


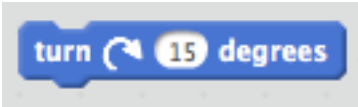

Scratch and Python Programming Constructs

This document maps programming constructs in Scratch 2 to equivalent (or near equivalent) constructs in Python 3. Where appropriate the constructs are also related to Logo and to spreadsheet features found in applications such as Excel. This is a quick guide not a complete explanation of constructs and features.

Contents

1. Motion	2
2. Pen	3
3. Control.....	4
If then block.....	4
If then else block.....	5
Repeat block/for loop	6
Forever block.....	6
Repeat until block.....	7
4. Looks (visual output to screen)	7
5. Data (variables).....	8
6. Sensing (input)	9
7. Operators	10



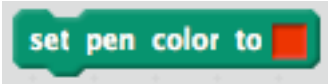
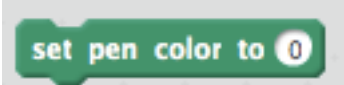
1. Motion

Scratch	Python (turtle module)	Logo
 <p>Move 10 steps forward. Use a negative number of steps to move backward.</p>	<pre>forward(10) fd(10)</pre> <p>forward and fd are pseudonyms. Use a negative distance to move backward. Or use the back, bk or backward functions</p>	<pre>forward 10 fd 10</pre> <p>Backward movement: back bk</p>
 <p>Turn right 15 degrees.</p>	<pre>right(15) rt(15)</pre>	<pre>right 15 rt 15</pre>
 <p>Turn left 15 degrees.</p>	<pre>left(15) lt(15)</pre>	<pre>left 15 lt 15</pre>

In the above the absolute values (e.g. 10 steps, 15 degrees) can be replaced by any valid number or by an appropriate variable. See Section 5 on Data for information about variables.

There are other motion blocks in Scratch and corresponding functions/commands in Python and Logo for finding and changing position etc.

2. Pen


Scratch	Python (turtle module)	Logo
 <p>Pen down to draw while moving.</p>	<pre>pendown () pd () down ()</pre>	<pre>pendown</pre>
 <p>Pen up to stop drawing.</p>	<pre>penup () pu () up ()</pre>	<pre>penup</pre>
  <p>Set the pen colour to red. Colours can be specified from a colour picker or by number – 0 for red, 70 for green, 130 for blue etc. (see Scratch help)</p>	<pre>color ('red')</pre> <p>RGB (red, green, blue) values can also be used to set the pen colour.</p>	<pre>setpencolor 4 setpc 4 setc 4</pre>

There are other pen blocks in Scratch and corresponding functions/commands in Python and Logo to control the pen, including fill colours for shapes.

The pen is up by default in Scratch and down by default in Python/Turtle.

3. Control

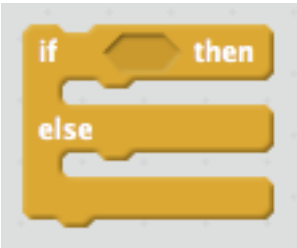
If then block

Scratch	 A yellow Scratch 'if then' block with 'if' on the left and 'then' on the right, with a notch on the top right for interlocking with other blocks.	Execute any blocks inside the if/then block if the boolean condition between "if" and "then" is true.
Python	<pre>if condition: # execute any statements # that are indented here</pre>	If boolean condition is true, execute the statements that are indented under the <code>if</code> statement.
Logo	<pre>if condition [...]</pre>	If boolean condition is true, execute the statements inside []
Excel	<code>IF(logical_test, value_if_true, value_if_false)</code>	This Excel function is equivalent to a variable assignment that depends on the result of a logical test (e.g. whether a boolean condition is true or not). Use VBA to execute statements that depend on the result of a logical test.

A boolean condition is a statement that can either be true or false, for example: "it is raining" or "it is Tuesday"

In Python, code that follows a # is a comment. The Python interpreter ignores anything that follows the # and that is on the same line as the #.


If then else block

Scratch		Execute any blocks inside the if/then branch of the block if the boolean condition between "if" and "then" is true. Otherwise, execute any blocks in the else branch of the "if then else" block.
Python	<pre>if condition: # execute statements # indented here # if condition is true else: # execute statements # indented here # if condition is not true</pre>	If condition is true, execute the statements indented under the if clause. Otherwise, execute the statements indented under the else clause.
Logo	<pre>ifelse condition [trueBlock] [falseBlock]</pre>	If condition is true, execute the statements inside the first [] - the trueBlock. Otherwise, execute the statements inside the second [] - the falseBlock.
Excel	IF(logical_test, value_if_true, value_if_false)	See "if then" block

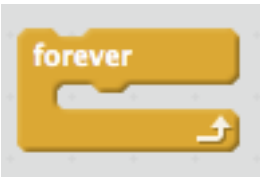
Python can have intermediate `elif` clauses between `if` and `else`. For example:

```
if x > 0:
    print('x is greater than 0')
elif x == 0:
    print('x is 0')
else:
    print('x is less than 0')
```


Repeat block/for loop

Scratch		Repeat "move 10 steps" 10 times. Any blocks inside a repeat block are repeated the specified number of times.
Python	<pre>for counter in range(10): forward(10)</pre>	Move forward 10, 10 times. Any statements indented under a <code>for</code> statement are repeated the specified number of times. In this example, the <code>counter</code> variable is given each value in turn from the range 0 to 9 inclusive.
Logo	<pre>repeat 10 [forward 10]</pre>	Move forward 10, 10 times. Anything in <code>[]</code> is repeated the specified number of times.
Excel	Use VBA to provide the equivalent of a for/repeat loop in Excel	

Forever block

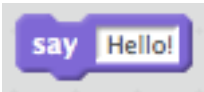
Scratch		Repeat any blocks inside the "forever block" forever.
Python	<pre>while True: # repeat whatever is # indented under the while # statement forever</pre>	This loop repeats as long as the boolean condition that is part of the <code>while</code> statement is true. The boolean value <code>True</code> is always true, therefore the loop repeats forever.
Logo	<pre>forever [...]</pre>	Repeat the statements inside <code>[]</code> forever.

Repeat until block

Scratch		Repeat any blocks inside the "repeat until" block until some condition is true (the condition is placed in the 5-sided box next to "repeat until")
Python	<pre>while not condition: # repeat whatever is # indented under the while # statement until the # condition is true</pre>	Repeat any statements indented beneath the while statement until the boolean condition is true, which will make the expression <code>not condition</code> false.
Logo	<pre>until condition [...]</pre>	Repeat any statements inside [] until condition is true.

The loops repeat until a termination condition is met. A `while` loop repeats as long as a condition is true. Therefore, for it to repeat until some condition is true we use its negation (`not condition`) as part of the `while` statement.



4. Looks (visual output to screen)

Scratch	Python	Logo
 <p>Sprite says 'Hello!' – outputs 'Hello!' to the screen.</p>	<pre>print('Hello!')</pre>	<pre>print word "Hello!"</pre>

In the above 'Hello!' can be replaced by any text (also called a string, as in string of characters) or an appropriate variable.

There are other looks blocks in Scratch to output to the screen (or Scratch stage). Python uses `print` to output to the console or standard output device (e.g. screen).



5. Data (variables)

Scratch	Python	Excel
 <p>The name "a1" is just an example to illustrate the relationship to Excel.</p>	<code>a1 = 0</code>	Enter 0 into cell A1
 <p>Add 1 to the variable a1.</p>	<code>a1 = a1 + 1</code>	Increment the value in cell A1 by 1.

In Excel we have cells named A1, B1, C1 etc. A cell can hold a value. Values in cells can be fixed or they can change as the result of a calculation. Cells can hold values of different types (text, integers, decimal numbers, dates etc.).

A variable is a named place in computer memory (basically a named cell). A variable can have a fixed value for the whole of a program execution or it may change as the result of some calculation. A variable can hold values of different types (strings, integers, decimal numbers, dates etc.).

6. Sensing (input)

Scratch	Python
 <p>This block outputs the string 'What's your name?' and waits for the user to enter an answer. The result is assigned to the built-in Scratch variable:</p>  <p>that stores the answer to the most recent question.</p>	<pre>answer = input('What's your name? ')</pre> <p>This statement uses the <code>input</code> function to print the string <code>What's your name?</code> to screen as a prompt for user input. Any string of characters the user enters is assigned to the variable <code>answer</code>. This is not a built-in variable. For example, we could have assigned the result of the <code>input</code> function to a variable called <code>your_name</code>.</p>

In Python the result of the `input` function is a string of characters. If we want to treat the answer to the user prompt as a number then we must convert the string of characters to a number. For example, the following Python code:














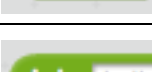

```
answer = input('What is your age? ')\nage = int(answer)
```

asks a user their age, and stores the answer in a variable called `answer`. The answer is then converted to an integer `age` using the `int` function.

Scratch does the preceding conversion between types automatically depending on the context in which the answer is used.

Logo has similar facilities and converts between types in a similar way to Scratch.

7. Operators

Operation	Scratch	Python
add		$x + y$
subtract		$x - y$
multiply		$x * y$
divide		x / y
modulus		$x \% y$
round		<code>round()</code>
boolean or		$x \text{ or } y$ True if either x or y is true
boolean and		$x \text{ and } y$ True if both x and y are true
boolean not		<code>not x</code> True if x is not true
equals comparison		$x == y$ True if x is equal to y Note: use of <code>==</code> not <code>=</code>
greater than comparison		$x > y$ True if x is greater than y
less than comparison		$x < y$ True if x is less than y
string concatenation		<code>'hello ' + 'world'</code> Excel equivalent: <code>"hello " & "world"</code>
length of string		<code>len('world')</code>
character in string		<code>'world'[0]</code> The index position of a character in a string picks out the character. The 0 th position is the first character.

There are Excel equivalents to all of the preceding operators (including string concatenation and conversion of numbers to strings/text before concatenation).

Logo also has equivalent operators.

Scratch, Python, and Excel also have various mathematical functions (e.g. to find the square root of a number or to generate a random number).