Dadian Zhu

Kan Jen Cheng

10/19/2018

Data Science Club

Lecture Note #1

Abstract:

In this meeting, we are going to introduce **data types**, **data storages**, and **data calculations**.

Data Types:

Boolean: *returns true or false*

True:

EX1: Is 3 equals 3.0?

```
print(3 == 3.0) #In python, '==' specifies 'equal'
True
```

False:

EX1: Is Dadian equals Dennis?

```
print('Dadian' == 'Dennis')
False
```

Double: *returns numbers*

Int: *integers*

Float: *numbers with decimal points*

```
print(type(3)) #type() is a function with one argument, or
parameter, or input
<class 'int'>
print(type(3.14))
<class 'float'>
```

String: *return texts*

Structure: *texts inside ‘ ‘ or “ “, even numbers inside the quotation will be treated as simple text*

```
print("Data Science is awesome")
Data Science is awesome
print('Data Science is awesome')
Data Science is awesome
```

Note: *In python, single quotation mark and double quotation mark have no difference.*

Data Storage:

In python, we use " **=** " to store our data into variables.

Storing numbers:

EX1:

```
pi = 3.14
```

Note: *Compared to other programming languages, python does not require users to declare the type of variable. When variable is set up, its type is automatically determined unless the user manually define.*

EX2:

```
x = 1
x = x + 1  #Is this trash?
print(x)

2 #why?
```

Note: In math, $x = x + 1$ is not a correct statement. However, in the world of computer science, single " = " is not "equality" but "assignment". Therefore, in this case, $x + 1$ is assigned to x, meaning that $1 + 1$ is assigned to x, that is the reason why we got 2 here.

What if we want to store several data in one line? In python, we have ***tuple*** and ***list***.

Tuple: *immutable, you cannot change the data inside a tuple*

*Tuple of data is stored in parenthesis **( )***

```
tuple1 = ('a', 'b', 'c')
print(tuple1[0]) #0 in here means offset = 0, referring to a
a                #you can view offset as the distance from the
                 #first number
```

```
print(tuple1[1])
b

print(tuple1[2])
c

tuple1[0] = 'd' #you cannot change the data inside a tuple
TypeError: 'tuple' object does not support item assignment
```

List: *mutable, you can change the data inside a list*

List of data is stored in square brackets ***[ ]***

```
list1 = ['a', 'b', 'c', 'd']
print(list1[0])
a

print(list1[2])
c

print(list1[:])
['a', 'b', 'c', 'd']

print(list1)
['a', 'b', 'c', 'd']

print(list1[:2]) #data from offset = 0 to offset = (2 - 1)
['a', 'b']

print(list1[:3]) #data from offset = 0 to offset = (3 - 1)
['a', 'b', 'c']

list1[0] = 'e'
print(list1[0])
e
```

Data Calculation:

Since the data in tuple or list in python cannot be applied to calculations, we should utilize the module from outside source, meaning that we need to import a module from outside, which is not a build-in module that can be directly used.

Here, we are going to introduce ***Numpy*** module, which is important in data calculations.

Numpy: http://www.numpy.org/

Download: *open your terminal or command line, type pip install numpy*

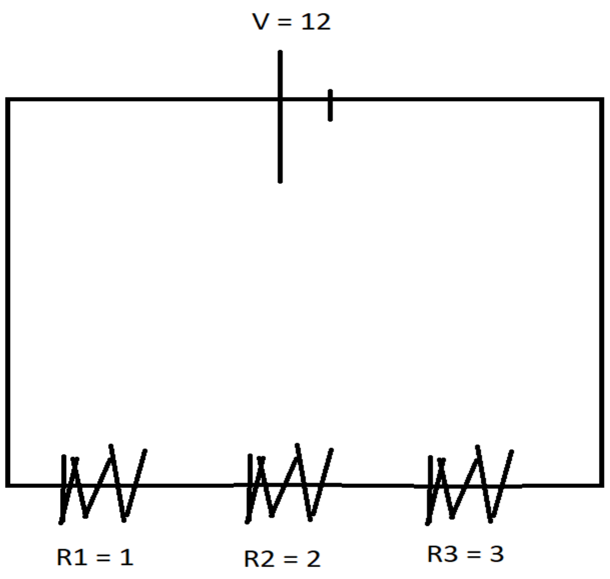
Usage: before using Numpy, you need to tell your computer that you are going to use it, therefore you need to import it

```
import numpy as np #as in here is to shorten the name numpy
                   #as np
```

EX1: Suppose you have a power supply V with 3 resistors R_1, R_2, and R_3 connected in series, how to calculate the potential difference, the power dissipated across each resistor?

$$V = I\,R$$

$$P = I\,V$$



```
import numpy as np

V = 12
R1 = 1
R2 = 2
R3 = 3
Req = R1 + R2 + R3
R = np.array([R1,R2,R3])

I = V / Req
V = R * I
P = I * V

print(V)
print(P)

[2. 4. 6.]
[ 4.  8. 12.]
```