

# Conditional Events

## Conditional Statements and Boolean Expressions

---

Produced      Dr. Siobhán Drohan  
by:            Mr. Colm Dunphy  
                Mr. Diarmuid O'Connor



Waterford Institute *of* Technology  
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

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

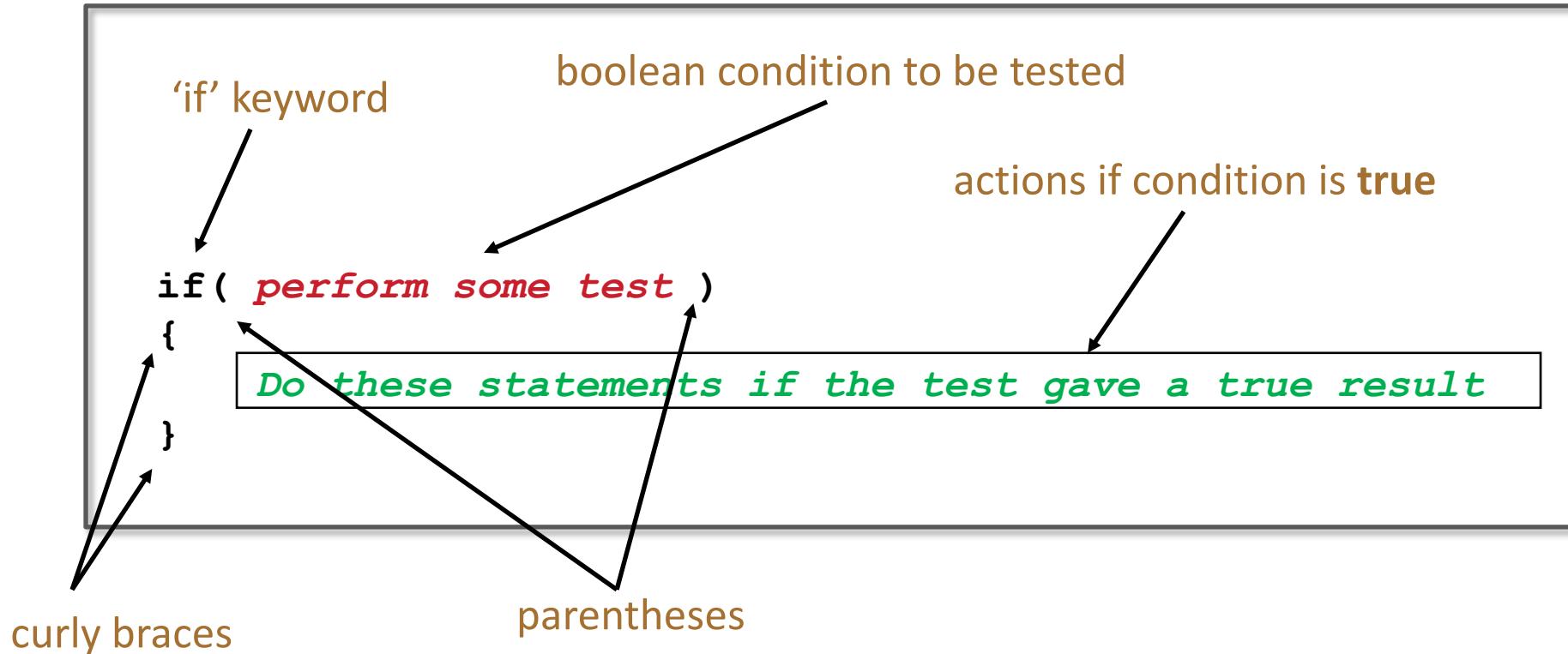
# Topics list

---

1. Conditional Statements
2. Boolean Conditions and Relational Operators
3. Logical Operators

# Conditional Statement Syntax (1)

---



# Conditional Statement Syntax (2)

---

```
if(perform some test) {  
    Do these statements if the test gave a true result  
}  
else {  
    Do these statements if the test gave a false result  
}
```

Diagram illustrating the syntax of an if-else conditional statement:

- 'if' keyword: Points to the 'if' keyword at the start of the first block.
- boolean condition to be tested: Points to the expression '(*perform some test*)' within the parentheses of the 'if' statement.
- actions if condition is true: Points to the code block following the opening brace '{' of the 'if' statement.
- 'else' keyword: Points to the 'else' keyword at the start of the second block.
- actions if condition is false: Points to the code block following the opening brace '{' of the 'else' statement.

# Conditional Statement Syntax (3)

---

```
if(condition1...perform some test)
{
    Do these statements if condition1 gave a true result
}
else if(condition2...perform some test)
{
    Do these statements if condition1 gave a false
result and condition2 gave a true result
}
else
{
    Do these statements if both condition1 and
condition2 gave a false result
}
```



# Topics list

---

1. Conditional Statements
2. Boolean Conditions and Relational Operators
3. Logical Operators

# Boolean conditions

---

- A boolean condition is an expression that evaluates to either **true** or **false** e.g.

`mouseX < 50`

- An if statement evaluates a **boolean condition** and its result will determine which portion of the if statement is executed.

# Boolean conditions

---

```
// Do these statements before.  
  
if (boolean condition)  
{  
    // Perform this clause if the  
    // condition is true.  
}  
  
// Do these statements after.
```

# Java Relational Operators

---

Operator	Use	Returns true if...
>	$op1 > op2$	op1 is <b>greater</b> than op2
$\geq$	$op1 \geq op2$	op1 is <b>greater than or equal</b> to op2
<	$op1 < op2$	op1 is <b>less</b> than op2
$\leq$	$op1 \leq op2$	op1 is <b>less than or equal</b> to op2
$\equiv$	$op1 \equiv op2$	op1 and op2 are <b>equal</b>
$\neq$	$op1 \neq op2$	op1 and op2 are <b>not equal</b>

**BEWARE** = is an assignment operator. It doesn't test for equality. Use == to test for equality

Source: [http://www.freejavaguide.com/relational\\_operators.htm](http://www.freejavaguide.com/relational_operators.htm)

# Some notes on the if statement

---

- An if statement **IS** a **statement**; it is only executed once.
- When your if statement only has one statement inside it, you do not need to use the curly braces.
  - But I would advise to use them to avoid errors when updating programs later.
- For example, both of these are the same:

```
if (mouseX < 50)
{
    rect(0, 0, 50, 100);
}
```

```
if (mouseX < 50)
    rect(0, 0, 50, 100);
```

# Some notes on the if statement

---

- The semi-colon (;) is a **statement terminator**.

```
if (mouseX < 50) ←  
{  
    rect(0, 0, 50, 100);  
}
```



Your if  
statement  
does not  
need  
a statement  
terminator.

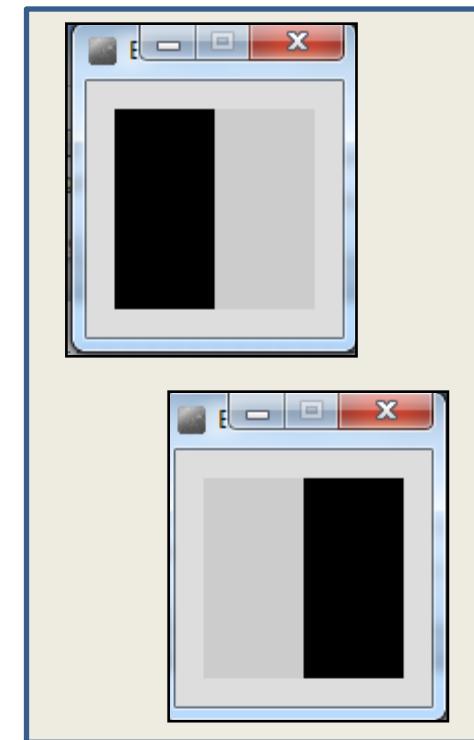
# Conditional Example 2.1

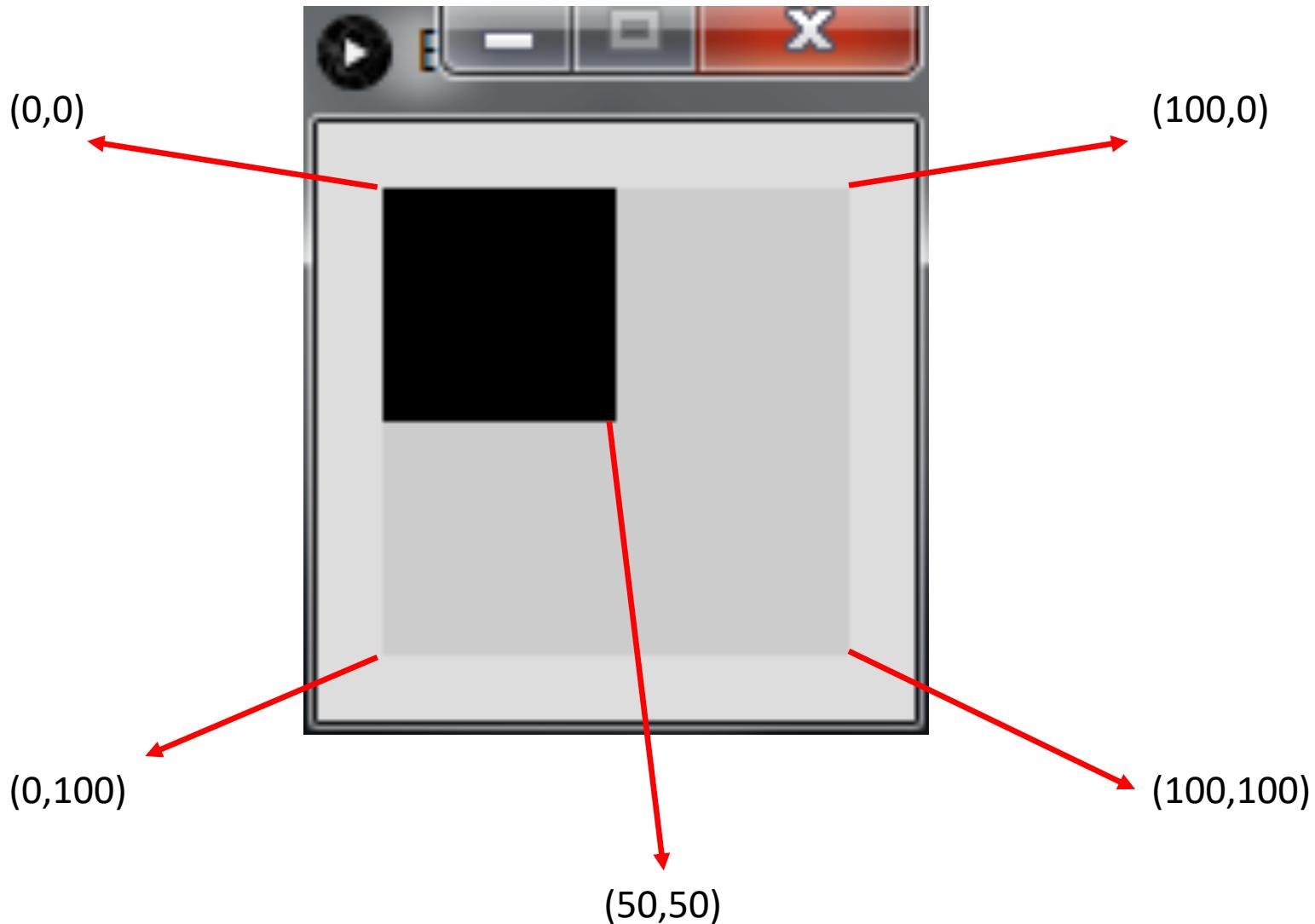
---

Functionality:

If the **x-coordinate of the mouse pointer** is on the:

- **left** half of the display window, draw a rectangle on the left hand side.
- **right** half of the display window, draw a rectangle on the right hand side.





# Conditional Example 2.1 - code

The image shows the Processing IDE interface with a sketch titled "Example\_2\_1". The code is as follows:

```
//Reas, C. & Fry, B. (2014) Processing - A Programming Introduction
void setup() {
    size(100, 100);
    noStroke();
    fill(0);
}

void draw() {
    background(204);
    if (mouseX < 50) {
        rect(0, 0, 50, 100); // Left
    } else {
        rect(50, 0, 50, 100); // Right
    }
}
```

Two output windows are shown on the right, each displaying a 100x100 pixel square divided into two 50x100 pixel quadrants. The top window shows the left half (0-50) in black and the right half (50-100) in light gray. The bottom window shows the left half (0-50) in light gray and the right half (50-100) in black. Red arrows point from the corresponding code blocks in the draw() function to these outputs.

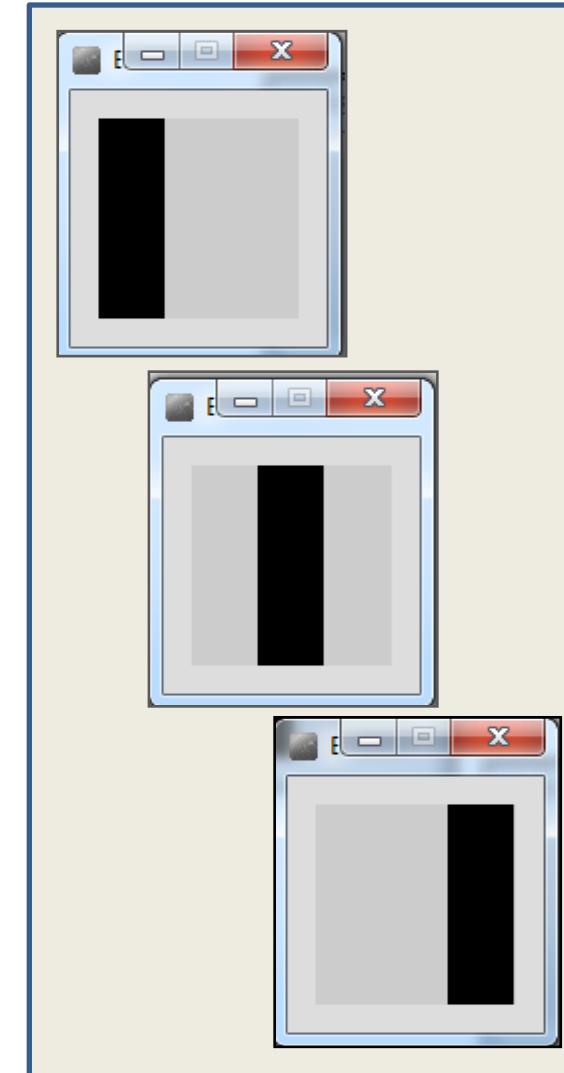
# Conditional Example 2.2

---

Functionality:

If the **x-coordinate of the mouse pointer** is on the:

- **left third** of the display window, draw a rectangle on the left third of the window.
- **middle third** of the display window, draw a rectangle on the middle third of the window.
- **right third** of the display window, draw a rectangle on the right third of the window.



# Conditional Example 2.2 - code

The image shows the Processing IDE with the sketch titled "Example\_2\_2". The code uses a conditional statement to draw different rectangles based on the mouse position:

```
//Reas, C. & Fry, B. (2014) Processing - A Programming Introduction
void setup() {
    size(100, 100);
    noStroke();
    fill(0);
}

void draw() {
    background(204);
    if (mouseX < 33) {
        rect(0, 0, 33, 100); // Left
    } else if (mouseX < 66) {
        rect(33, 0, 33, 100); // Middle
    } else {
        rect(66, 0, 33, 100); // Right
    }
}
```

Three separate windows show the resulting output for different mouse positions:

- Top window: Mouse at approximately (16, 50). A black rectangle is drawn from (0, 0) to (33, 100).
- Middle window: Mouse at approximately (50, 50). A black rectangle is drawn from (33, 0) to (66, 100).
- Bottom window: Mouse at approximately (83, 50). A black rectangle is drawn from (66, 0) to (100, 100).

# Topics list

---

1. Conditional Statements
2. Boolean Conditions and Relational Operators
3. Logical Operators

# Logical operators

---

- Logic operators operate on **boolean** values.
- They produce a new **boolean** value as a result.
- The ones that we will use, so far, are:

**&&**            (and)

**||**              (or)

**!**              (not)

# Logical operators - AND

---

a `&&` b

- This evaluates to `true` if both **a** and **b** are true.
- It is `false` in all other cases.

a	b	a && b
0	0	0
0	1	0
1	0	0
1	1	1

# Logical operators - OR

---

**a || b**

- This evaluates to **true** if either **a** or **b** or both are true, and **false** if they are both false.

<b>a</b>	<b>b</b>	<b>a    b</b>
0	0	0
0	1	1
1	0	1
1	1	1

# Logical operators - NOT

---

!a

- This evaluates to **true** if **a** is false, and **false** if **a** is true.

a	!a
0	1
1	0

# Logical operators - summary

---

a `&&` b                    *(and)*

- This evaluates to `true` if both **a** and **b** are true.
- It is `false` in all other cases.

a `||` b                    *(or)*

- This evaluates to `true` if either **a** or **b** or both are true, and `false` if they are both false.

`!a`                            *(not)*

- This evaluates to `true` if **a** is false, and `false` if **a** is true.

# Logical operators - quiz

---

```
int a = 5;  
int b = 10;  
int c = 7;
```

What is the result of each of these **boolean** expressions:

Q1  $(a > b) \&\& (a < c)$

Q2  $(a < b) \mid\mid (c < a)$

Q3  $!(b < a) \&\& (c > b)$

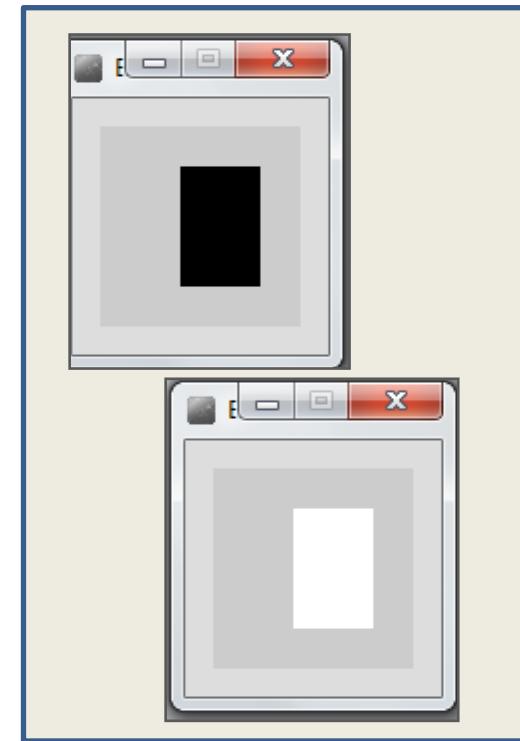
# Conditional Example 2.3

---

Functionality:

If the mouse pointer is:

- inside the rectangle coordinates,  
then fill the rectangle with white.
- otherwise, fill with black.



# Conditional Example 2.3 - code

The image shows the Processing 3.3.6 IDE with a sketch titled "Example\_2\_3". The code uses a conditional statement to draw either a white or black rectangle based on the mouse position.

```
//Reas, C. & Fry, B. (2014) Processing - A Programming Handbook
void setup() {
    size(100, 100);
    noStroke();
    fill(0);
}

void draw() {
    background(204);
    if ((mouseX > 40) && (mouseX < 80) &&
        (mouseY > 20) && (mouseY < 80)) {
        fill(255); //White
    } else {
        fill(0); //Black
    }
    rect(40, 20, 40, 60);
}
```

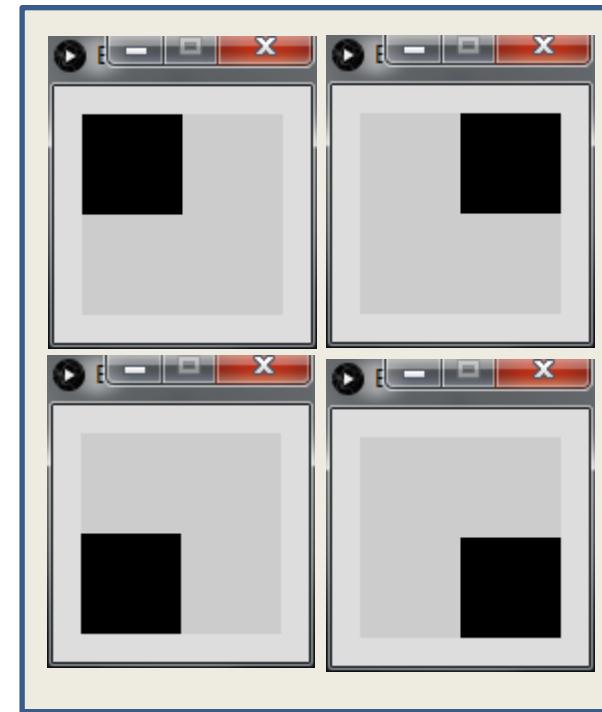
Two output windows are shown on the right, each containing a gray square frame. The top window contains a black rectangle, and the bottom window contains a white rectangle. Red arrows point from the corresponding code blocks (the "if" condition and the "else" block) to their respective outputs.

# Conditional Example 2.4

---

## Functionality:

- If the mouse pointer is in the upper-left quadrant of the display window, draw a black rectangle covering the upper-left quadrant of the window.
- Repeat this approach for upper-right, lower-left and lower-right quadrants.



## Conditional Example 2.4 - code

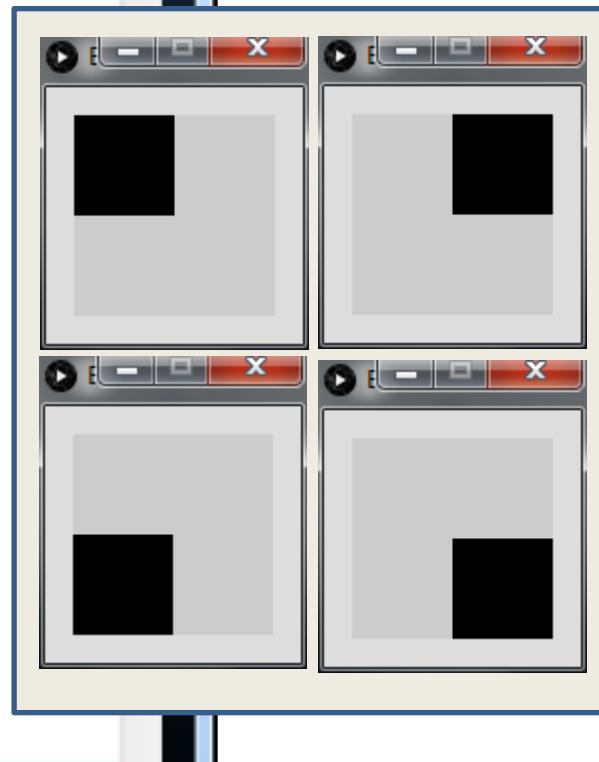
Example\_2\_4 | Processing 3.3.6

File Edit Sketch Debug Tools Help

Java

Example\_2\_4

```
1 //Reas, C. & Fry, B. (2014) Processing - A Programming Handbook
2
3 void setup() {
4     size(100, 100);
5     noStroke();
6     fill(0);
7 }
8
9 void draw() {
10    background(204);
11    if ((mouseX <= 50) && (mouseY <= 50)) {
12        rect(0, 0, 50, 50);      // Upper-left
13    }
14    else if ((mouseX <= 50) && (mouseY > 50)) {
15        rect(0, 50, 50, 50);    // Lower-left
16    }
17    else if ((mouseX > 50) && (mouseY <= 50)) {
18        rect(50, 0, 50, 50);    // Upper-right
19    }
20    else {
21        rect(50, 50, 50, 50);   // Lower-right
22    }
23 }
```



# Questions?

---



# References

---

- Reas, C. & Fry, B. (2014) Processing – A Programming Handbook for Visual Designers and Artists, 2<sup>nd</sup> Edition, MIT Press, London.