Let’s Learn Python!

Young Coders at PyCon 2015

What is programming?

- A **problem** to solve
- A **solution** to the problem
- The solution translated into a language the computer can understand

Math

```
>>> 1 + 2
>>> 12 - 3
>>> 9 + 5 - 15

>>> 6 * 5
>>> 6 / 2
>>> 10 * 5 * 3

>>> 8 / 5
>>> 20 / 7
>>> 10 / 3
```

```
>>> 10/3
>>> 10/2

>>> 10/3.0
>>> 10.0/2
```

Rule: If you want Python to answer in floats, you must talk to it in floats.

```
>>> 5 < 4 + 3
>>> 12 + 1 >= 12
>>> 16 * 2 == 32
>>> 16 != 16
>>> 5 >= 6
```

==	Equal to
!=	Not equal to
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to

Strings

```
>>> “garlic breath”
>>> “Hello!”

>>> apple

>>> “apple”
>>> “What’s for lunch?”
>>> “3 + 5”

>>> “Hi” + “there!”

>>> “HAHA” * 250
```

Rule: A string must be in quotes

String operators:

concatenation: +
multiplication: *

```
>>> “H” + “e” + “l” + “l” + “o”

>>> print “Hello”
>>> print “Hello”[0]
>>> print “Hello”[4]

>>> print “Hey, Bob!”[6]
>>> print “Hey, Bob!”[6 - 1]

>>> print “Hey, Bob!”[4]
```

Strings are made up of **characters.** Spaces in the string also count as characters.

Each character has a position called an *index*

In Python, indexes start at **0**

Variables

```
>>> 12 * 12
>>> donuts = 12 * 12
>>> donuts

>>> color = “yellow”
>>> color
>>> color = “red”
>>> color
>>> color = “fish”
>>> color = 12

>>> donuts = 12 * 12
>>> fishes = 3
>>> fishes + donuts
```

Calculate once, keep the result to use later

Keep the same name, change the value

```
>>> color = “yellow”
>>> day = “Monday”
>>> color + day
>>> color * fishes
>>> color + day * fishes

>>> fruit = “watermelon”
>>> print fruit[2]

>>> number = 3
>>> print fruit[number-2]
```

Assigning values vs making comparisons:

```
>>> fruit = “watermelon”
>>> 5 = 6

>>> fruit == “watermelon”
>>> 5 == 6
```

Booleans

```
>>> 1 == 1
>>> 15 < 5

>>> True
>>> False

>>> true
>>> false
>>> type(True)
>>> type(“True”)

>>> 1==1 or 2==2
>>> 1==1 or 2!=2
>>> 1==2 or 2==3

>>> 1==1 and 2==2
>>> 1==1 and 2==3
>>> 1==2 and 2==3

>>> 1==1
>>> not 1==1
>>> not True
```

A Boolean value can be: `True` or `False`.

```
>>> True and True
>>> True and False
>>> False and False

>>> True or True
>>> False or True
>>> False or False

>>> not True and True
>>> not True or True
```

```
>>> True and True

>>> False and True

>>> 1 == 1 and 2 == 1

>>>"test" == "test"

>>> 1 == 1 or 2 != 1

>>> True and 1 == 1

>>> False and 0 != 0

>>> True or 1 == 1

>>>"test" == "testing"

>>> 1 != 0 and 2 == 1
```

Lists

```
>>> fruit = [“apple”, “banana”, “grape”]
>>> numbers = [3, 17, -4, 8.8, 1]
>>> type(fruit)
>>> type(numbers)
```

```
>>> print “apple”[0]
>>> fruit
>>> print fruit[0]
```

```
>>> colors = [‘red’, ‘orange’, ‘purple’]
>>> print colors[1]
```

Errors

```
>>> “friend” * 5
>>> “friend” + 5
>>> “friend“ + “5”
>>> print “friend“, 5
```

Data types

```
>>> type(“Hi!”)
```

if Statements

```
>>> name = "Katie"
>>> if name == "Katie":
        print "Hi Katie!"


>>> if name == "Katie":
        print "Hi Katie!"
    else:
        print “Impostor!”


>>> if name == "Katie":
        print "Hi Katie!"
   elif name == "Barbara":
        print "Hi Barbara!"
   else:
        print "Who are you?"
```

```
>>> color = "blue"
>>> if color == "yellow":
         print "Yay!"
   elif color == "purple":
         print "Try again!"
   else:
        print "We want yellow!"
```

Loops

Counting loops repeat a number of times - they keep going until they get to the end of a count.

```
>>> for mynum in [1, 2, 3, 4, 5]:
        print "Hello", mynum
```

The for keyword is used to create this kind of loop, so it is usually just called a for loop.

Conditional loops repeat until something happens (or as long as some condition is True).

```
>>> count = 0
>>> while (count < 4):
        print 'The count is:', count
        count = count + 1
```

The while keyword is used to create this kind of loop, so it is usually just called a while loop.

Functions

Functions are a way to *group* instructions.

```
>>> def say_hello(myname):
        print ‘Hello’, myname
>>> say_hello("Katie")
>>> say_hello("Barbara")


>>> def double(number):
        print number * 2
>>> double(14)
>>> double("hello")

>>> def multiply(num1, num2):
        print num1 * num2
>>> multiply(4, 5)
>>> multiply("hello", 5)
```

```
>>> def double(number):
        print number * 2
>>> double(12)
>>> new_number = double(12)
>>> new_number


>>> def double(number):
        return number * 2
>>> double(12)
>>> new_number = double(12)
>>> new_number
```

`def`	keyword	We use this to let Python know that we’re defining a function.
`myname`	parameter (& variable)	We use this to represent values in the function.
`print ...`	body	This is where we say what the function *does*.

Input

```
>>> def hello(myname):
      print "Hello", myname
>>> hello_there("Katie")

>>> def hello_there():
      print “Type your name:”
      name = raw_input()
      print “Hi”, name, “how are you?”
>>> hello_there()

>>> def hi_there():
      name = raw_input(“Type your name: ”)
      print “Hi”, name, “how are you?”
>>> hello_there()
```

Modules

```
>>> import random
>>> print random.randint(1,100)

>>> import time
>>> time.tzname

>>> import calendar
>>> calendar.prmonth(2014, 4)

>>> import os
>>> for file in os.listdir(“/home/pi”):
        print file

>>> import urllib
>>> myurl = urllib.urlopen('http://www.python.org')
>>> print myurl.read()
```

Games!

```
secret_number = 7

guess = input("What number am I thinking of? ")

if secret_number == guess:
    print "Yay! You got it."
else:
    print "No, that's not it."
```

```
from random import randint

secret_number = randint(1, 10)

while True:
    guess = input("What number am I thinking of? ")

    if secret_number == guess:
        print "Yay! You got it."
        break
    else:
        print "No, that's not it."
```

Games!

```
from random import randint

secret_number = randint(1, 10)

while True:
    guess = input("What number am I thinking of? ")

    if secret_number == guess:
        print "Yay! You got it."
        break
    elif secret_number > guess:
        print "No, that's too low."
    else:
        print "No, that's too high."
```

Minecraft

```
>>> from mcpi import minecraft
>>> mc = minecraft.Minecraft.create()
>>> mc.postToChat("Hello world”)
>>> pos = mc.player.getPos()
>>> print pos.x
>>> x, y, z = mc.player.getPos()
>>> mc.player.setPos(x, y+100, z)
>>> mc.setBlock(x+1, y, z, 1)
```

Minecraft

```
>>> from mcpi import block

>>> dirt = block.DIRT.id
>>> mc.setBlock(x, y, z, dirt)

>>> stone = block.STONE.id
>>> mc.setBlocks(x+1, y+1, z+1, x+11, y+11, z+11, stone)

>>> tnt = 46
>>> mc.setBlocks(x+1, y+1, z+1, x+11, y+11, z+11, tnt)
>>> mc.setBlocks(x+1, y+1, z+1, x+11, y+11, z+11, tnt, 1)
```

Raspberry Pi

Help setting up your new computer:

http://www.raspberrypi.org/quick-start-guide

Minecraft on your new Raspberry Pi:

https://www.raspberrypi.org/learning/getting-started-with-minecraft-pi/worksheet/

http://www.stuffaboutcode.com/p/minecraft-api-reference.html