

Creative Coding 2023

Instructor: Neng-Hao (Jones) Yu

Course website: <https://openprocessing.org/class/83620>

Announcement

- Change the EMI format to a regular class that will be taught in **Chinese**
- Provide **English slides**
- Feel free to let me know if you need any **additional English materials**
- Feel free to ask any questions you may have, **whether in English or Chinese**

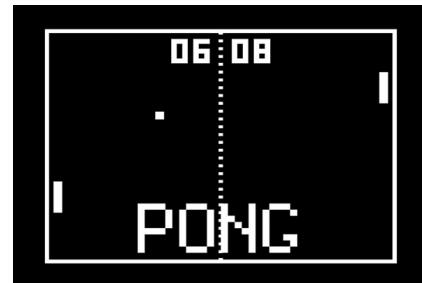
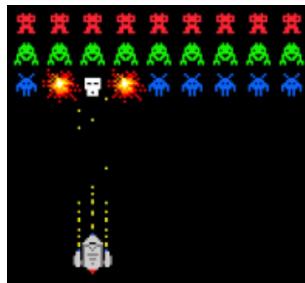
Course format

Lectures

+

In-class exercise

- ❑ Fundamental concepts
- ❑ Demo w/ games



- ❑ Hand-on coding challenges
- ❑ Completing challenges earns points
- ❑ Encourage peer support

Supplementary material

- <http://programming101.formosasoft.com/course/26>



Assignments



<https://programming101.ntust.edu.tw/games.html?nth=6>

Final group project



<https://www.youtube.com/watch?v=TBgJsPvM3EY>

Honor code

- ❑ Cheating, plagiarism, or any form of academic dishonesty will not be tolerated
- ❑ Violations of the honor code will be taken seriously and may result in penalties, up to and including failure of the course
- ❑ Fair use of code:
 - ❑ Provide proper attribution of the source code
 - ❑ Understand the code you used

Grading

- ❑ Code assignments (36%)
- ❑ In-class exercises (34%)
- ❑ Final group project (30%)

Recap of Week1

- ❑ Programs consist of lines of **statements**.
- ❑ Statements are **executed sequentially**, one after another.
- ❑ Code must follow the correct **syntax**, otherwise, you may encounter **syntax errors**.
- ❑ Processing provides a [reference page](#) where you can look up the syntax of each command.

Syntax

```
rect(a, b, c, d)
```

```
rect(a, b, c, d, r)
```

```
rect(a, b, c, d, t1, tr, br, b1)
```

Parameters

- a** (float) x-coordinate of the rectangle by default
- b** (float) y-coordinate of the rectangle by default
- c** (float) width of the rectangle by default
- d** (float) height of the rectangle by default

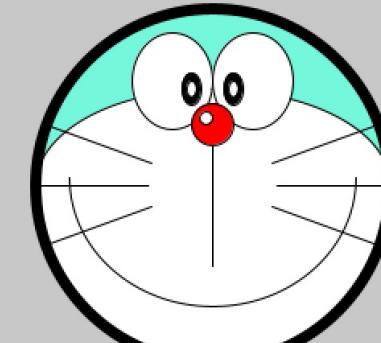
doraemo | Processing 4.1.3

```
//blue face  
strokeWeight(1);  
fill(#73F7DD);  
ellips(0,0,220,220);  
  
//white face  
strokeWeight(1);  
fill(255);  
arc(0,0,220,120,PI,TWO_PI); //upper  
arc(0,0,220,220,0,PI); //lower  
noFill();  
  
// black outline
```

The function ellips(int, int, int, int) does not exist.

Console Errors

Use **comments** to describe what the program does



The function ellips(int, int, int, int) does not exist.

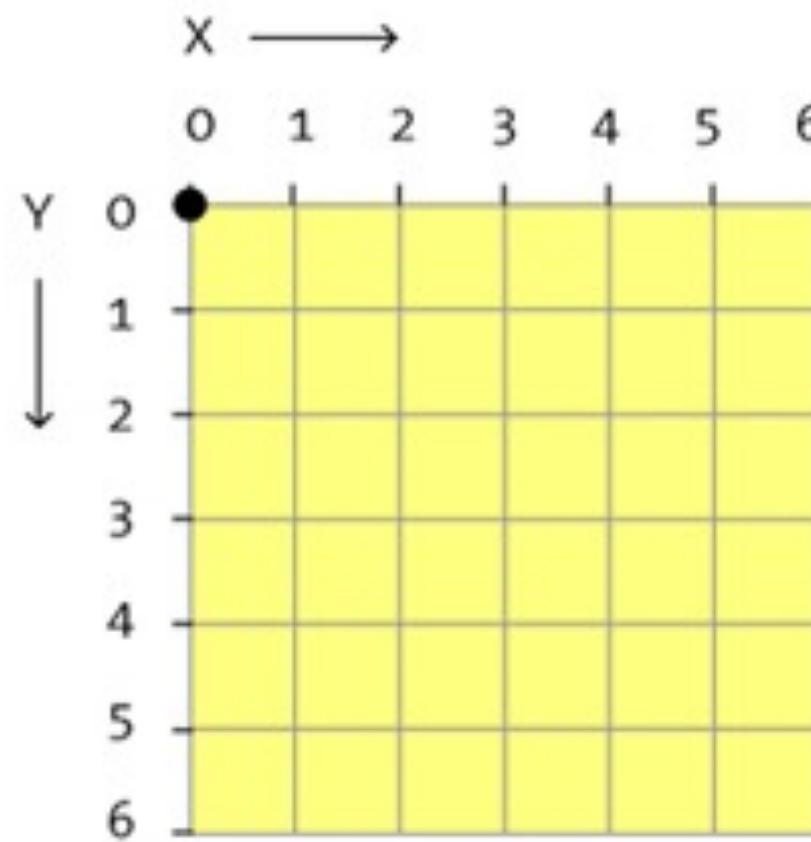
Complete statements should end with a **semicolon** ;



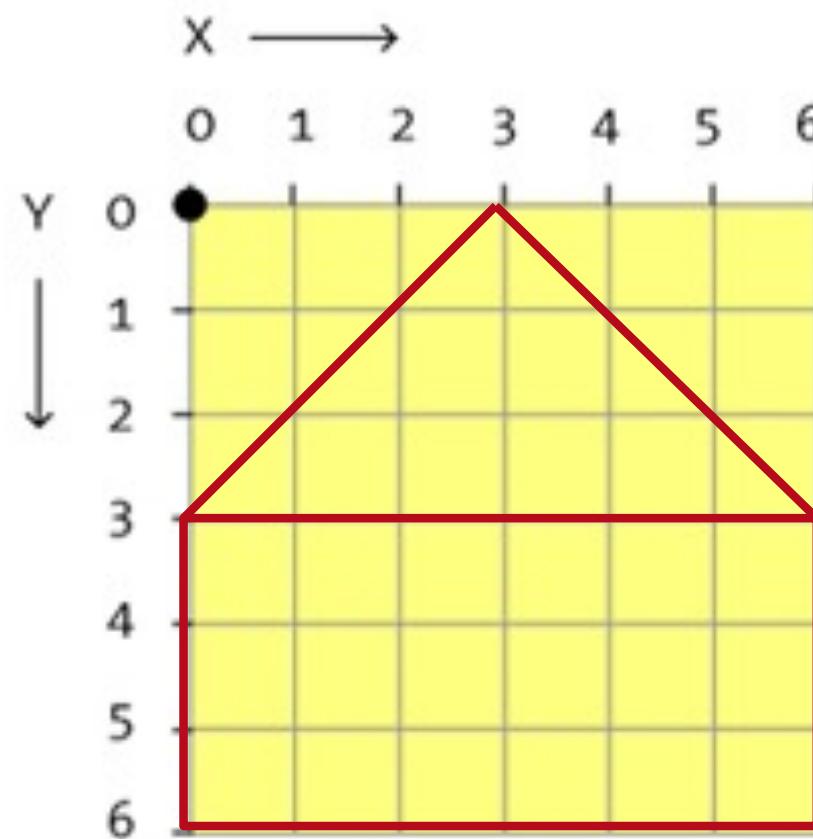
Set the canvas size

1. `size(640, 480);`
2. `//create a 640 x 480 canvas`

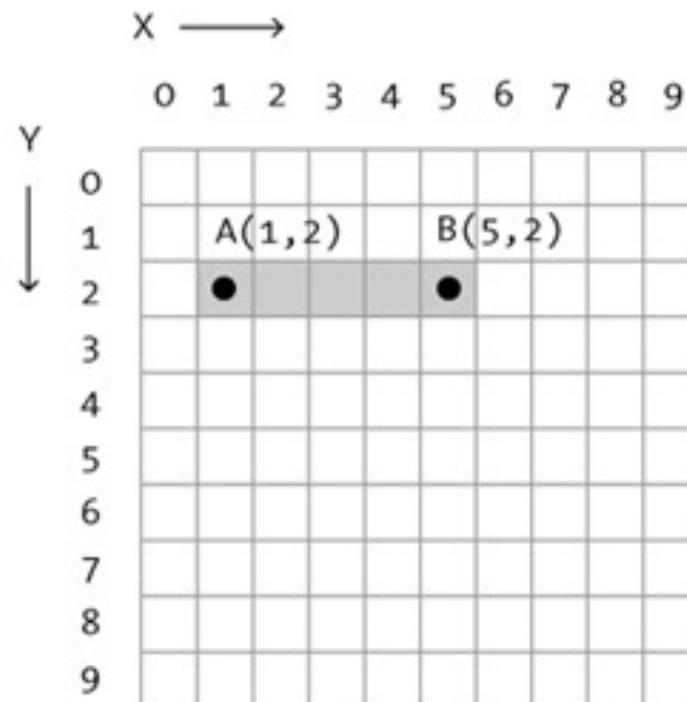
Screen coordinate system



Draw a house



Draw a line



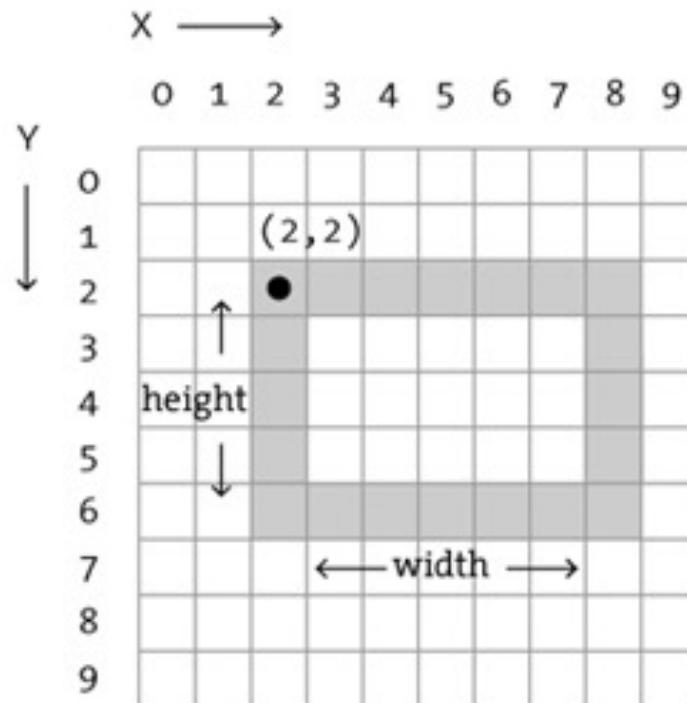
`line(x1,y1,x2,y2);`
Point A Point B

Example:
`line(1,2,5,2);`

```
// draw a line from point A to point B  
line(x1,y1,x2,y2);
```

http://processing.org/reference/line_.html

Draw a rectangle



`rect(x, y, w, h);`

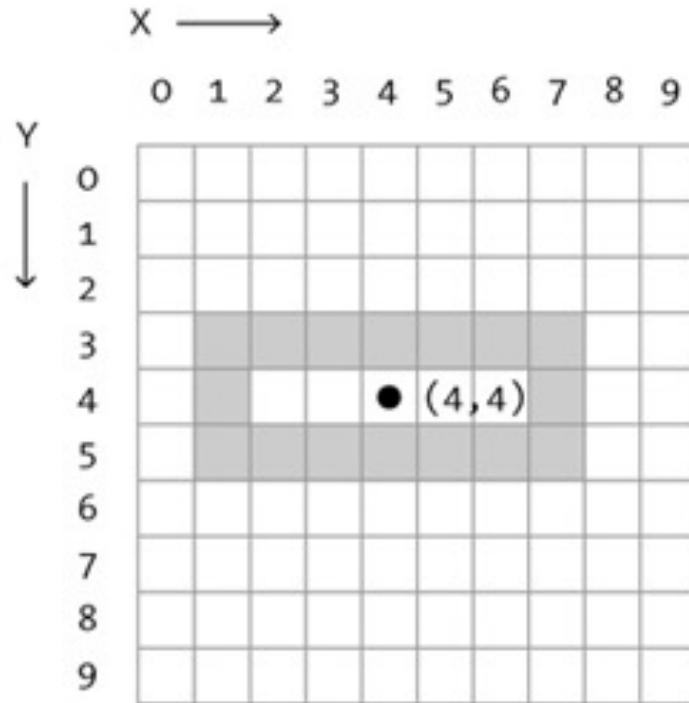
`rect(x,y,width,height);`

Example:

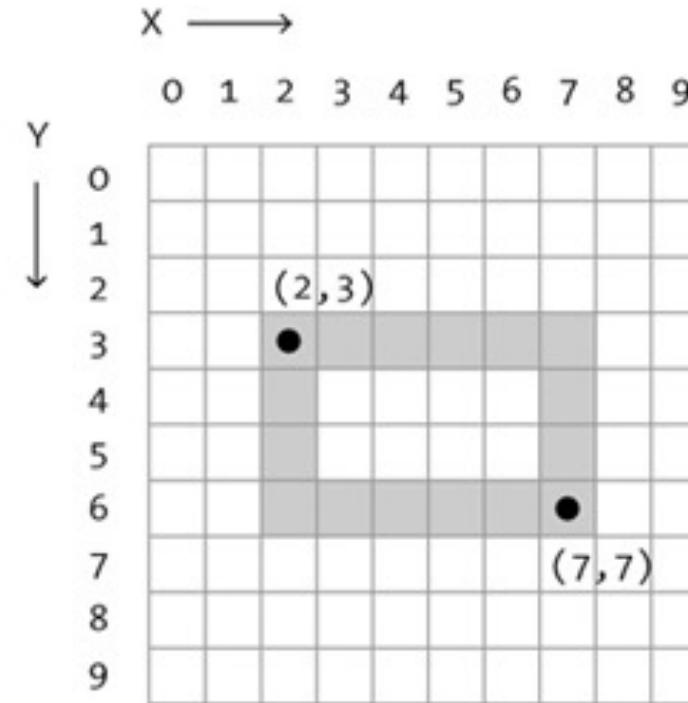
`rect(2,2,7,5);`

http://processing.org/reference/rect_.html

Rectangle mode



```
rectMode(CENTER);  
rect(4, 4, 7, 3)
```



```
rectMode(CORNERS);  
rect(x1,y1,x2,y2);
```

Example:
`rectMode(CORNERS);
rect(2,3,7,7);`

http://processing.org/reference/rectMode_.html

Change Rectangle mode

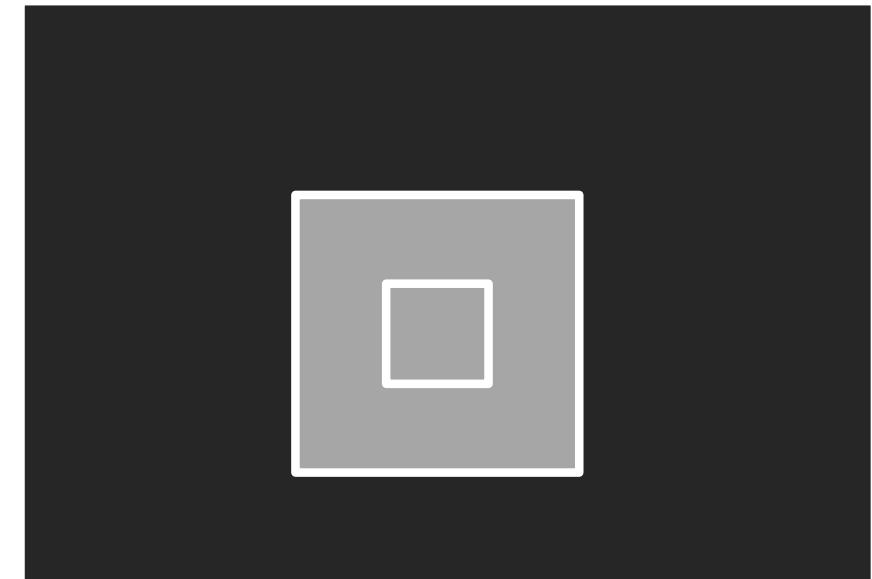
1. `rectMode(CORNERS);`
2. `rect(x1, y1, x2, y2);`
3. `rect(x1, y1, x2, y2);`
4. `rectMode(RADIUS); // change to RADIUS mode`
5. `rect(x1, y1, width, height);`

Switching **rectMode** will change the subsequent rect command, until a new rectMode is set

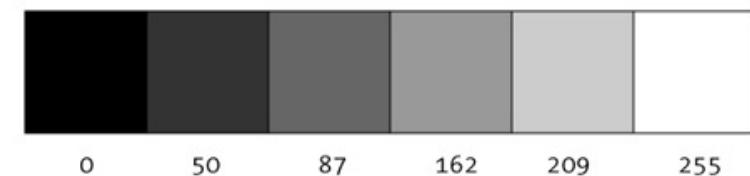


Grey color

1. background(50);
2. stroke(255);
3. fill(162);
4. rect(30,30,30,30);
5. rect(40,40,10,10);

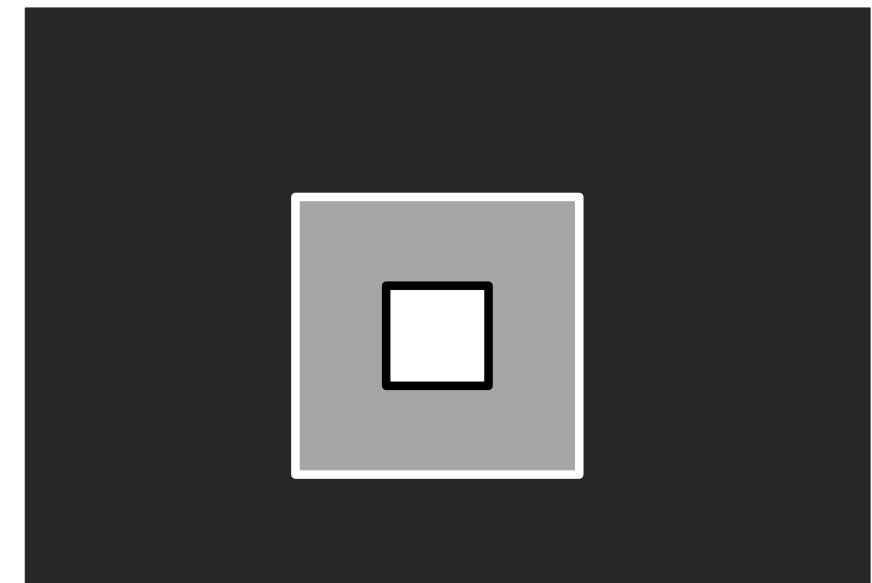


Grayscale

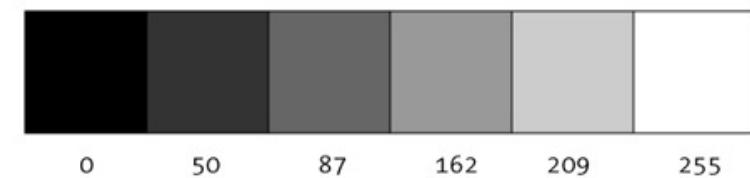


Change the stroke / fill color

1. `background(50);`
2. `stroke(255);`
3. `fill(162);`
4. `rect(30,30,30,30);`
5. `stroke(0);`
6. `fill(255);`
7. `rect(40,40,10,10);`



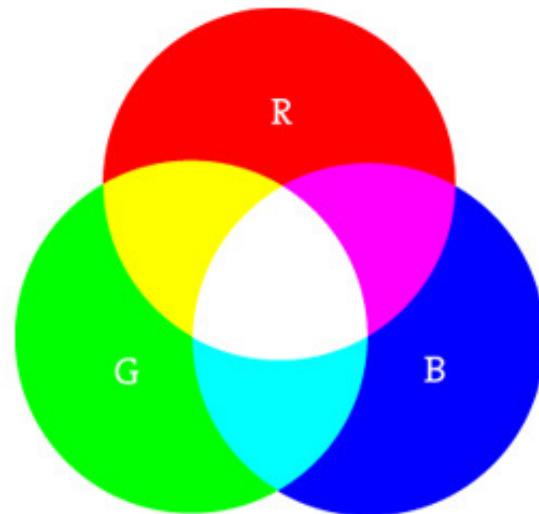
Grayscale



Color

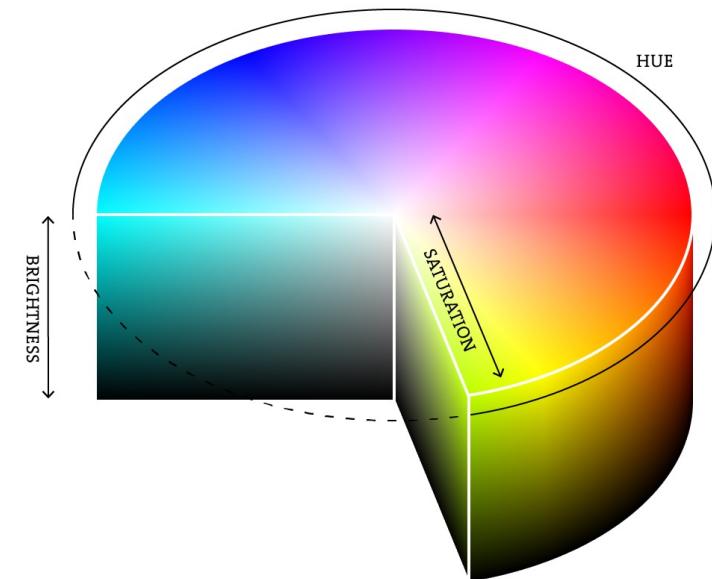
RGB Color

```
colorMode (RGB)  
fill (R, G, B);  
fill (255, 0, 0); //red
```



HSB Color

```
colorMode (HSB);  
fill (H, S, B);
```



https://processing.org/reference/colorMode_.html

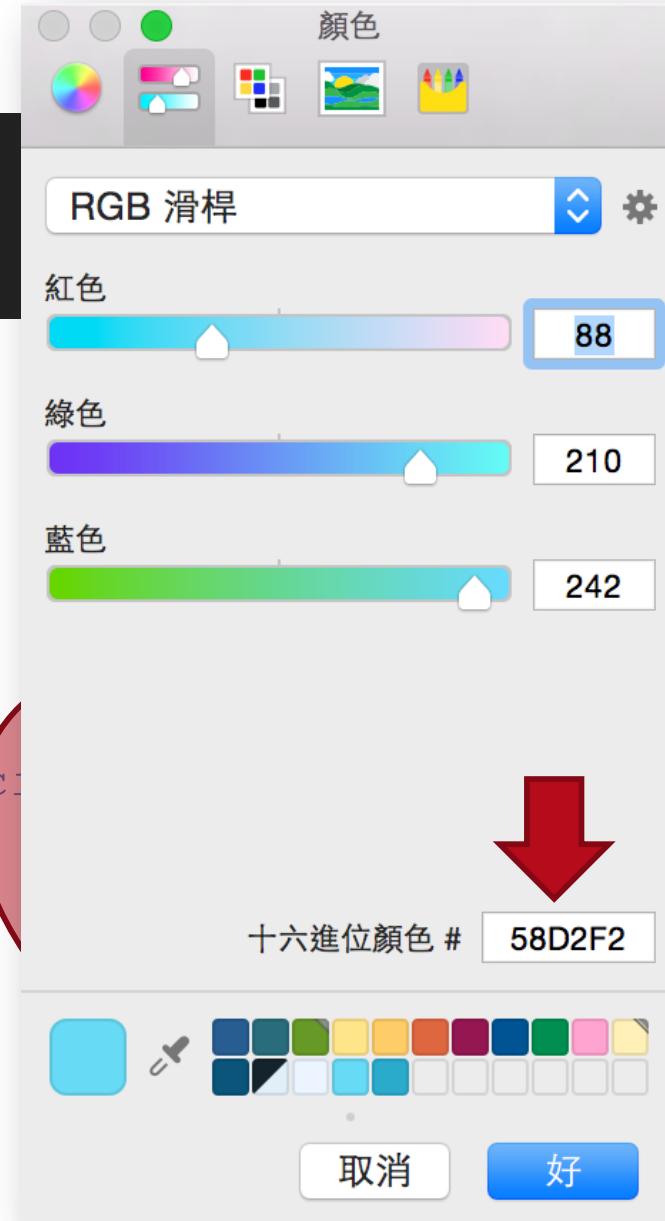
Color value

Hexadecimal notation

```
fill(#FF0000); // red
```

Alpha

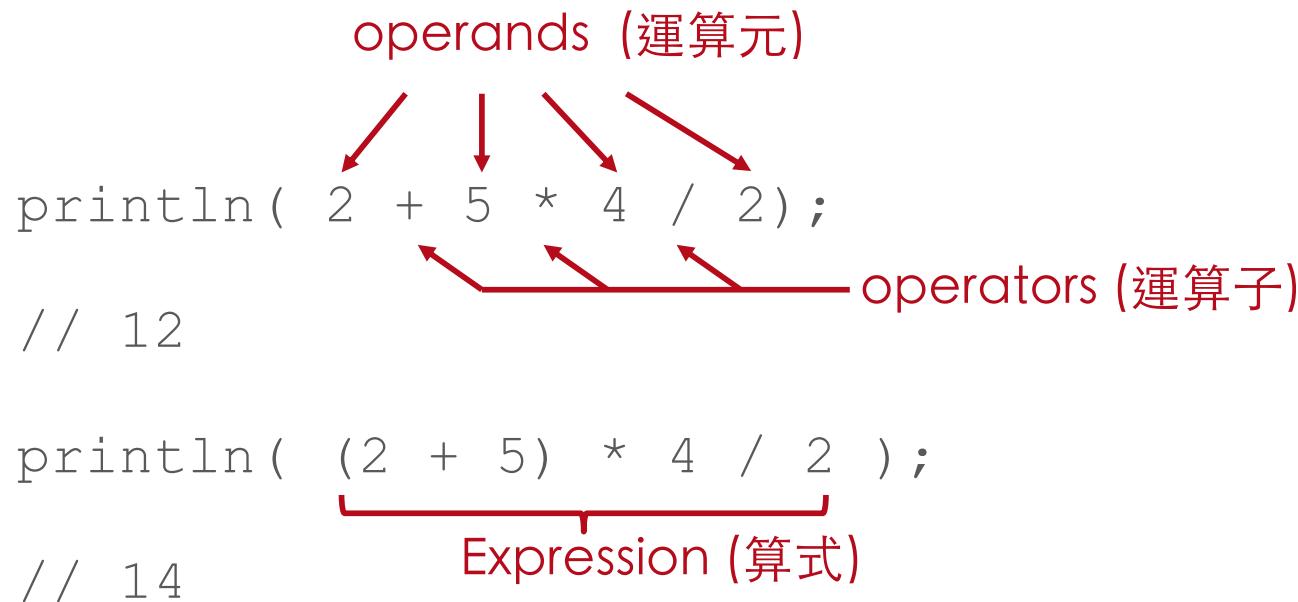
```
fill(#FF0000, 128); // opacity  
fill(255, 0, 0, 128);
```



https://processing.org/reference/fill_.html

Performing Calculations

```
operands (運算元)  
println( 2 + 5 * 4 / 2 );  
// 12  
operators (運算子)  
  
println( (2 + 5) * 4 / 2 );  
// 14  
Expression (算式)
```



Arithmetic operators (常用數學運算子)

- + Addition : Adds numeric expressions
- Subtraction : Negates or subtracts numeric expressions
- * Multiplication : Multiplies two numerical expressions
- / Division : Divides expression1 by expression2
- % Modulo : Calculates the remainder of division

Modulo operator

1. `println(11 % 5);`
2. // remainder = 1

- ❑ Clever use of remainders:
 - ❑ Generating a cyclical sequence (0,1,2,3,4,0,1,2,3,4,0,1,2....)
 - ❑ Calculating Row and Column Positions:
 - ❑ `11 / 5;` // row position: 2
 - ❑ `11 % 5;` // column position: 1

	0	1	2	3	4
0	0	1	2	3	4
1	5	6	7	8	9
2	10	n			
3					
4					

Storing data into memory

Variable (變數)

1. int x; // declare an integer: x

2. x = 2; // assign value 2 into x

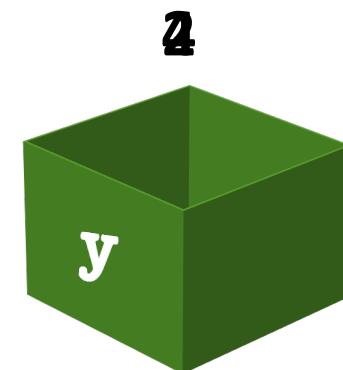
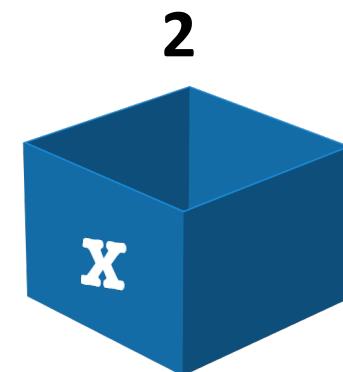
3. println(x + 5 * 4 / 2); // 12

4. int y;

5. y = x;

6. y = x * x; // 2*2

7. println(y); // 4



How to Use Variables

- Declare the variable (宣告變數)

Case sensitivity!



```
int myAge;
```

- Initialize the variable (變數初始化)

Data assignment (assignment operator)



```
myAge = 18; // assign value(right) to variable(left)
```

- Use the variable (使用變數)

```
println( 2023-myAge ); // birth year
```

ESP Game

- [1] First, choose a random number between 0 and 9.
- [2] Multiply this number by 2.
- [3] Add 5 to the result.
- [4] Multiply the new result by 50.
- [5] Add 1773 to the result.
- [6] Finally, subtract your birth year from the number obtained in step 5.

You should now have a three-digit number.
The first digit represents the initial number you chose, while
the second and third digits represent your age.

ESP Game

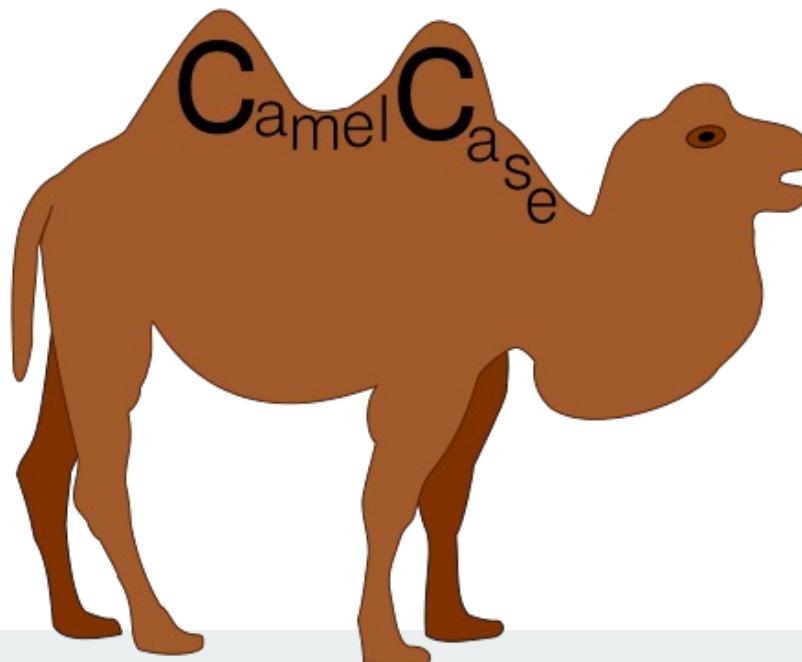
```
int guess = 3;  
int birthYear = 1998;  
int result;  
result = (guess*2 + 5)*50 + 1773 - birthYear;  
  
/* How does it work?  
result = (guess*100) + (2023-birthYear);  
*/  
  
println(result);
```

Variable naming rules

- Variable names must **start with a letter**, an **underscore**, or a **dollar sign**
- **No spaces** in the variable name
- Variable names must be **unique**
- Variable names **cannot be a reserved word** or keyword.
 - catch, this, class, throw ... (<http://processing.org/reference/>)
- Use **descriptive variable names** so that the content of any variable is obvious

Variable naming conventions

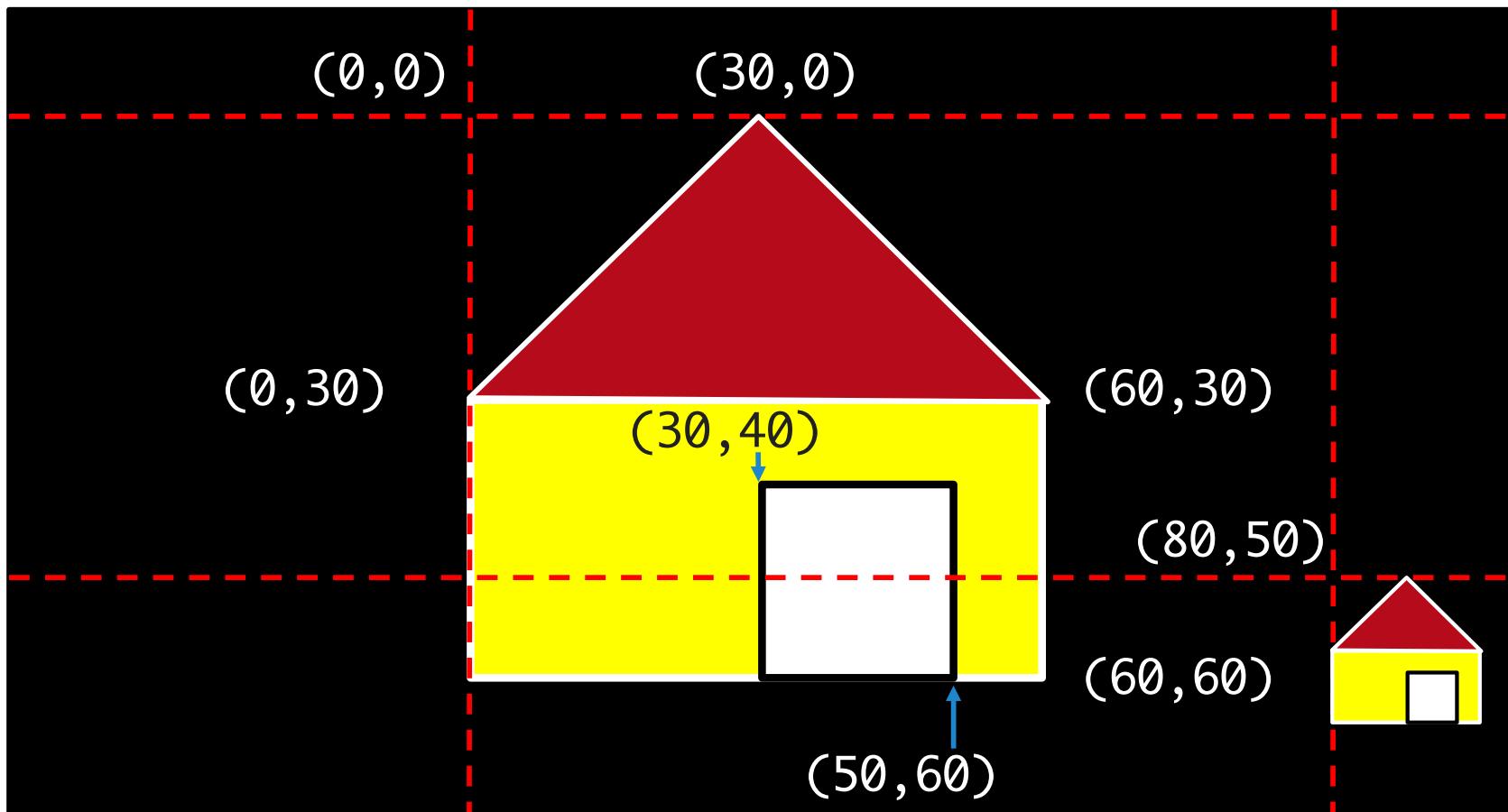
- ❑ Use **camel case**: Start with a lowercase letter and include uppercase letters.
- ❑ e.g. totalCost or mySecretNumber



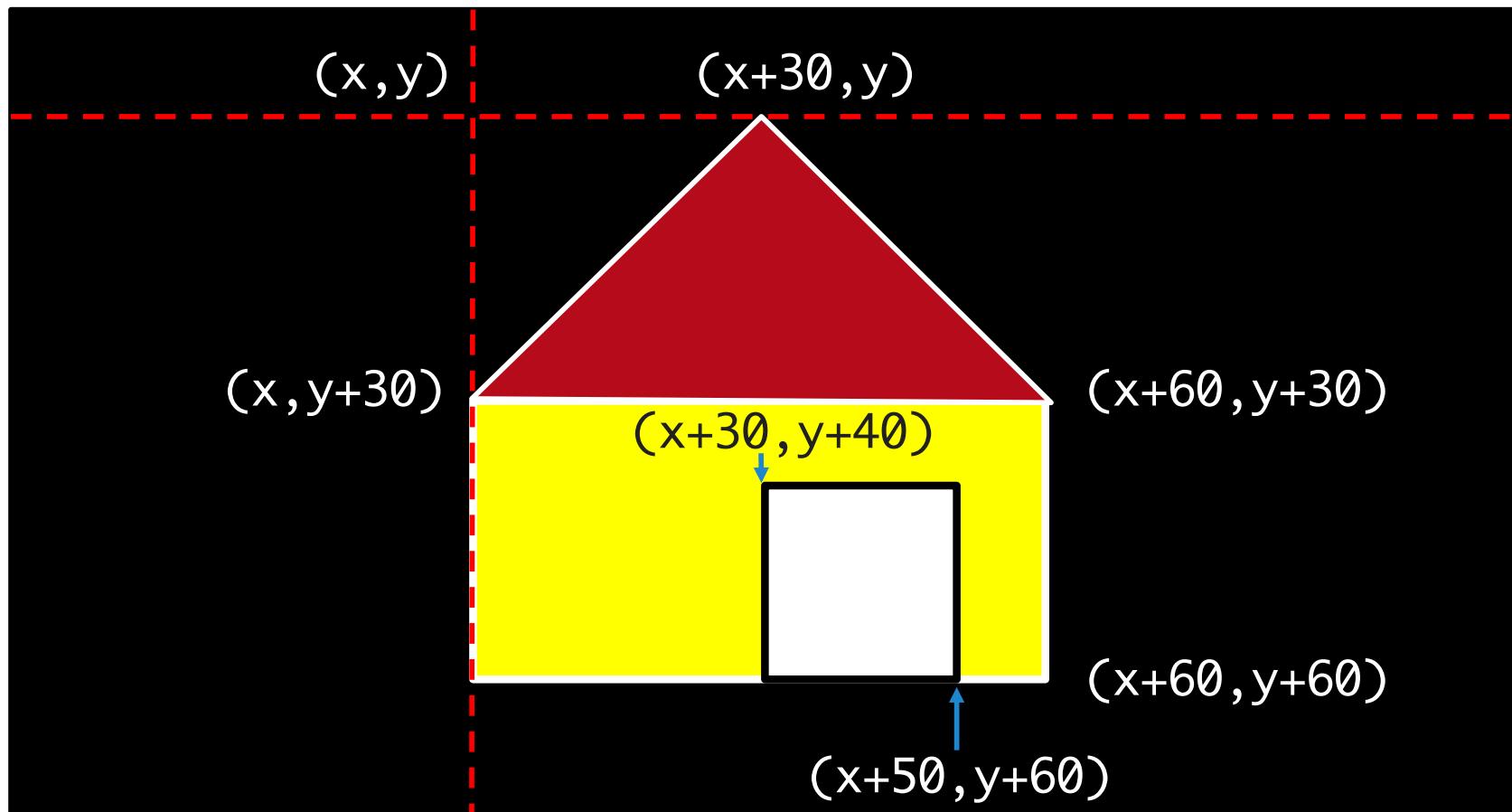
Valid Variable Names

- ❑ int face book; X → int facebook;
- ❑ int 3cats; X → int threeCats;
- ❑ int my_secret_number; ✓ → int mySecretNumber;
- ❑ int showMe\$; ✓ → int showMeMoney;
- ❑ int \$_\$; ✓ → Use descriptive names
- ❑ int goodName; ✓

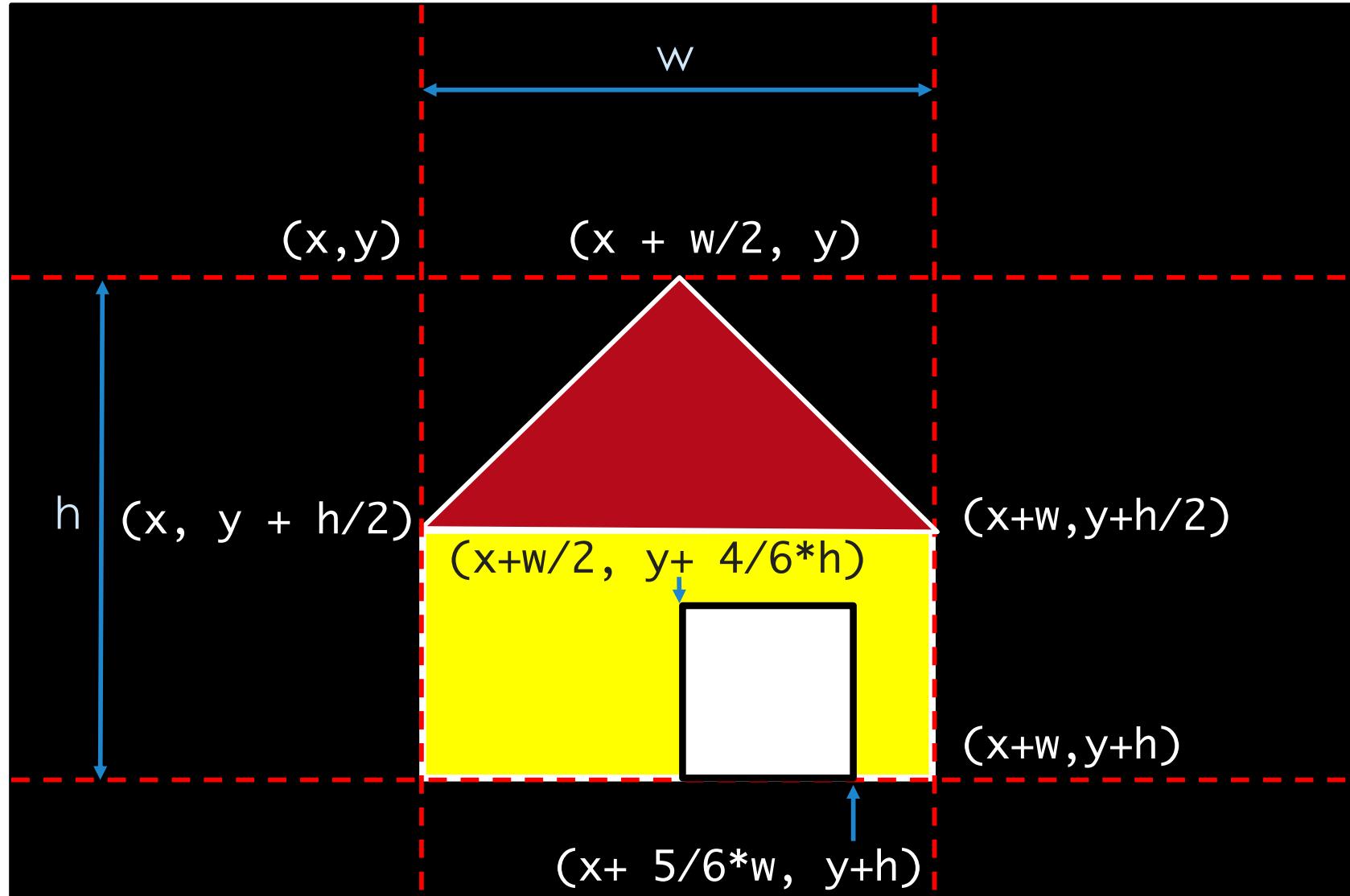
Exercise



Relative to the Origin

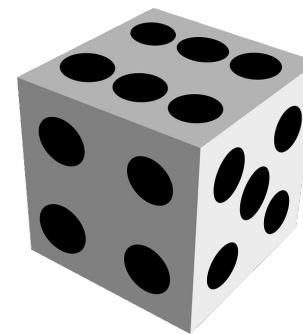


Scaling by a Specified Size Ratio



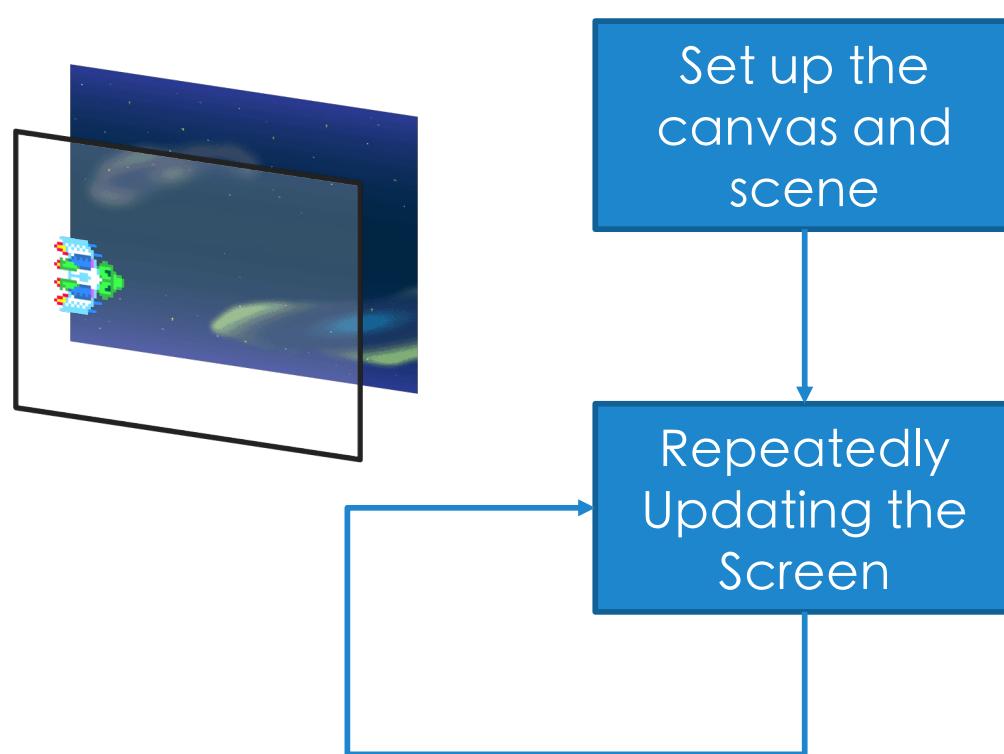
Random number (隨機變數)

□ `random(n)` // n=6: $0 \rightarrow 5.999999999$



□ `int dice = floor(random(6)); // 0,1,2,3,4,5`
無條件捨去

Animation



```
void setup(){
```

```
}
```

```
void draw(){
```

```
}
```

Function block

Static mode vs Dynamic mode

■ Static mode:

```
int x;  
  
size(300, 300);  
  
background(0);  
  
line(10,10,80,80);  
  
//execute statements  
//line by line
```

■ Dynamic mode:

```
int x;  
  
void setup() {  
    x = 1;  
    size(300, 300);  
}  
  
void draw() {  
    background(0);  
    ellipse(x++, 100, 80, 80);  
}
```

Initialization,
only executed once
at the beginning

Repeatedly executing
over time

Increment / Decrement operator

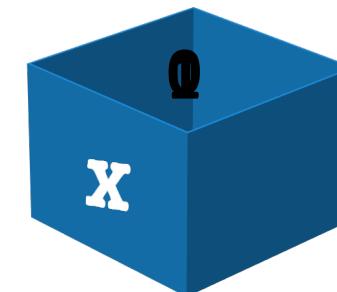
❑ `x = x + 1;`

→ `x++;`

❑ `x = x - 1;`

→ `x--;`

`0+1`



`++` Increment : Adds 1 to a numeric expression

`--` Decrement : Subtracts 1 from a numeric expression

Compound assignment operators

```
x += 5;           // x = x + 5;
```

```
x -= 6;           // x = x - 6;
```

```
x *= x;           // x = x * x;
```

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

```
x %= width; // x = x % width;
```

Postfix vs Prefix

```
int x = 0;  
println(x++); // 0  
println(x); // 1
```

{
 println(x);

 x = x + 1;

Big Difference
between Pre-Fix
and Post-Fix

```
int y = 0;  
println(++y); // 1  
println(y); // 1
```

{
 y = y + 1;

 println(y);



Operator precedence

```
int x = 1;
```

```
int y = (2 + 3) * 4 / x++ ;
```

```
println (x); // 2
```

```
println (y); // 20
```

Name	Symbol	Examples
Parentheses	()	a * (b + c)
Postfix, Prefix	++ -- !	a++ --b !c
Multiplicative	* / %	a * b
Additive	+ -	a + b
Relational	> < <= >=	if (a > b)
Equality	== !=	if (a == b)
Logical AND	&&	if (mousePressed && (a > b))
Logical OR		if (mousePressed (a > b))
Assignment	= += -= *= /= %=	a = 44

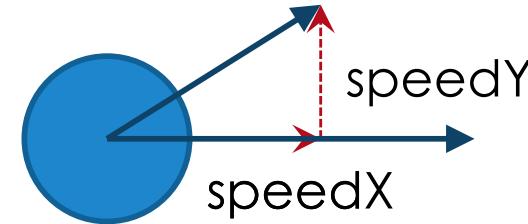


Exercise (Pong: game start)

- Generate a ball in the center of the canvas and move it in a random direction and speed

Hint:

- random(a,b) → a~b之間的任意數，但不包含 b
-



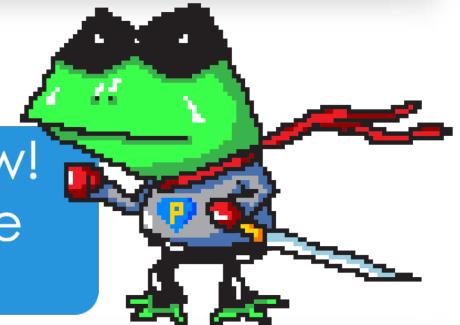
```
x += speedX;  
y += speedY;
```

Load Image Files and Draw them onto the Canvas

```
// Declaring a variable of type PImage  
1 PImage img;  
  
void setup() {  
    size(320,240);  
    // Make a new instance of a PImage by loading an image file  
2    img = loadImage("mysummervacation.jpg");  
}  
  
void draw() {  
    background(0);  
    image(img,0,0);  
3}
```

Declaring a variable of type `PImage`, a class available to us from the *Processing* core library.

Reading files is slow!
Read the file once
in the `setup()`



The `image()` function displays the image at a location—in this case the point (0,0).

<https://openprocessing.org/sketch/1856800>

Recap

- ❑ Operators
- ❑ Variables
- ❑ Random numbers
- ❑ Basic concept of animation
- ❑ Read and show image files

Assign1 – part 1/2

- ❑ [Fork assignment here.](#)
- ❑ Commit your code directly to the GitHub
- ❑ Due: 3/19 12pm
- ❑ Grading rubric :
 - ❑ C: Completing all 1-star requirements.
 - ❑ B: Completing all 1-star and 2-star requirements.
 - ❑ A: Completing all requirements (1-star, 2-star, and 3-star)



Assign1 – part 2/2

- ❑ Requirements :

- ❑ 1-star

- ❑ Add a spaceship to the right-center of the screen.
 - ❑ Add a health bar to the left-top corner of the screen at position (10, 10).
 - ❑ Show the player's health using a red bar that fills up to 50%.
 - ❑ Add a treasure to a randomly generated position on the screen.
 - ❑ Add comments and label the code related to the requirements.
 - ❑ Variable names should be able to properly represent their meaning and be named using camelCase.

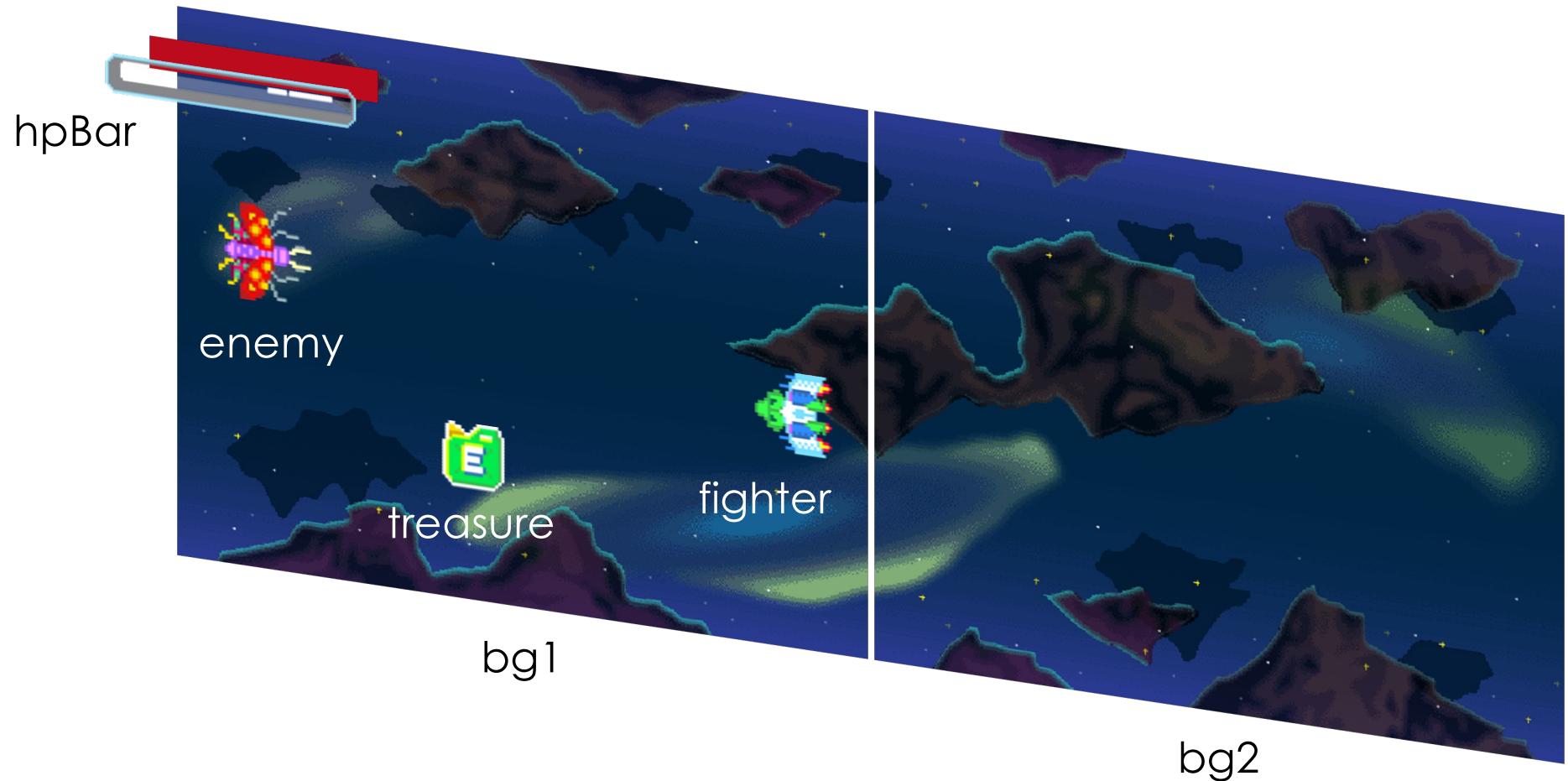
- ❑ 2-star

- ❑ Place an enemy aircraft that flies repeatedly from left to right.
 - ❑ After the enemy aircraft completely exits the right side of the screen, it restarts flying from the left side.

- ❑ 3-star

- ❑ Add a background and scroll it infinitely to the right

Hint



* Use the modulo operator to achieve the infinite scrolling effect