ 9 &= 4 + 5; \end{aligned}$$

x	y	z
4	-4	9

# Variable Rules & Guidelines

- ❖ Variable naming rules:
  - Variables are case-sensitive (e.g. `myx` vs. `myX`)
  - Numbers allowed, but not at beginning (e.g. `k9` vs. `9k`)
  - Generally avoid symbols other than underscore (e.g. `my_x`)
- ❖ Variable names are meaningless to computers, but meaningful to humans
  - Choosing informative names improves readability and reduces confusion ★
- ❖ In this class, most of our variables will be declared and initialized at the very top of our programs

# Drawing a House with Variables

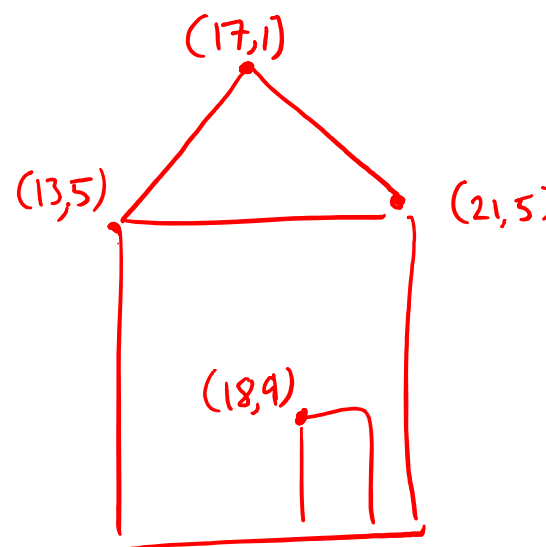
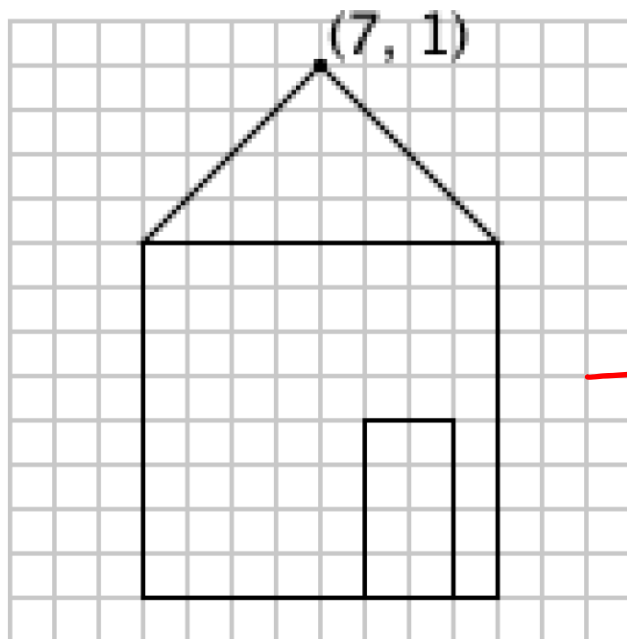
*int houseX = 7;*

❖ Initial solution:

```
triangle(7x, 1, 3x-4, 5, 11x+4, 5);
rect(3x-4, 5, 8, 8);
rect(8x+1, 9, 2, 4);
```

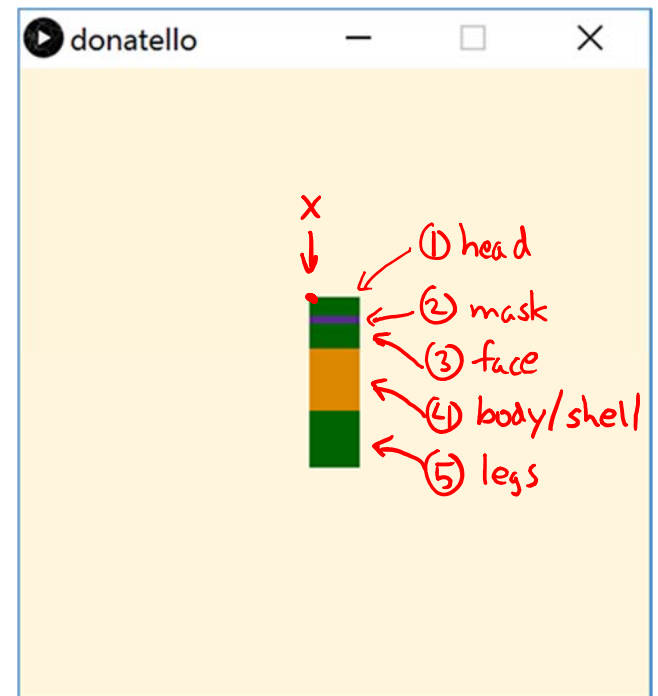
- What properties might be useful to store in variables?

*x  
y  
size/scale*



# TMNT: Donatello

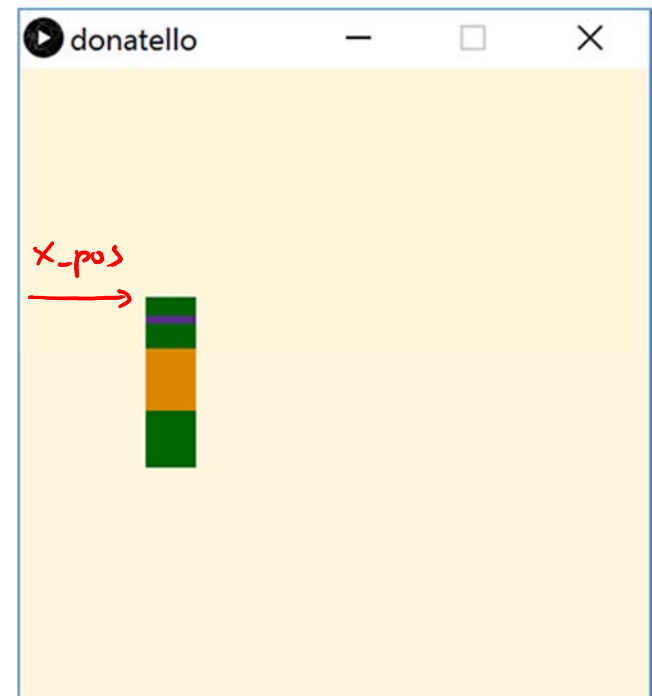
```
donatello
1 size(500,500);
2 noStroke();
3 background(255,245,220);
4
5 // Donatello
6 fill(0,100,0); // dark green
7 rect(230,182,40,15); // top of head
8
9 fill(88,44,141); // purple
10 rect(230,197,40,6); // bandana mask
11
12 fill(0,100,0); // dark green
13 rect(230,203,40,20); // bottom of head
14
15 fill(219,136,0); // dark yellow
16 rect(230,223,40,50); // shell
17
18 fill(0,100,0); // dark green
19 rect(230,273,40,45); // lower body
```



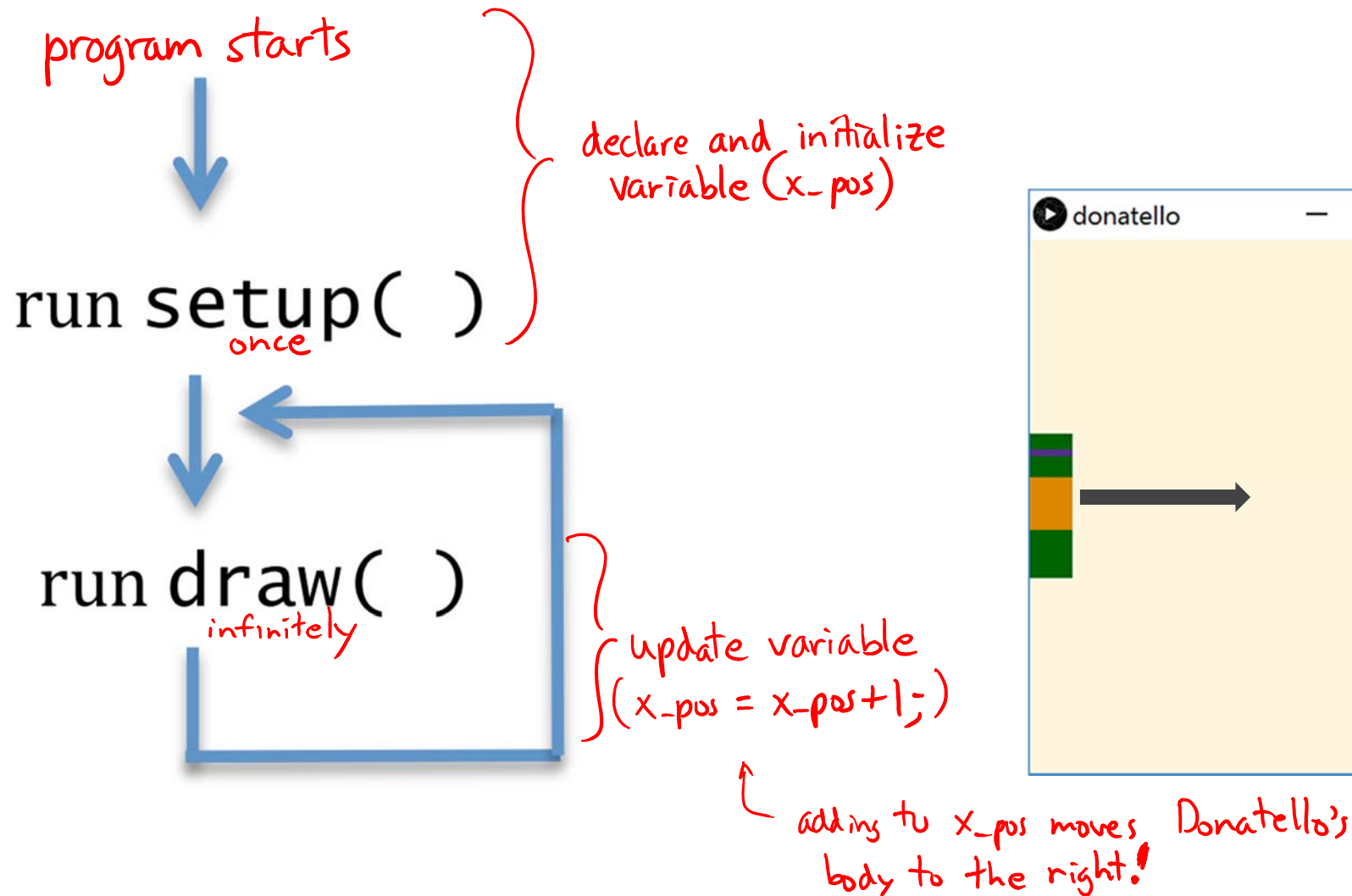
# Donatello with a Variable

```
1 int x_pos = 100;           // x-position
2
3 size(500,500);
4 noStroke();
5 background(255,245,220);
6
7 // Donatello
8 fill(0,100,0);           // dark green
9 rect(x_pos,182,40,15);   // top of head
10
11 fill(88,44,141);        // purple
12 rect(x_pos,197,40,6);   // bandana mask
13
14 fill(0,100,0);           // dark green
15 rect(x_pos,203,40,20);  // bottom of head
16
17 fill(219,136,0);        // dark yellow
18 rect(x_pos,223,40,50);  // shell
19
20 fill(0,100,0);           // dark green
21 rect(x_pos,273,40,45);  // lower body
```

*x\_pos moves entire  
drawing!*



# Donatello with Motion





# Stopping Motion

- ❖ Stop Donatello from running off the right side of the screen:

```
x_pos = min(x_pos + 1, 460);
```

*returns minimum of these two numbers*

old x_pos	x_pos+1	min(x_pos+1, 460)	new x_pos
0	1	1	1
1	2	2	2
2	3	3	3
⋮	⋮	⋮	⋮
458	459	459	459
459	460	460	460
460	461	460	460

*← sets maximum x\_pos of 460*

- ❖ Stop Donatello from running off the left side of the screen:

```
x_pos = max(x_pos - 1, 0);
```

*returns maximum of these two numbers*  
*// sets minimum x\_pos of 0*

# Falling Into Place

- ❖ Introduce variables for each body segment:

```
3 int head_pos = 0; // head position
4 float mask_pos = 15; // mask position
5 int face_pos = 21; // face position
6 float body_pos = 41; // body position
7 int leg_pos = 91; // leg position
```

*initial y-positions for each body segment*

- ❖ Update each variable at different speeds:

```
33 head_pos = min(head_pos + 3, 364);
34 mask_pos = min(mask_pos + 3.5, 379);
35 face_pos = min(face_pos + 4, 385);
36 body_pos = min(body_pos + 4.5, 405);
37 leg_pos = min(leg_pos + 5, 455);
```

*← higher segments fall slower*

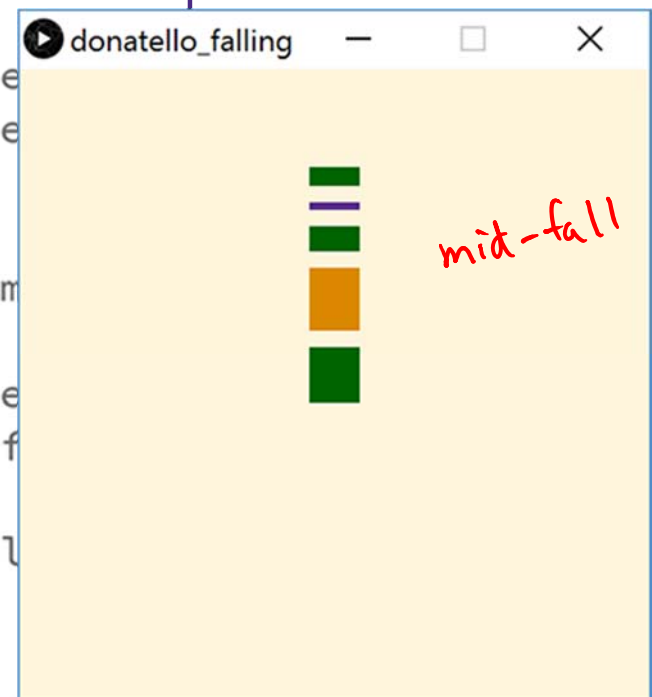
*← lower segments fall faster*

*variables that use decimals need to be declared as float*

# Falling Into Place

- ❖ Update y-positions of drawing based on new variables:

```
17 // Donatello
18 fill(0,100,0); // dark green
19 rect(x_pos, head_pos, 40, 15); // top of head
20
21 fill(88,44,141); // purple
22 rect(x_pos, mask_pos, 40, 6); // bandana mask
23
24 fill(0,100,0); // dark green
25 rect(x_pos, face_pos, 40, 20); // bottom of face
26
27 fill(219,136,0); // dark yellow
28 rect(x_pos, body_pos, 40, 50); // shell
29
30 fill(0,100,0); // dark green
31 rect(x_pos, leg_pos, 40, 45); // lower body
```



# Summary

- ❖ Variables are named quantities that can vary during the execution of a program
- ❖ Datatypes specific different forms of data
  - e.g. `int`, `float`, `color`, `Boolean`
- ❖ Variable *declarations* specify a variable datatype and name to the program
  - Generally occurs at top of program
- ❖ `min( )` and `max( )` functions can be used to limit or stop change in a variable value