Data Programming 101

Sai Sivam

Software Research Industrial Life Analytics Labs
SRILA Labs, Srirangam
srila.labs@gmail.com

To Guru and Gauranga

Knowledge Computation

Knowledge Computation (KC) is all about identifying and extracting hidden knowledge present in the vast amounts of data that we find ourselves surrounded by.

By learning data programming and practicing these techniques, you can then begin walking on the long road to becoming a proficient Knowledge Computational software engineer.

Welcome to KC using Data Programming 101!

Installation

Establish a connection to the Internet

```
Install Python 3.4                      # from python.org
Install Git Bash i.e. msysgit           # from msysgit.github.io
```

Then, run the the following commands in a Git Bash command shell

```
cd ~/Desktop
mkdir SRILA
cd SRILA
git clone https://github.com/srilalabs/DataProgramming.git
cd DataProgramming
ls
cd DataProgramming
cat README
```

Integers

```
count = 108
count
print(count)
```

Strings

```
name = "SRILA Labs"
name
print(name)
```

Math

```
54 + 54
16 * 108
3/2
```

```
int(3/2)
3//2
3//2.0
```

Integers to Real Numbers (Floating Point)

```
i = 1510;     print(i)
f = float(i); print(f)
f = 1947.1;   print(f)
i = int(f);   print(i)
```

Remainder

```
4 % 2
7.5 % 2
10 % 3.3
```

Power

```
2**2                                  # power, note the double  *
2**10
2**32
2**64
```

Arithmetic Evaluation

```
3*(1+4)                               # brackets, / , *, + , -
3/(1+4)
```

Math Utils

```
gpa = 8.5
import math
math.floor(gpa)
math.ceil(gpa)
```

Convenient Operators

```
count = 1                    # do not use ++count
count                        # it is not what you think
count += 1
count
count *= 10
count
```

Strings

```
s = 'software'
r = 'research'
i = 'industrial'
l = 'life'
a = 'analytics'

s + r + i + l + a
```

```
print(s, r, i, l, a)
print(s, r, i, l, a, "labs")
srila = s + r + i + l + a
s + ' ' + r + ' ' + i + ' ' + l + ' ' + a + ' ' + 'labs'

name = "srila\nlabs"
name
print(name)

name = "srila\tlabs"
name
print(name)

if ("sri" in "srila"):
    print("yes")

"srila" in name
'srila' in name
```

```
s + 2014
s + str(2014)
str(2014)
str(1)
str(5*10)

"SRILA".lower()
name = "srila"
name.upper()
name = "Software Research Industrial Life Analytics Labs"
name.split()
name.lower().split()
namelist = name.split()
```

Output Formatting

```
marks = 55
str(marks)
```

```
print ("i got", marks)
print ("i got " + str(marks))
print ("i got %d " % marks)
print ("i got %f " % marks)
print ("i got %.2f " % marks)
```

Arrays (also called Lists)

```
[ "Gopala Bhatta", "Srirangam", 184, 9427357762, 85 ]
student = [ "Gopala Bhatta", "Srirangam", 184, 9427357762, 85 ]
student

print(student)
student.append("SRILA Labs")
student
print(student)

student.insert(1, 'Male')
student
```

```
student.pop()
student
student.append('Radien Software')
student

student.pop()
student

student.append('Orca Labs')
student

name = "srila"
len(name)
len("srila")
len(" srila labs ")

student
len(student)
```

```
capitals = 'SRILA'
list(capitals)

string = 'SRILA'
anotherlist = list(string)
anotherlist
```

Substring Matching

```
"read" in "are you ready to code"
"nectar" in "nectarine"
"of" in "office"
"devotion" in "we should be devotional in coding"

student
'Srirangam' in student
'Gopala' in student
```

```
adjustable = [ 'Gopala Bhatta Goswami', 'Srirangam' ]
adjustable.pop(1)
adjustable
adjustable.append('Vrindavan')
```

Fixed Lists

Note: Fixed lists are called tuples. They cannot be modified. Tuples () *are fixed. Lists are denoted with box brackets* []. *You may remember it as tuPles. The P in tuples needs parantheses* ().

```
fixed = ( 'SAI SIVAM', 'IIT Madras' )       # I studied in this college
fixed.pop(1)                                # That cannot be changed
fixed                                       # Note that nothing changed
fixed.append('IIM Trichy')                  # IIM Trichy cannot be added
                                            # to my list of colleges
                                            # but if I change my name to Jai
fixed = ( 'JAI NARASIMHA', 'DAV', 'IITM, 'IISc' )
```

```
# from Sai, which I am
# planning to, still my schooling
# history remains
# so you can reassign
# but not modify
```

Dictionaries

Note: dictionaries are sometimes called maps because you map a key to a value

```
properties = {}
properties [ "Name" ] = "Caitanya"
properties [ "Born" ] = 'Bengal'
properties [ "Visited" ] = 'Srirangam'
properties [ "SrirangamAddress" ] = "N Chitra St, Srirangam 620006"
properties

properties [ "Name" ] = "Nimai"
properties
```

```
properties [ "Name" ] = "Mahaprabhu"
properties

del properties["Visited"]
properties
properties.keys()
properties.values()
```

Slices

```
marks = (10, 20, 30, 40, 50, 60, 70, 80, 90, 100)
marks[:]
marks[:3]
marks[5:8]
marks[:5]
marks[5:]
marks[5:8]
```

Conditionals

```
mark = 55
passmark = 50
if (mark >= passmark):
    print("you have passed")

if (mark >= passmark):                # pass or fail
    print("passed")
else:
    print("failed")

if (mark % 2 == 0):                   # even or odd
    print("even")
else:
    print("odd")
```

```
month = "february"                    # leap month or not
year = 2012
if (month == "february" and year % 4 == 0)
    ndays = 29
else:
    ndays = 28

n = 5
if (n > 0):                           # positive or negative
    print(n, "is positive")
elif (n < 0):
    print(n, "is negative")
else:
    print(n, "is zero")
```

While Loops

```
i = 1
while (i <= 10):
    print(i)
    i += 1

i = 1
while (i <= 20):
    if (i % 2 == 0):
        i += 1
        continue
    print(i)
    i += 1
```

Check if a number is a power of 2

```
n = 1024
while (True):
    if (n < 2):
        print("no, it is not a power of 2")
        break
    if (n == 2):
        print("yes, it is a power of 2")
        break
    remainder = n % 2
    if (remainder == 1):
        print("no, it is odd, so it is not a power of 2")
        break
    n = n / 2                                   # think about it, if a number is a
                                                # a power of 2, then twice that
                                                # number is also a power of 2,
                                                # similarly, half that number
                                                # the exception being the number 2
```

For Loops

```
for i in [1, 2, 3, 4, 5]:
    print(i)

for i in range(1, 10):
    print(i)

for i in range(10):
    print(i)

for i in range(0, 20):
    print(i)

for i in range(0, 20, 2):
    print(i)

for i in range(1, 100, 2):
    print(i)
```

```
for year in range(2000, 2014, 4):
    print(year)

for year in range(2000, 2014, 4):
    if (year % 4 == 0):
        print(year, "is a leap year")
    else:
        print(year, "is not a leap year")

weekdays = [ "monday", "tuesday", "wednesday", "thursday", "friday" ]
for d in weekdays:
    print(d)
```

Exceptions

```
done = False
while not done:
    try:                                # try to execute
        s = input(“Enter number>”)      #     user input of a string
        i = int(s)                      #     convert string to integer
        if (i < 0):                     #     if negative number
            done = True                 #         remember to exit
    except:                             # on error i.e. unable to convert
        done = False                    #     we need to try again
        print(“Exception Handling”)     #     notify user of input error
    finally:                            # whatever happens
        print(“Let us continue ...”)    #     always execute this
```

Functions

```
define functionName(arguments):
    body

def learnCoding():
    print("Learn from SRILA Labs")

learnCoding()

def learnCodingWell():
    print("Learn better from SRILA Labs")
    return 1

learnCodingWell()

value = learnCodingWell()
value
```

```
def isEven(number):
    remainder = number % 2
    if (remainder == 0):
        print(number, "is even")
    else:
        print(number, "is odd")

>>> isEven(0)
0 is even
>>> isEven(1)
1 is odd
>>> isEven(10)
10 is even
>>> isEven(11)
11 is odd
```

Local Variables

```
>>> def doubleNumber(i):
...     i = i * 2
...     return i
...
>>> a = 10
>>> b = doubleNumber(a)
>>> a
10
>>> b
20
```

Global Variables

```
a = 10
b = 20
def doubleNumber2(i):
    global a
    i = i * 2
    a = a * 10
    return i

>>> a
10
>>> b
20
>>> b = doubleNumber2(a)
>>> a
100
>>> b
20
```

Comments

```
def doubleNumber3(i):
    '''
    Multiline comments starting and ending with 3 single quotes
    This function returns twice the value of parameter
    It also has the side effect of scaling the global
    variable a by a factor of 10
    '''
    global a                    # you can write single line comments
    a = a * 10                  # like this ...
    i = i * 2                   # double the parameter
    return i

>>> a
100
>>> doubleNumber3(a)
200
```

Random Numbers

```
import random
random.randint(0,999999999)
9000000000 + random.randint(0,999999999)  # random cell number
9444000000 + random.randint(1,     999999)  # generate bsnl cell number
                                            # with prefix 9444...

while (True):                               # forever loop, ^C to stop
    cellno = 9444000000 + random.randint(1, 999999)
    print(cellno)
```

```
def isEven(n):
    return (n % 2 == 0)

def isOdd(n):
    return (n % 2 == 1)

def isPowerOf2(n):
    if (n <= 1):
        return False
    if (n == 2):
        return True
    if (n % 2 == 1):
        return False
    # now n is > 2 and is Even
    # check if it's half is a power of 2
    # e.g. 8 is a power of 2 because 4, 2 are powers of 2
    return isPowerOf2(n/2)

def isDivisibleBy(a, b):
    return (a % b == 0)

def isPrime(number):
    if (number < 1):
        return False
    if (number == 1):
        return True
    if (number == 2):
        return True
    if (number > 1):
        if isEven(number):
            print(number, "is divisible by", 2)
            return False
        for divisor in range(3, number, 2):
            print("checking if", number, "is divisible by", divisor)
            if isDivisibleBy(number, divisor):
                print(number, "is divisible by", divisor)
```

```
                    return False
                # number is not divisible by divisor
            # number is not divisible by 3, 5, 7, 9, ... number-1
        # we have handled smaller primes 1, 2, and the 3 and above
        return True

    stop = False
    while not stop:
        s = input("Enter a number> ")
        i = int(s)
        even = isEven(i)
        if (even):
            print(i, "is even")
        else:
            print(i, "is not even")

        odd = isOdd(i)
        if (odd):
            print(i, "is odd")
        else:
            print(i, "is not odd")

        isPower = isPowerOf2(i)
        if (isPower):
            print(i, "is power of 2")
        else:
            print(i, "is not power of 2")

        prime = isPrime(i)
        if (prime):
            print(i, "is prime")
        else:
            print(i, "is not prime")

        if (i == 0):
            stop = True
```