3. Deep learning tools
3.1 Phython: an overview

Manel Martínez-Ramón
Meenu Ajith
Aswathy Rajendra Kurup

Introduction

- First released in 1991.
- Syntax designed for code readability.
- Extendable: Interacts with C/C++.
- Large range if libraries. Compatible with all OS.

- Downloads and tutorials in python.org.

Variables

- Lists and tuples:

```
1  #this is a list
2  list=[2,4,'hello',3]
3  #this is a tuple
4  tuple=(2,4,'hello',3)
```

- Strings, and sets:

```
1  #this is a string
2  string='hello'
3  #This is a proper set
4  set={0,1, (2,4,'hello',3)}
5  #This set will not work
6  set={0,1, [2,4,'hello',3]}
```

- Tuples and sets are immutable, so they cannot contain lists.

Variables

- Collection of elements represented as key-value pairs. Keys are immutable data types.

```
1 diction = {'number':23, 'name':'Tom','age': 16}
2 #defining a dictionary
3 the_diction = {} #empty dictionary
```

- To access each element of a dictionary, the keys are used along with the square brackets.

```
1 print(diction['age']) #output is 16.
2 diction['name'] = 'Harry'
3 print(diction)
4 #outputs is {'number':23,'name':'Harry','age': 16}
5 del diction['age']
6 print(diction) #outputs is {'number':23,'name':'Harry'}.
7 print(diction.pop('name'))
8 #outputs 'Harry' and removes it from the dictionary.
```

Conditional Statements

```
1
2 a = 1
3 if a == 2:
4     print('Even prime number')
5 elif a == 3:
6       print('Odd prime number')
7 else:
8     print('Neither prime nor composite number')
```

Loops

```python
for j in range(6,10):
    print(j) #outputs the numbers 6,7,8 and 9.
```

```python
num = [2,4,6,8,10]
for i in num:
  if i == 6:
      break
    print(i) #outputs numbers 2 and 4.
```

```python
seq = ['h','e','l','l','o']
for j in seq:
    print(j, end = '')
#prints "hello"
#command "pass" is used as a placeholder for future code.
```

Functions

Functions are the basic way to write reusable code.

```python
1 def addition(a,b): #function name and parameters.
2     out = a+b #function body.
3     return  out  #return value of the function.
4 #Here the function definition ends.
5 #Note the indentation change from next line.
6 a = 10
7 b = 40
8 out = addition(a,b)
9 print(out) #outputs the number 50.
```

Objects and classes

```
1  class Smartphone: #Defining the object.
2   def __init__(self, brand,storage):
3    self.brand=brand \#Defines the object properties.
4    self.storage=storage
5   def capacity(self): \#Defining the object behaviour.
6    if self.storage >= 512:
7     return 'Large Storage'
8    elif (self.storage < 512) & (self.storage > 128):
9     return 'Average Storage'
10   else:
11    return 'Low Storage'
12
13 phone1 = Smartphone('EyePhone',256)
14 space = phone1.capacity()
15 print(phone1.brand) #outputs 'EyePhone'.
16 print(phone1.storage) #outputs 256.
17 print(phone1.brand, ":",space)
18 #outputs EyePhone : Average Storage.
```