Week 10

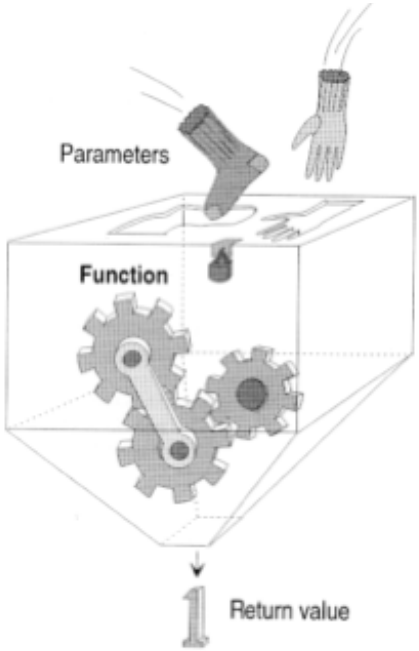
範例程式: https://github.com/jonesfish/programming101

函式 (Functions)

- 可重複使用的程式碼模組
  - rect()
  - line()
  - point()
  - background()

函式 (Functions)

- 可自行指定輸入並回傳數值
  - `int a = abs(-15);`
  - `float r = radians(135);`
  - `float n = dist(x1, y1, x2, y2);`



為什麼要使用函式

模組化 Modularity

- Break down code into smaller parts
- More manageable and readable
- Easy to debug

為什麼要使用函式

可重複使用 Reusability

- Duplicated code (copy/paste?) is not good
  - Need to maintain it in multiple places
- Better to put duplicate code in a new function and ‘call’ it from multiple places

宣告／呼叫自訂函式


```
void sayHi() {
    println("Hi");
}
sayHi();     // Hi
sayHi();     // Hi
sayHi();     // Hi
```

參數傳遞


```
void sayHi(String name) {
    println("Hi " + name);
}
sayHi("Jones");    // Hi Jones
sayHi("Alice");    // Hi Alice
sayHi("Joe");      // Hi Joe
```

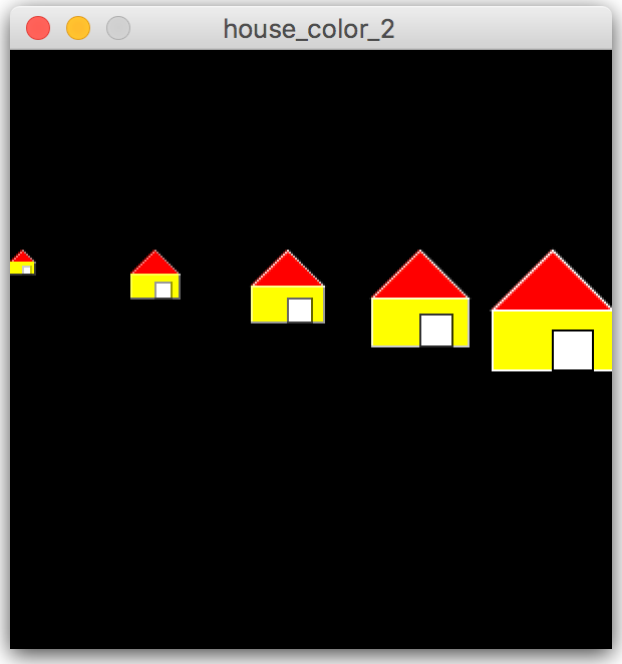
參數傳遞

return type　　function name　　parameter / argument

```
void circle(int x, int y, int diameter) {
  ellipse(x, y , diameter, diameter);
}

circle(150, 150, 50);
```

Exercise

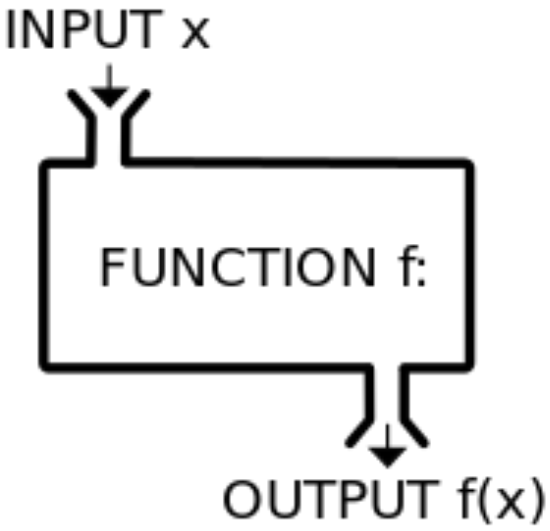
- house(x,y);
- house(x,y,size);

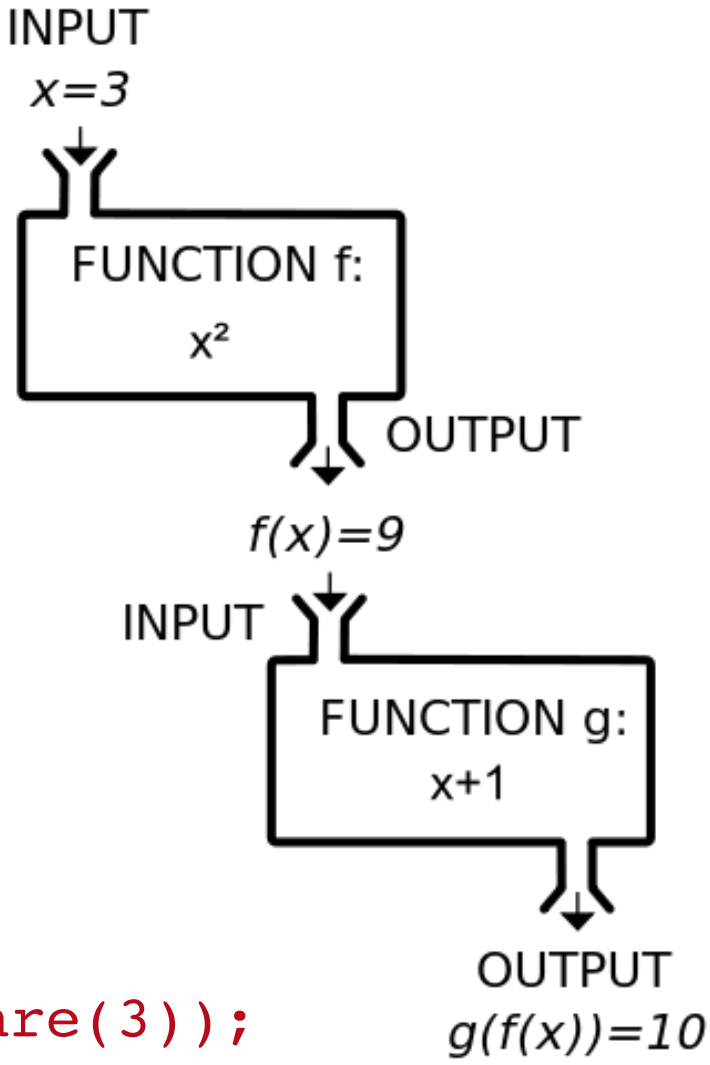


回傳值


```
int square(int num) {

    return(num * num );

}

int area = square(5); // call function

//returns the int value 25
```

連續運算





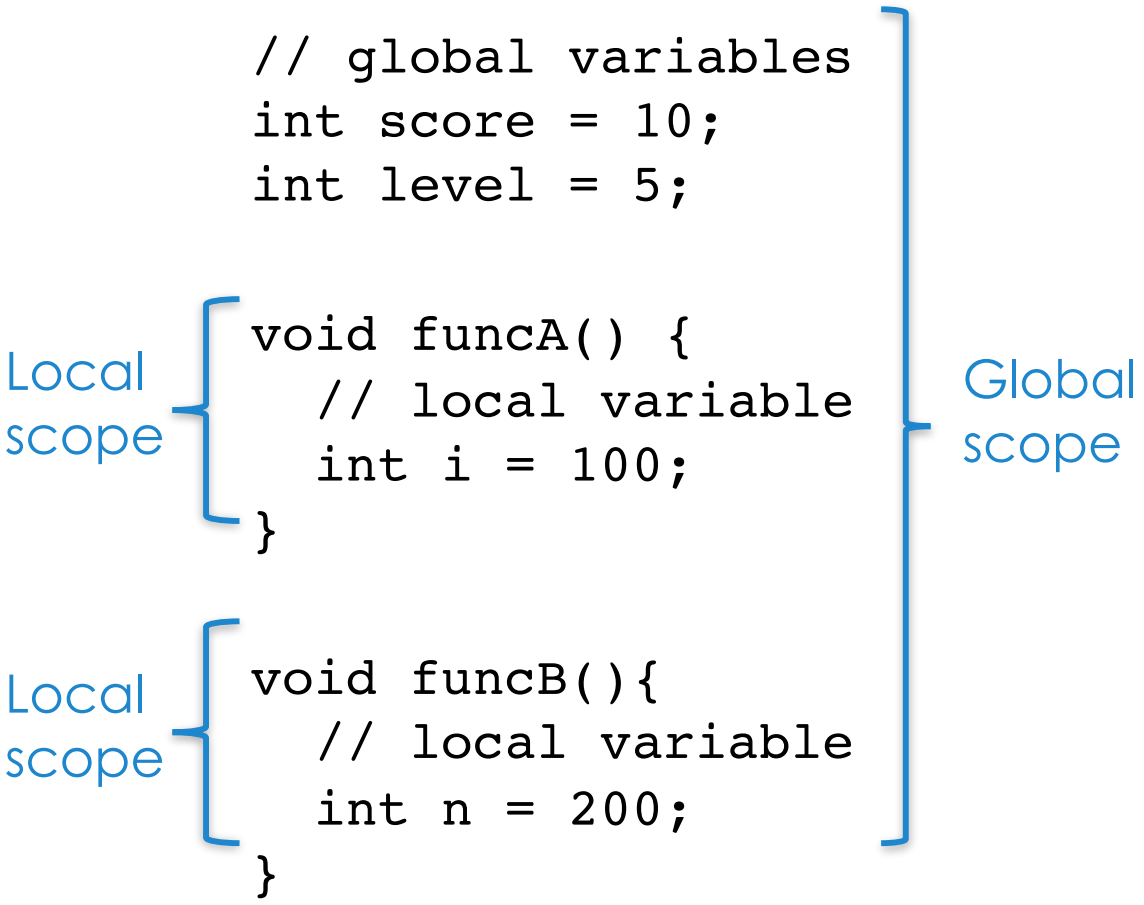

```
int y = add1(square(3));
```

Exercise

- 計算三角型面積
  - `input: base and height`
  - `output: triangle's area`

- 計算 Array中所有數值的和
  - `input: integer array`
  - `output: sum`

變數的有效範圍 (Variable Scope)



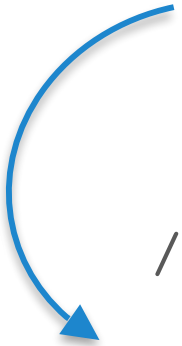
Function scope

Local scope

```
float area(float hh, float ww) {

    return(hh * ww );

}

float rect = area(6,5);

//returns the float value 30


println (hh); // error: undefined variables
```

The return statement terminates the function

```
float area(float hh, float ww) {
    return(hh * ww );
    println("after return");
 // Error: unreachable code.
}
```



Be aware of incomplete return

```
Boolean isOdd(int num) {
    if (num % 2 == 0){
      return false;
    }
}
println(isOdd(2));
// error: Function does not return a value.
```

Be aware of incomplete return

```
Boolean isOdd(int num) {
    if (num % 2 == 0){
      return false;
    } else{
      return true;
    }
}
```

Exercise

大樂透是一種樂透型遊戲。您必須從01~49中任選6個號碼進行投注。開獎時，開獎單位將隨機開出六個號碼，這一組號碼就是該期大樂透的中獎號碼，也稱為「獎號」。

您的六個選號中，如果有三個以上（含三個號碼）對中當期開出之六個號碼（特別號只適用於貳獎、肆獎、陸獎和柒獎），即為中獎，並可依規定兌領獎金。

Function overloading

- one function can perform different tasks.
  - same function name
  - different types of input/output
  - different numbers of inputs

```
int addXY(int x, int y){
    return x+y;
}

println( addXY(3, 2) );
// 5
```

```
float addXY(float x, float y){
    return x+y;
}

println( addXY(5.5, 2) );
// 7.5
```

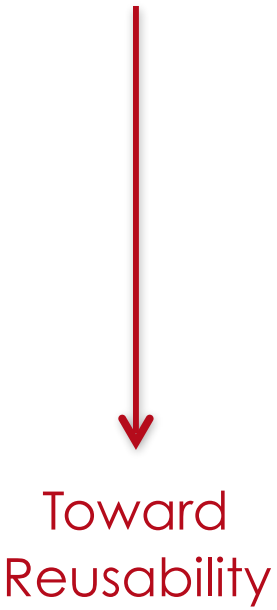
Pass by value vs pass by reference

- Primitive data types are passed by value
  - String, int, float, Boolean
- Complex data types are passed by reference
- Incidentally, these rules apply to variable assignments, too.

```
float cArea(float r) {
  return (PI * r * r);
}
float radius = 10;
println( cArea(raduis) );
```


```
void shuffle(int[]arr){
… }
int[] myArray= {0,1,2};


shuffle(myArray);
```

Define functions in practice



- a block of codes
  - no input parameters
  - no output value
- a dynamic module
  - with input parameters
  - output the computed value

- A good function does only one task