CoderDojo Athenry

HaCkeRS

Notes by Michael Madden, CoderDojo Athenry, 2020

Reminder: Programming Languages

Input, Output & Store Data

- E.g. text, numbers, LEDs, buttons

Operate on Data

- E.g. add numbers, change text

Loops

- Repeat commands several times

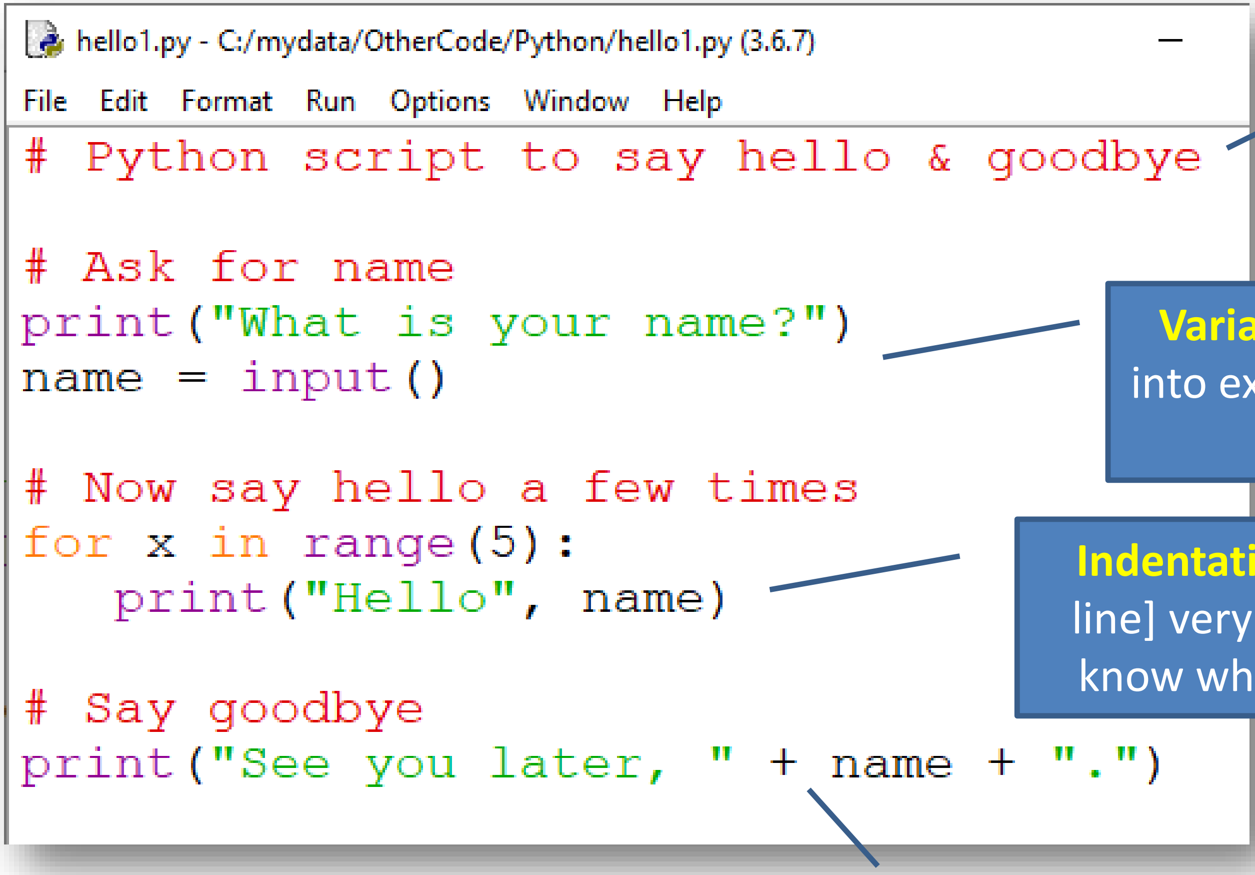
Decisions

- Do something IF something else is true

Programming in Python

- Python is a cross-platform programming language that is one of the world's most popular
- Designed for writing programs to be readable and relatively easy to modify
- Widely used in industry, in areas such as computer vision, AI, and hardware control
- Huge set of libraries and tools for all sorts of tasks
- Pre-installed on the Raspberry Pi with IDLE interpreter; can also install on PCs
- Scripting language suited for interactive and experimental programming: easy to try new things out

Python Example


hello1.py - C:/mydata/OtherCode/Python/hello1.py (3.6.7)
File Edit Format Run Options Window Help

```python
# Python script to say hello & goodbye

# Ask for name
print("What is your name?")
name = input()

# Now say hello a few times
for x in range(5):
    print("Hello", name)

# Say goodbye
print("See you later, " + name + ".")
```

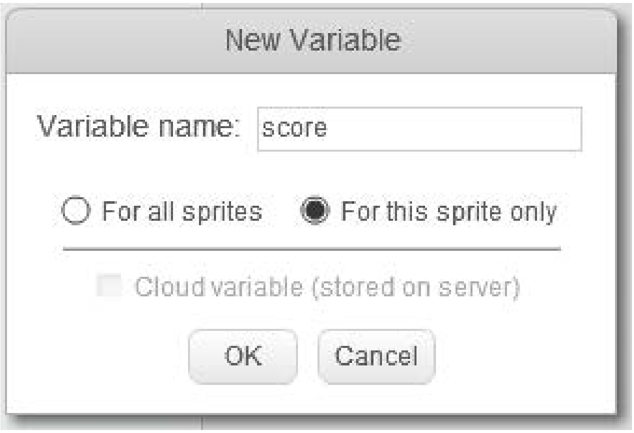
Comments start with #

Variables (name, x) come into existence when you use them

Indentation [space at start of line] very important: how you know what is in a loop or not

Variables are **dynamically typed:** Python figures out what kind of data it is and treats it accordingly

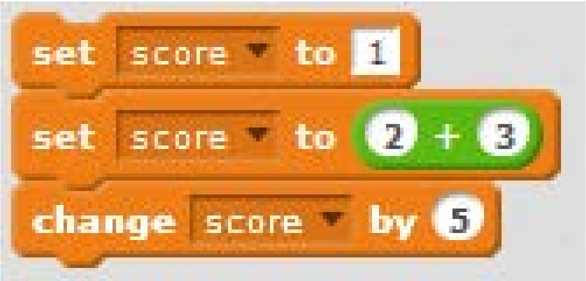
Scratch vs. Python: Variables & Operators



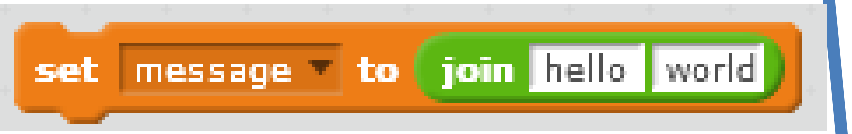
Variable name: score

○ For all sprites ● For this sprite only

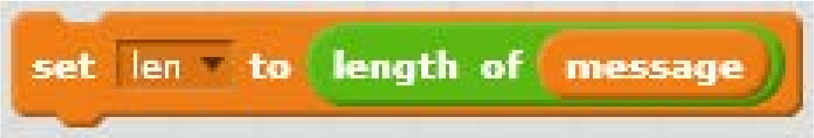
□ Cloud variable (stored on server)



```
set score to 1
set score to 2 + 3
change score by 5
```



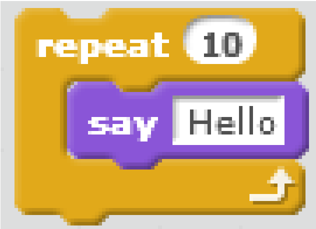
```
set message to join hello world
```



```
set len to length of message
```

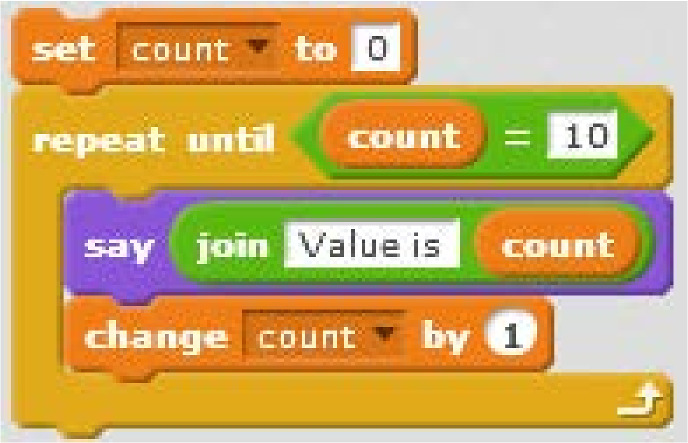
```
score = 1 # Variable created
          # on first use


score = 1
score = 2 + 3
score = score + 5

message = "hello " + "world"

mlen = len(message);
```

Scratch vs. Python: Loops



```
repeat 10
    say Hello
```

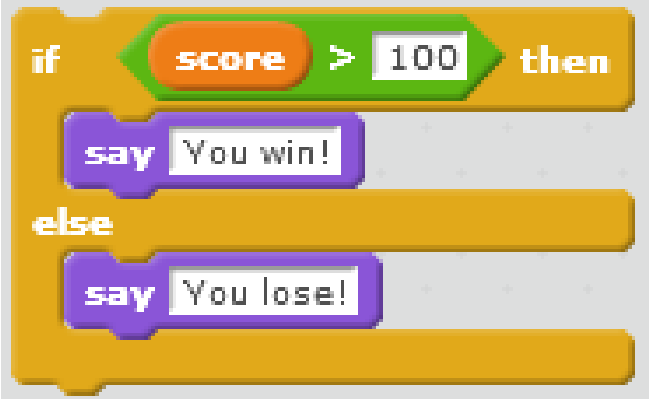
```
for i in range(10):
    print("Hello")
```



```
set count to 0
repeat until count = 10
    say join Value is count
    change count by 1
```

```
count = 0
while count < 10:
    print("Value is", count)
    count = count + 1
```

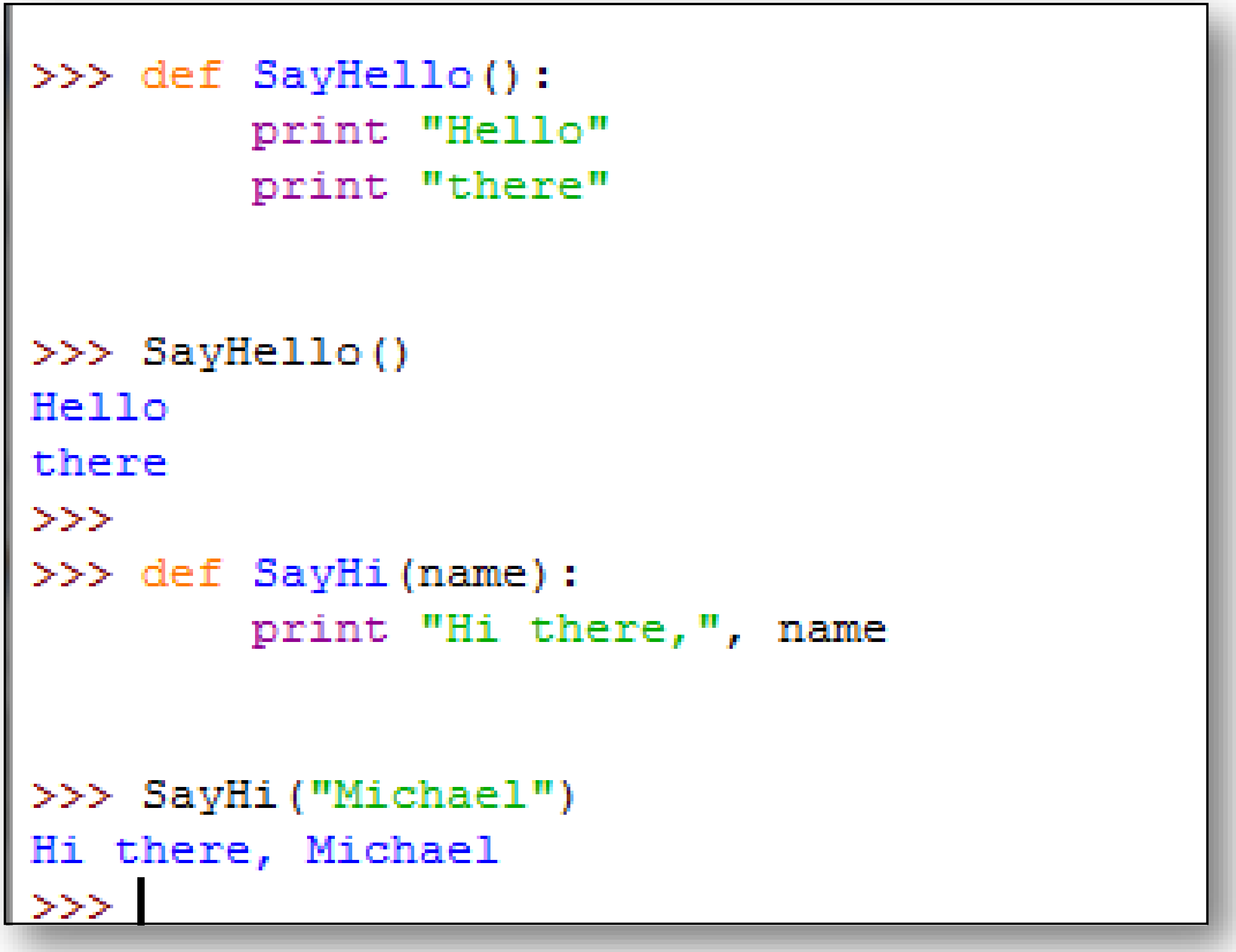
Scratch vs. Python: Decisions



```
if score > 100 then
    say You win!
else
    say You lose!
```

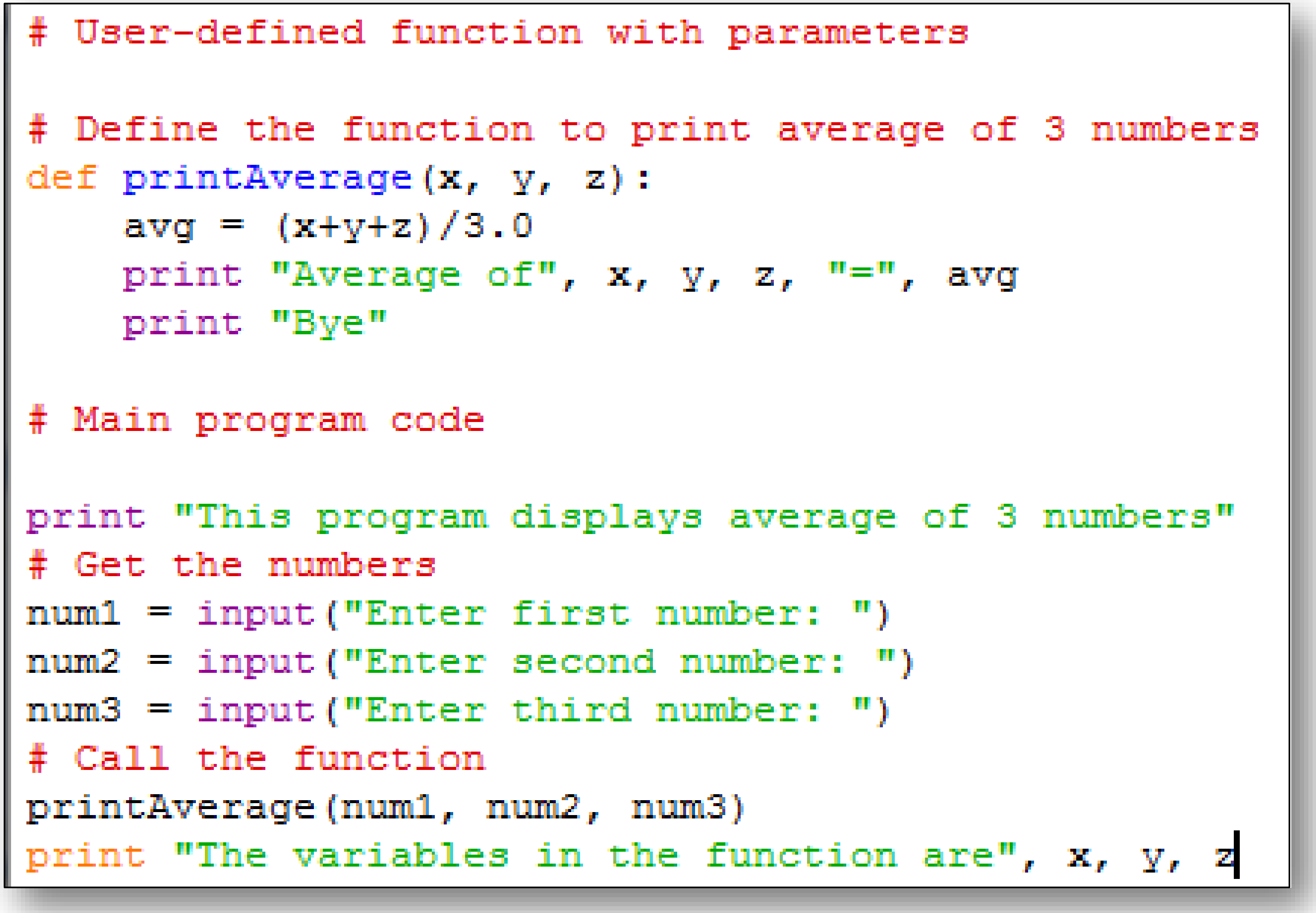
```
if score > 100:
    print("You win!")
else:
    print("You lose!")
```

Python: Define & Call Functions



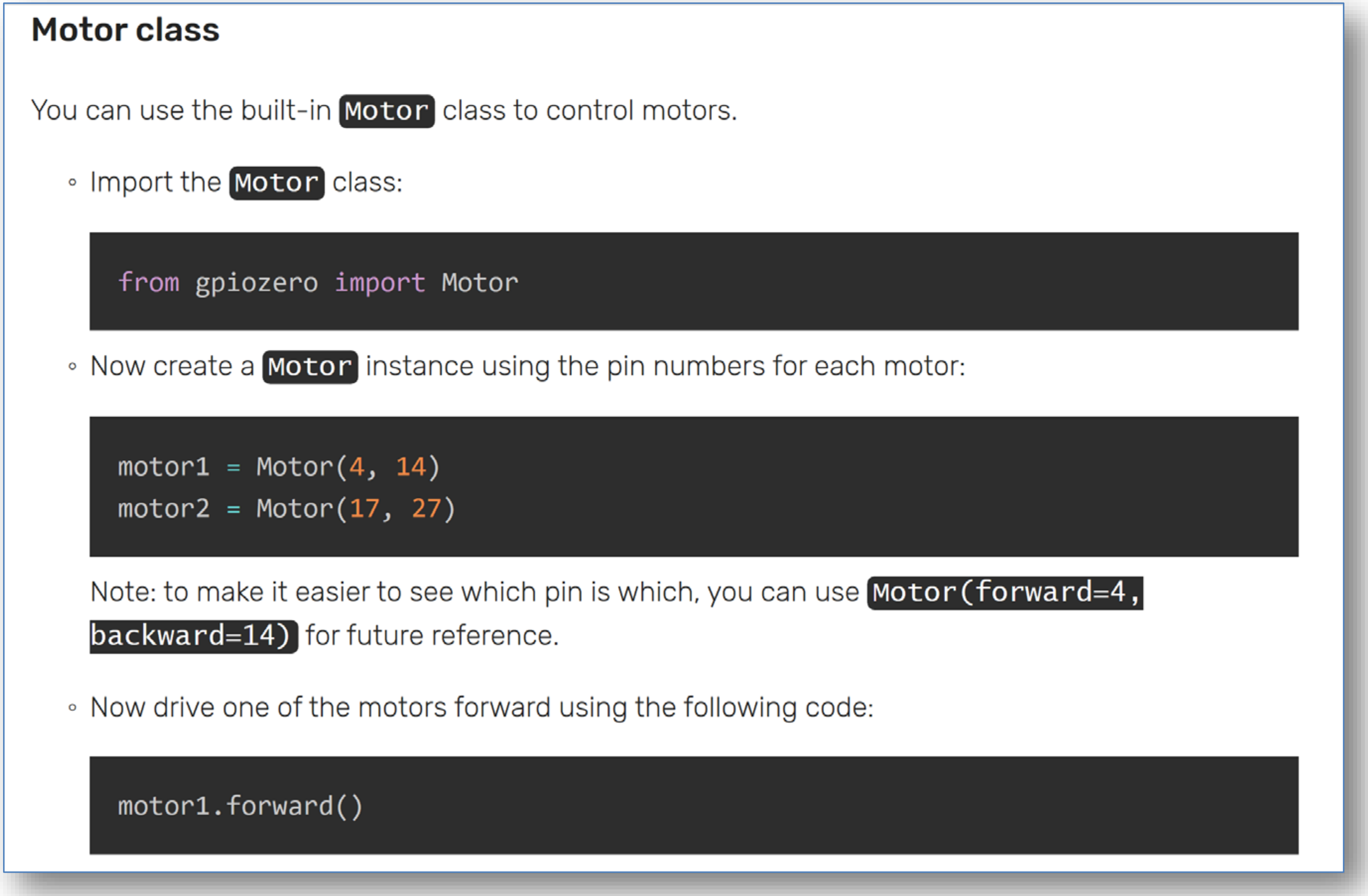
```
>>> def SayHello():
        print "Hello"
        print "there"


>>> SayHello()
Hello
there
>>>
>>> def SayHi(name):
        print "Hi there,", name


>>> SayHi("Michael")
Hi there, Michael
>>> 
```

Python: Define & Call Functions



```
# User-defined function with parameters

# Define the function to print average of 3 numbers
def printAverage(x, y, z):
    avg = (x+y+z)/3.0
    print "Average of", x, y, z, "=", avg
    print "Bye"

# Main program code

print "This program displays average of 3 numbers"
# Get the numbers
num1 = input("Enter first number: ")
num2 = input("Enter second number: ")
num3 = input("Enter third number: ")
# Call the function
printAverage(num1, num2, num3)
print "The variables in the function are", x, y, z
```

Lots of Great Libraries for RPi



Motor class

You can use the built-in `Motor` class to control motors.

- Import the `Motor` class:

```
from gpiozero import Motor
```

- Now create a `Motor` instance using the pin numbers for each motor:

```
motor1 = Motor(4, 14)
motor2 = Motor(17, 27)
```

Note: to make it easier to see which pin is which, you can use `Motor(forward=4, backward=14)` for future reference.

- Now drive one of the motors forward using the following code:

```
motor1.forward()
```