Using Methods

Writing your own Methods

Lecturer: Caio Fonseca

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Department of Computing and Mathematics
http://www.wit.ie/

Topics List

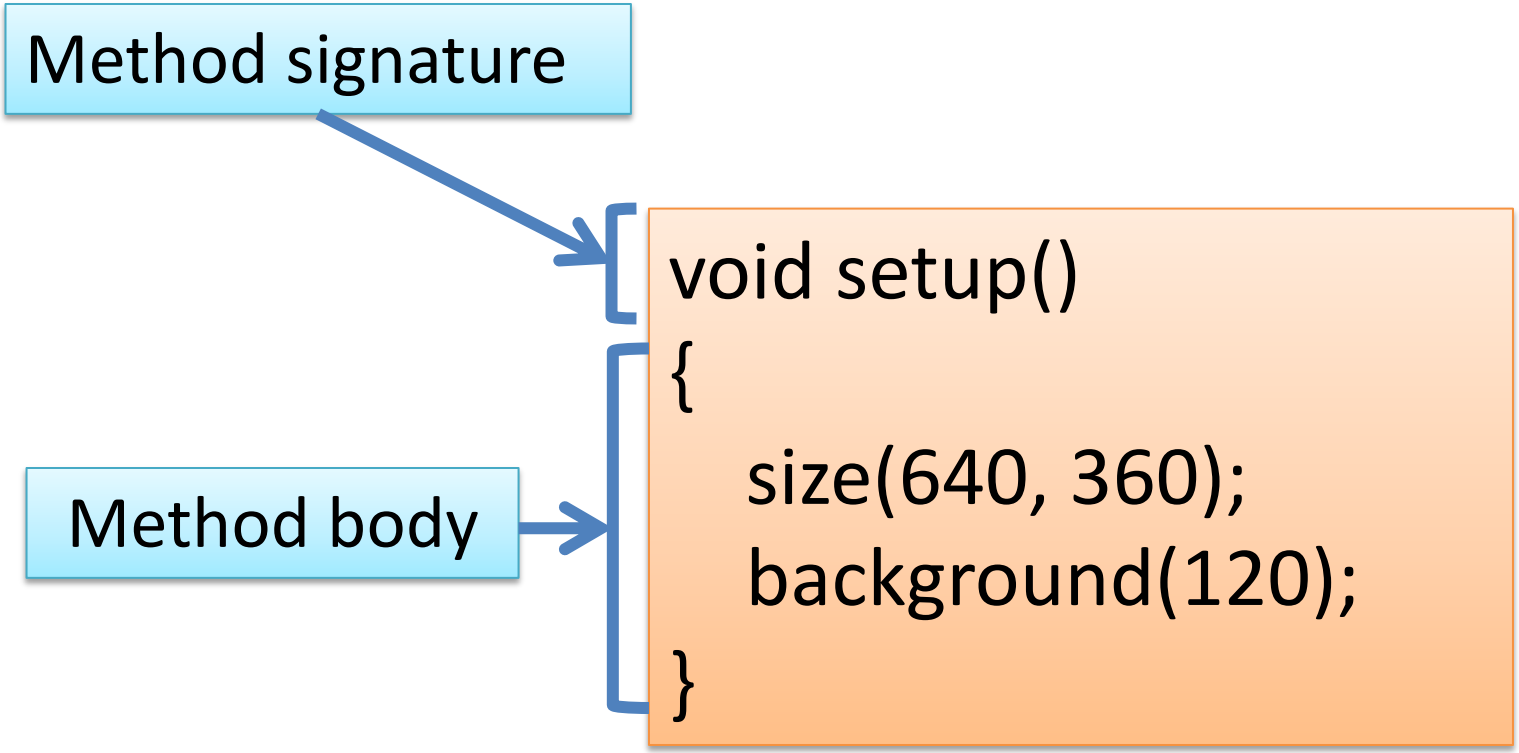
- Recap of method terminology:
  - Return type
  - Method names
  - Parameter list

- Writing your own methods:
  - With no parameters
  - With parameters

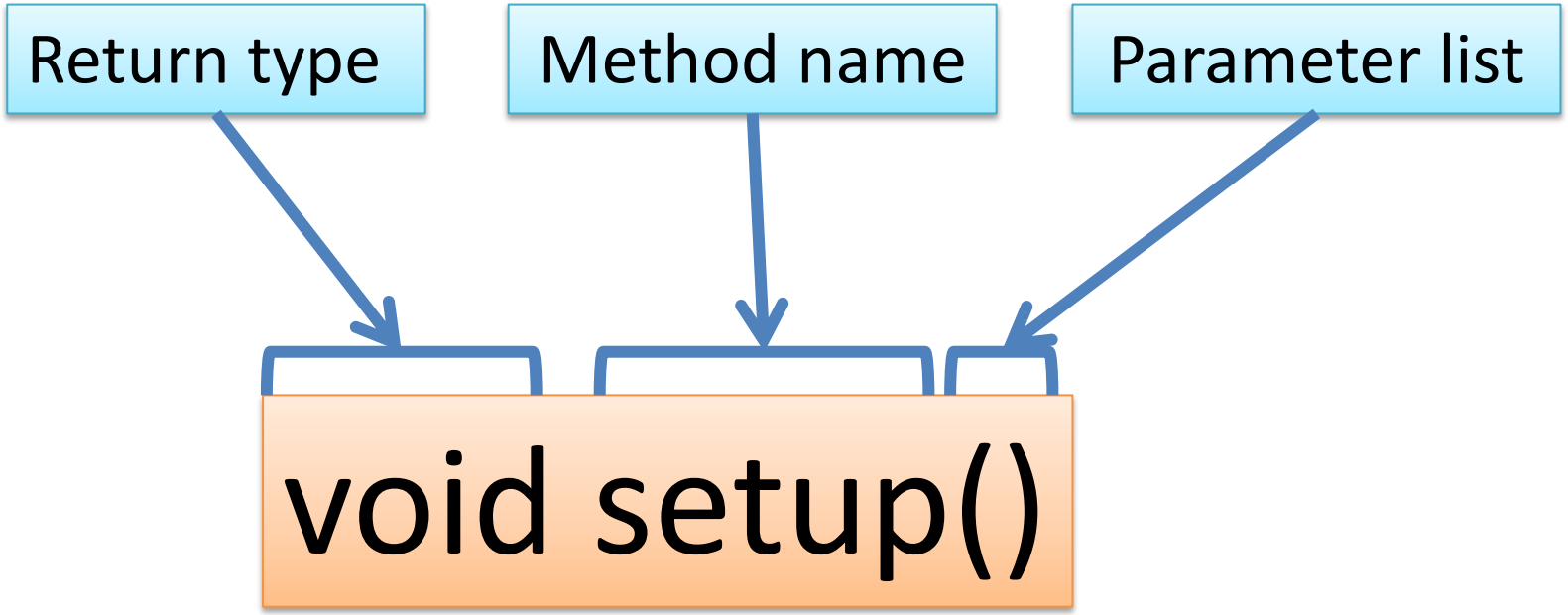
Recap: Methods in Processing

- A methods comprises a set of instructions that performs some task.
- When we invoke the method, it performs the task.
- Some methods we have used are:
  - rect, ellipse, stroke, line, fill, etc.
  - void mousePressed()
  - void setup, void draw()

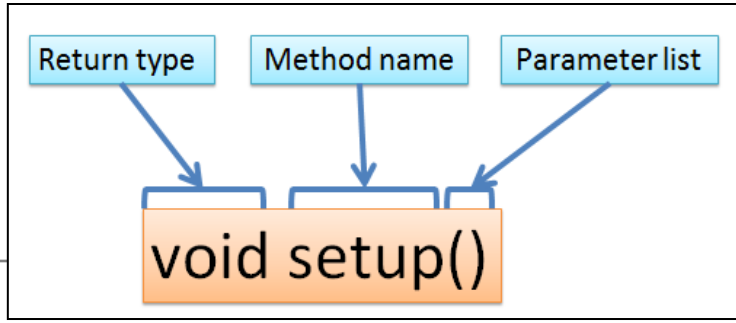
Recap: Method terminology



Recap: Method signature



Recap: Return Types



- Methods can return information.
- The void keyword means that nothing is returned from the method.
- When a data type (e.g. int) appears before the method name, this means that something is returned from the method.
- Within the body of the method, you use the return statement to return the value.
- You can only have one return type per method.
- Methods can return any type of data e.g. boolean, byte, char, int, float, String, etc.

Recap: Return Types

```
int val = 30;


void draw()
{
    int result = timestwo(val);
    println(result);
}
```

```
int timestwo(int number)
{
   number = number * 2;
   return number;
}
```

// The red **int** in the function declaration
// specifies the type of data to be returned.

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

Recap: Method name

- Method names should:
  - Use verbs (i.e. actions) to describe what the method does e.g.
    - calculateTax
    - printResults
  - Be mixed case with the first letter lowercase and the first letter of each internal word capitalised.

Recap: Parameter list

- Methods take in data via their parameters.

Methods do not have to pass parameters. These methods don't need any additional information to do its tasks.

If a method needs additional information to execute, we provide a parameter so that the information can be passed into it. A method can have any number of parameters.

```
void makeInvisible()
void makeVisible()
void moveDown()
```

```
void moveVertical(int distance)
void slowMoveHorizontal(int distance)
void slowMoveVertical(int distance)
```

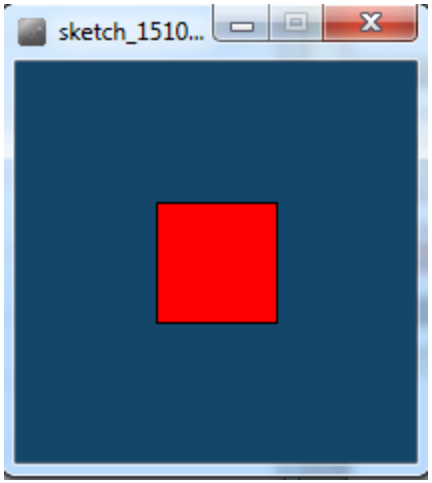
Topics list

- Recap of method terminology:
  - Return type
  - Method names
  - Parameter list

- Writing your own methods:
  - With no parameters
  - With parameters
  - That return data

Writing Methods with NO parameters

- Draw a red square at certain (x, y) coordinates.



Example 5.2

```
void setup()
{
  size(200,200);
  background(20,70,105);
}
```

```
void draw()
{
   drawRedSquare();
}

void drawRedSquare()
{
  fill(255,0,0);
  rect(70,70,60,60);
}
```

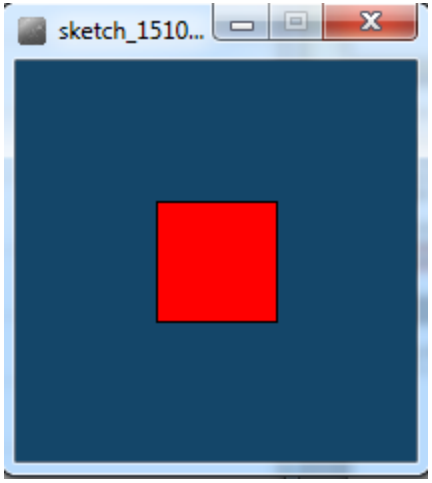
Topics List

- Recap of method terminology:
  - Return type
  - Method names
  - Parameter list

- Writing your own methods:
  - With no parameters
  - With parameters
  - That return data

Writing methods WITH parameters

- Now update the code so that you can pass in the length of the square into the method, drawRedSquare.



Example 5.3

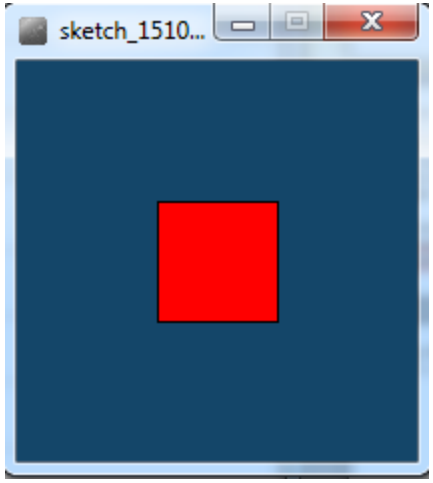
```
void setup()
{
  size(200,200);
  background(20,70,105);
}
```

```
void draw()
{
   drawRedSquare(60);
}

void drawRedSquare(int length)
{
  fill(255,0,0);
  rect(70,70,length, length);
}
```

Writing methods WITH parameters

- Now update the code so that you can pass in the:
  - length of the square
  - xCoordinate of the square
  - yCoordinate of the square
- into the method, drawRedSquare.



Example 5.4

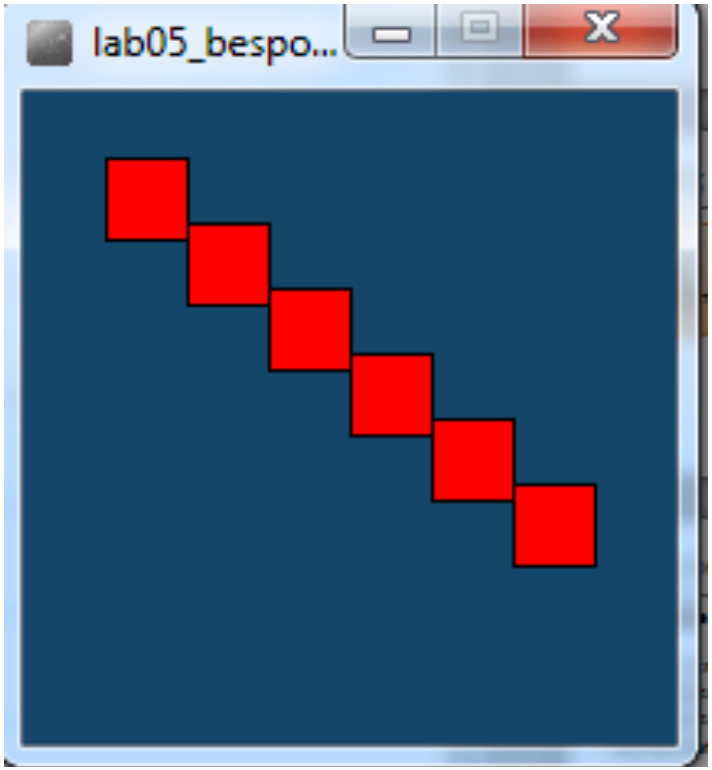
```
void setup()
{
  size(200,200);
  background(20,70,105);
}
```

```
void draw()
{
   drawRedSquare(60, 70, 40);
}

void drawRedSquare(int length, int xCoord, int yCoord)
{
    fill(255,0,0);
    rect(xCoord,yCoord, length, length);
}
```

Writing methods with parameters

- Now update the code so that you can call the drawRedSquare multiple times (using a loop).



Example 5.5

```
void setup()
{
  size(200,200);
  background(20,70,105);
}
```

```
void draw(){
   for (int i = 1; i < 7; i++)
   {
      drawRedSquare(25,  i*25,  i*20);
   }
}

void drawRedSquare(int length, int xCoord, int yCoord){
    fill(255,0,0);
    rect(xCoord,yCoord, length, length);
}
```

Example 5.6a

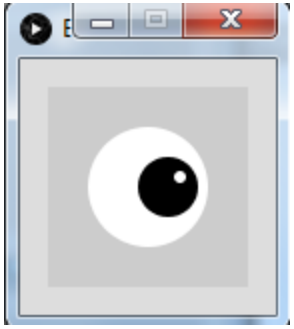
Ex_01 | Processing 3.0b6

File Edit Sketch Debug Tools Help

Java ▾

Ex_01

```
1  void setup() {
2    size(100, 100);
3    noStroke();
4  }
5
6  void draw() {
7    background(204);
8    fill(255);
9    ellipse(50, 50, 60, 60);     // White circle
10   fill(0);
11   ellipse(50+10, 50, 30, 30);  // Black circle
12   fill(255);
13   ellipse(50+16, 45, 6, 6);    // Small, white circle
14 }
15
```



Q. What if you wanted two eyes?

Example 5.6b

Ex_02 | Processing 3.0b6

File Edit Sketch Debug Tools Help

Java ▾

Ex_02

```
1  void setup() {
2    size(100, 100);
3    noStroke();
4  }
5
6  void draw() {
7    background(204);
8    // First eye
9    fill(255);
10   ellipse(65, 44, 60, 60);
11   fill(0);
12   ellipse(75, 44, 30, 30);
13   fill(255);
14   ellipse(81, 39, 6, 6);
15   // Second eye
16   fill(255);
17   ellipse(20, 50, 60, 60);
18   fill(0);
19   ellipse(30, 50, 30, 30);
20   fill(255);
21   ellipse(36, 45, 6, 6);
22 }
```

A1. You write the code a second time with different values.

A2. Or even better, instead of writing the code a second time, we write the code once in a method and call it twice.

Example 5.6c

Ex_03 | Processing 3.0b6

File Edit Sketch Debug Tools Help

Java ▾

Ex_03

```
1  void setup() {
2    size(100, 100);
3    noStroke();
4  }
5
6  void draw() {
7    background(204);
8    eye(65, 44);   // call the eye method for first eye
9    eye(20, 50);  // call the eye method for second eye
10 }
11
12 void eye(int x, int y) {
13   fill(255);
14   ellipse(x, y, 60, 60);
15   fill(0);
16   ellipse(x+10, y, 30, 30);
17   fill(255);
18   ellipse(x+16, y-5, 6, 6);
19 }
```

- This time we wrote the code once and called the eye method twice.
  - The first time we sent in 2 values (parameters) 65 and 44.
  - The first time we sent in 2 values (parameters) 20 and 50.

Example 5.6d

Ex_04 | Processing 3.0b6

File Edit Sketch Debug Tools Help

Java

Ex_04

```
1  void setup() {
2    size(100, 150);
3    noStroke();
4  }
5
6  void draw() {
7    background(204);
8    eye(65, 44);
9    eye(20, 50);
10   eye(65, 74);
11   eye(20, 80);
12   eye(65, 104);
13   eye(20, 110);
14 }
15
16 void eye(int x, int y) {
17   fill(255);
18   ellipse(x, y, 60, 60);
19   fill(0);
20   ellipse(x+10, y, 30, 30);
21   fill(255);
22   ellipse(x+16, y-5, 6, 6);
23 }
24
```

- This time we called the eye method 6 times (sending in different parameters).

Topics List

- Recap of method terminology:
  - Return type
  - Method names
  - Parameter list

- Writing your own methods:
  - With no parameters
  - With parameters
  - That return data

Writing methods that return data

- Write a method called **timesTwo.**
- This method should take in one Integer parameter.
- The method should multiply this Integer by 2 and return it back to where the **timesTwo** method was called from.
- The returned value should be printed to the console.

Example 5.7

```
int value = 30;

void setup()
{
    int result = timesTwo(value);
    println(result);
}
```

```
int timesTwo(int val)
{
    val = val * 2;
    return val;
}
```


//source: https://processing.org/reference/return.html

Questions?

Produced by:

Dr. Siobhán Drohan
Mairead Meagher
Sinéad Walsh

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Department of Computing and Mathematics
http://www.wit.ie/