Siddaganga Institute of Technology

IoT Systems

- Logical design with Python

Lecture : Dr. Sumalatha Aradhya
Asst. Professor,
Dept. of CSE, SIT,
Tumakuru

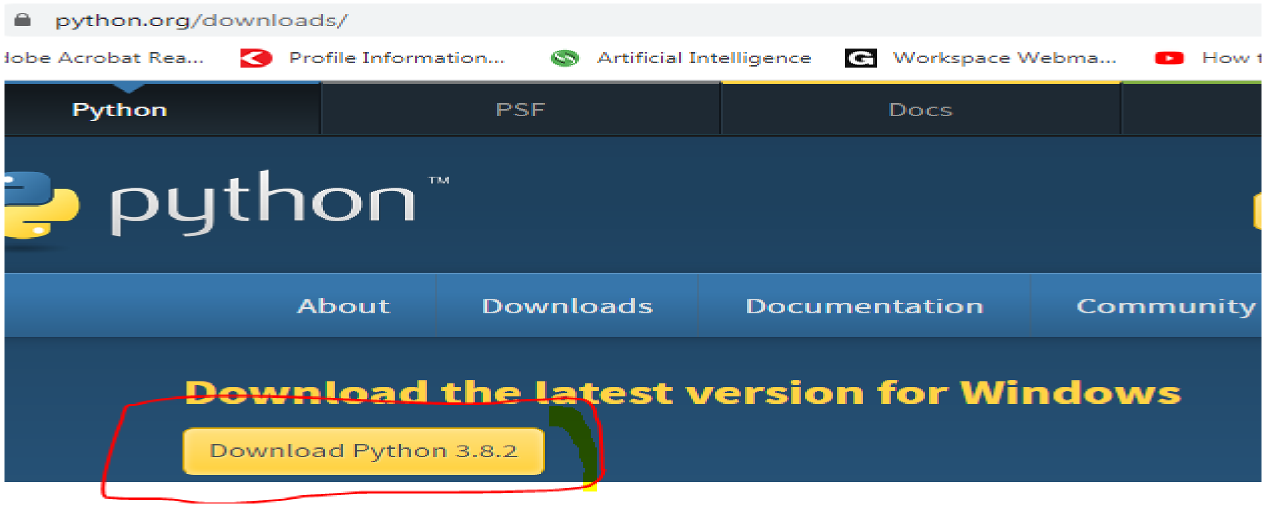
IoT Systems- Introduction to Python

- Python is a general purpose high level programming language
- Python releases: https://www.python.org/doc/versions/
- Characteristics of Python:
  - Multi paradigm programming language
  - Interpreted Language
  - Interactive Language
  - Easy to learn , read and maintain
  - Object and Procedure Oriented
  - Extendable
  - Scalable
  - Portable
  - Board Library Support

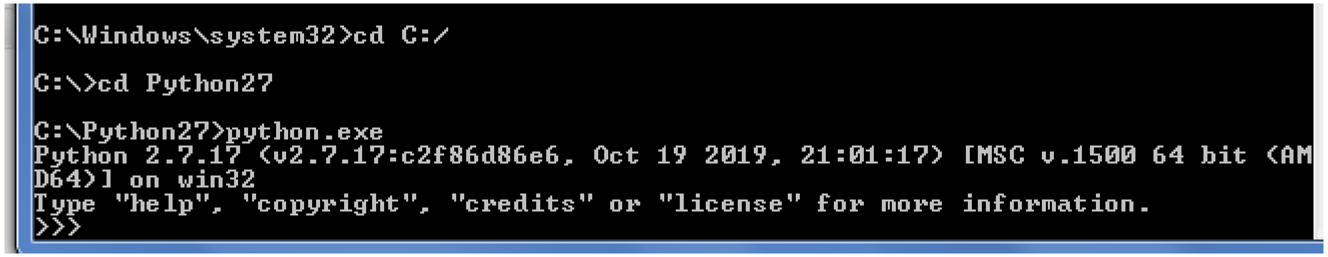
Siddaganga Institute of Technology

Installing Python

- Windows
  - https://www.python.org/downloads/



- run the python at the command shell prompt



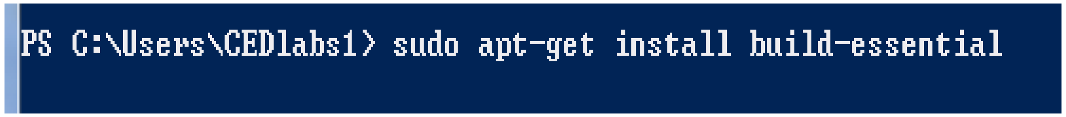
```
C:\Windows\system32>cd C:/

C:\>cd Python27

C:\Python27>python.exe
Python 2.7.17 (v2.7.17:c2f86d86e6, Oct 19 2019, 21:01:17) [MSC v.1500 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Installing Python

- **Linux**

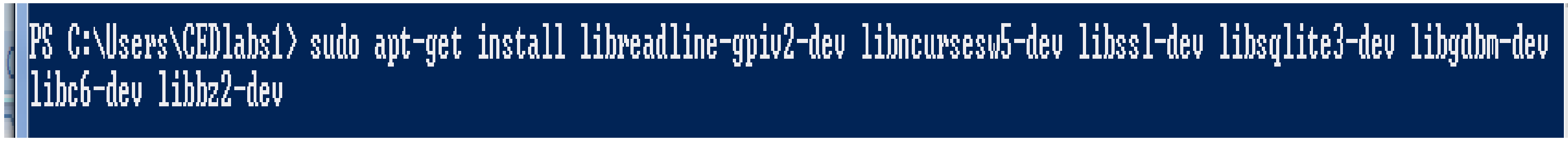
Follow the commands to install python

1



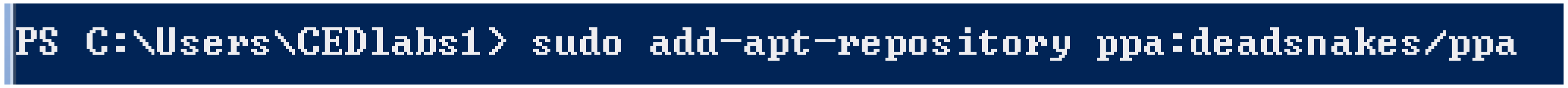
```
PS C:\Users\CEDlabs1> sudo apt-get install build-essential
```

2



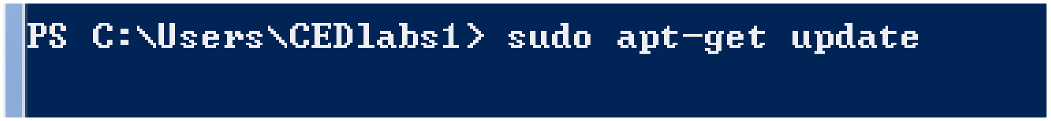
```
PS C:\Users\CEDlabs1> sudo apt-get install libreadline-gplv2-dev libncursesw5-dev libssl-dev libsqlite3-dev libgdbm-dev
libc6-dev libbz2-dev
```

3



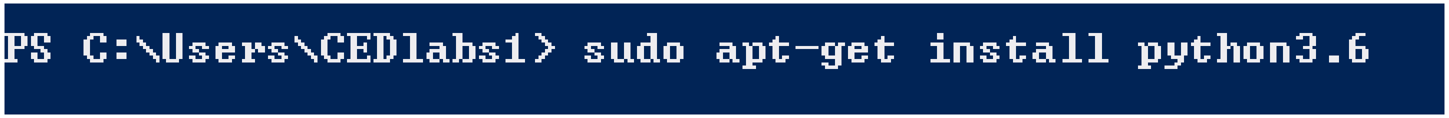
```
PS C:\Users\CEDlabs1> sudo add-apt-repository ppa:deadsnakes/ppa
```

4



```
PS C:\Users\CEDlabs1> sudo apt-get update
```

5



```
PS C:\Users\CEDlabs1> sudo apt-get install python3.6
```

Python Data Types and Data Structures

- Numbers
- Strings
- Lists
- Tuples
- Dictionaries

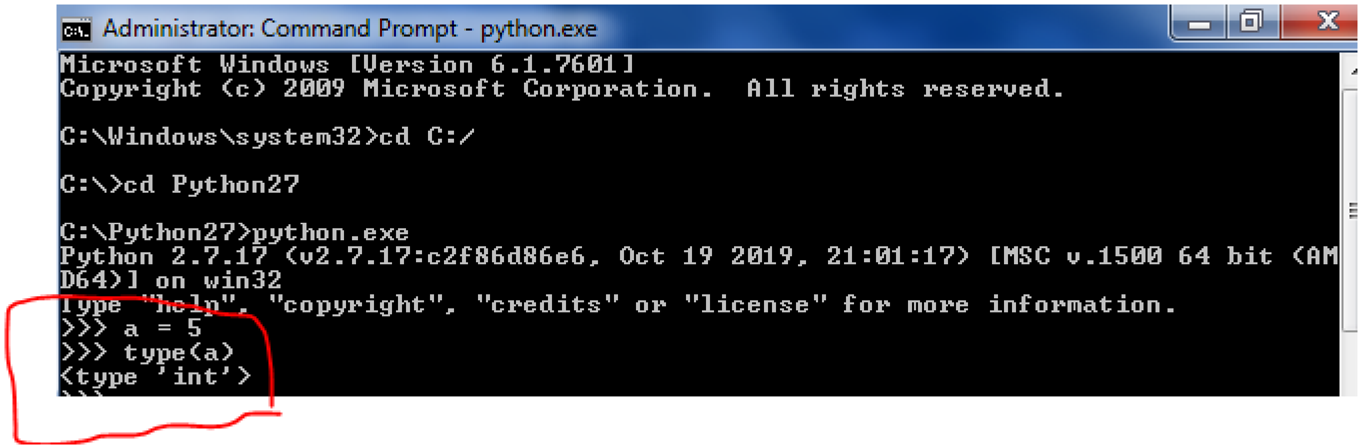
Python Data Types and Data Structures:

❖Numbers

Used to store numeric values.
Immutable data types =>
changing the value of a number data type results in a newly allocated object.

➢Working with numbers in Python:

○***Integers:***


Administrator: Command Prompt - python.exe

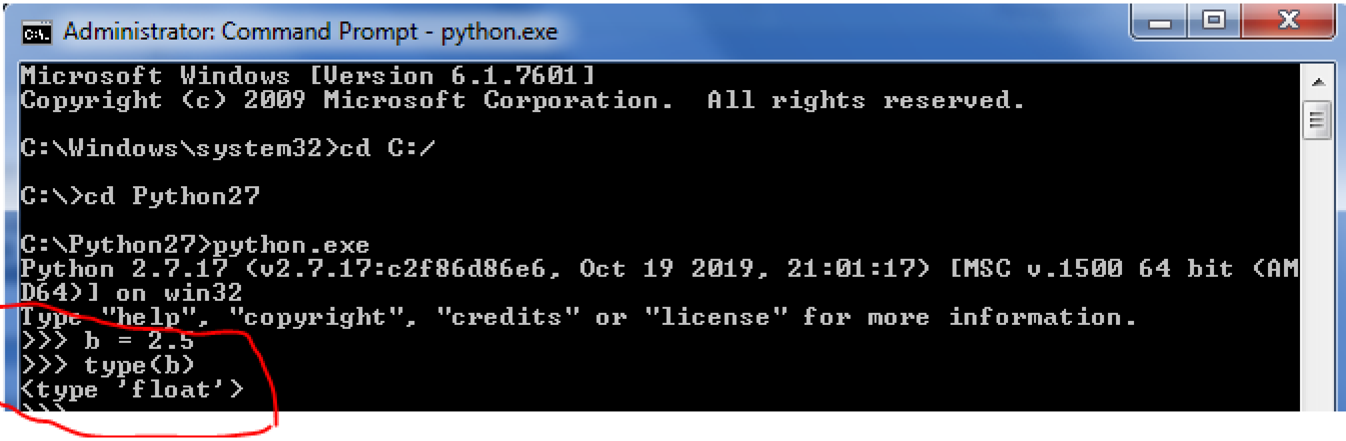
```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>cd C:/

C:\>cd Python27

C:\Python27>python.exe
Python 2.7.17 (v2.7.17:c2f86d86e6, Oct 19 2019, 21:01:17) [MSC v.1500 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> a = 5
>>> type(a)
<type 'int'>
>>>
```

Siddaganga Institute of Technology

Python Data Types and Data Structures:

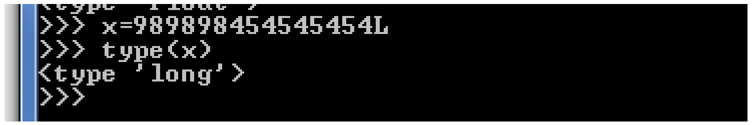
Working with numbers in Python:

○ ***Floating Point:***



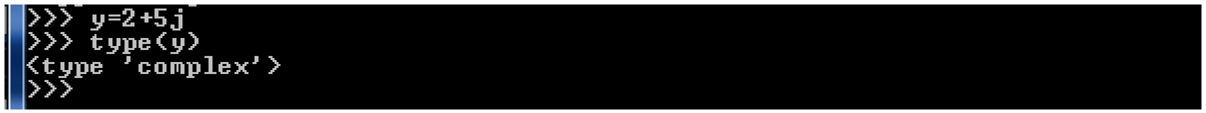
```
Administrator: Command Prompt - python.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>cd C:/

C:\>cd Python27

C:\Python27>python.exe
Python 2.7.17 (v2.7.17:c2f86d86e6, Oct 19 2019, 21:01:17) [MSC v.1500 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> b = 2.5
>>> type(b)
<type 'float'>
>>>
```

○ ***Long***

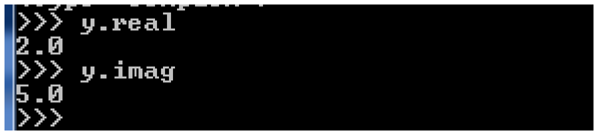


```
>>> x=9898984545454 54L
>>> type(x)
<type 'long'>
>>>
```

○ ***Complex***



```
>>> y=2+5j
>>> type(y)
<type 'complex'>
>>>
```

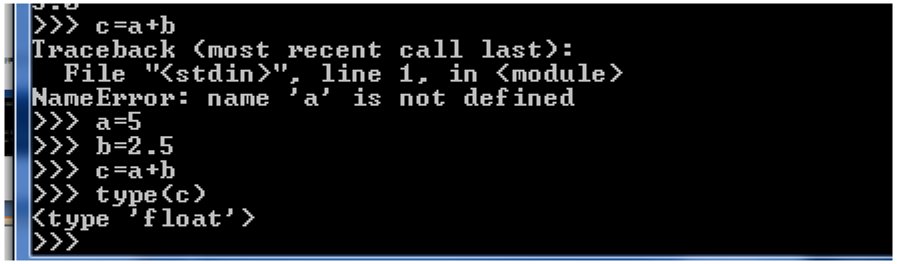


```
>>> y.real
2.0
>>> y.imag
5.0
>>>
```

Python Data Types and Data Structures:

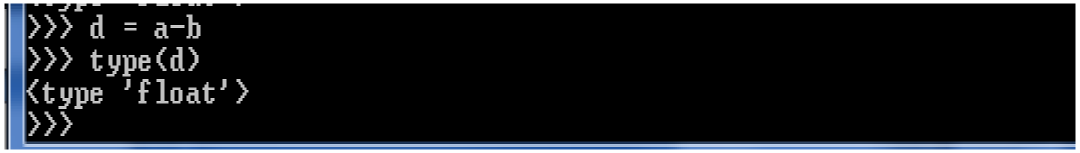
Working with numbers in Python:

- ***Addition*** ➡



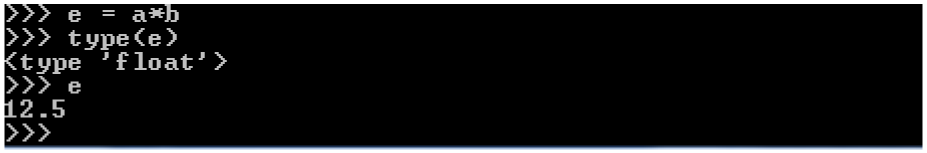
```
>>> c=a+b
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'a' is not defined
>>> a=5
>>> b=2.5
>>> c=a+b
>>> type(c)
<type 'float'>
>>>
```

- ***Subtraction*** ➡



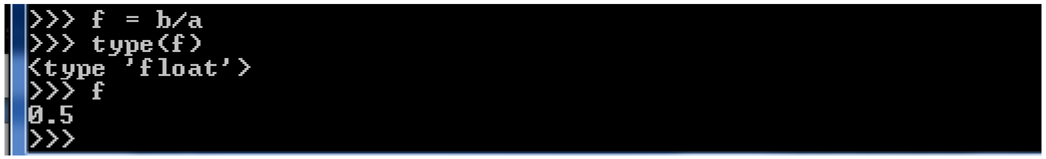
```
>>> d = a-b
>>> type(d)
<type 'float'>
>>>
```

- ***Multiplication*** ➡



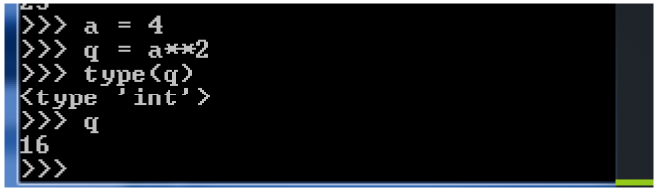
```
>>> e = a*b
>>> type(e)
<type 'float'>
>>> e
12.5
>>>
```

- ***Division*** ➡



```
>>> f = b/a
>>> type(f)
<type 'float'>
>>> f
0.5
>>>
```

- ***Power*** ➡



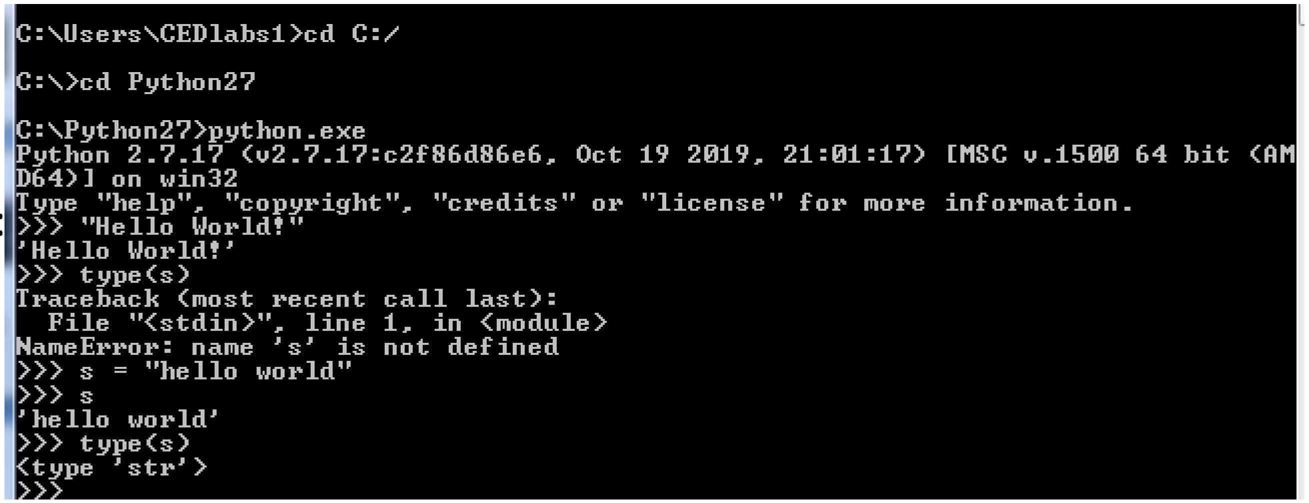
```
>>> a = 4
>>> q = a**2
>>> type(q)
<type 'int'>
>>> q
16
>>>
```

Python Data Types and Data Structures:

Working with Strings in Python:

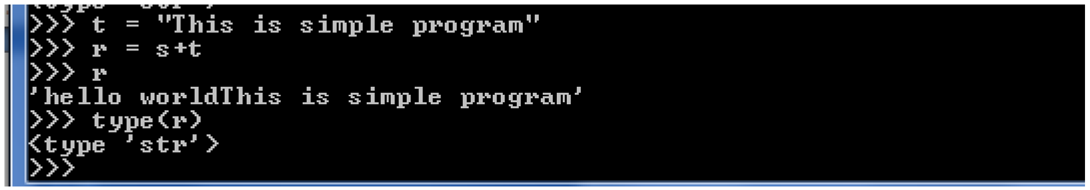
- A string is simply a list of characters in order.
  - No limit to number of characters
  - Empty string -> A string with zero characters
    - Few Examples:

Create a String:



```
C:\Users\CEDlabs1>cd C:/

C:\>cd Python27

C:\Python27>python.exe
Python 2.7.17 (v2.7.17:c2f86d86e6, Oct 19 2019, 21:01:17) [MSC v.1500 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> "Hello World!"
'Hello World!'
>>> type(s)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 's' is not defined
>>> s = "hello world"
>>> s
'hello world'
>>> type(s)
<type 'str'>
>>>
```

String Concatenation:



```
>>> t = "This is simple program"
>>> r = s+t
>>> r
'hello worldThis is simple program'
>>> type(r)
<type 'str'>
>>>
```

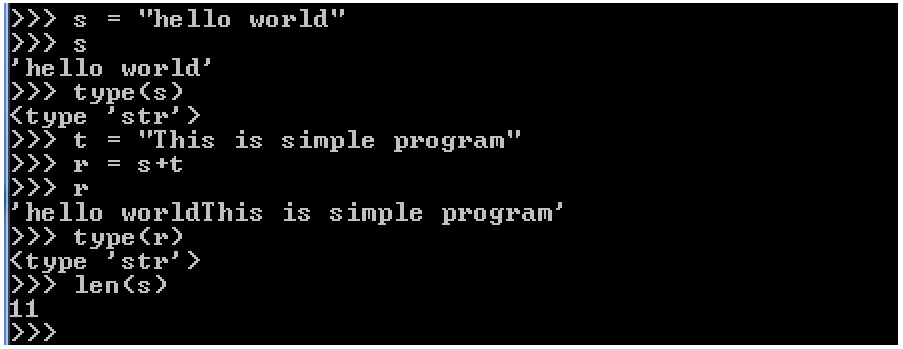
Python Data Types and Data Structures:

Working with Strings in Python:

Siddaganga Institute of Technology

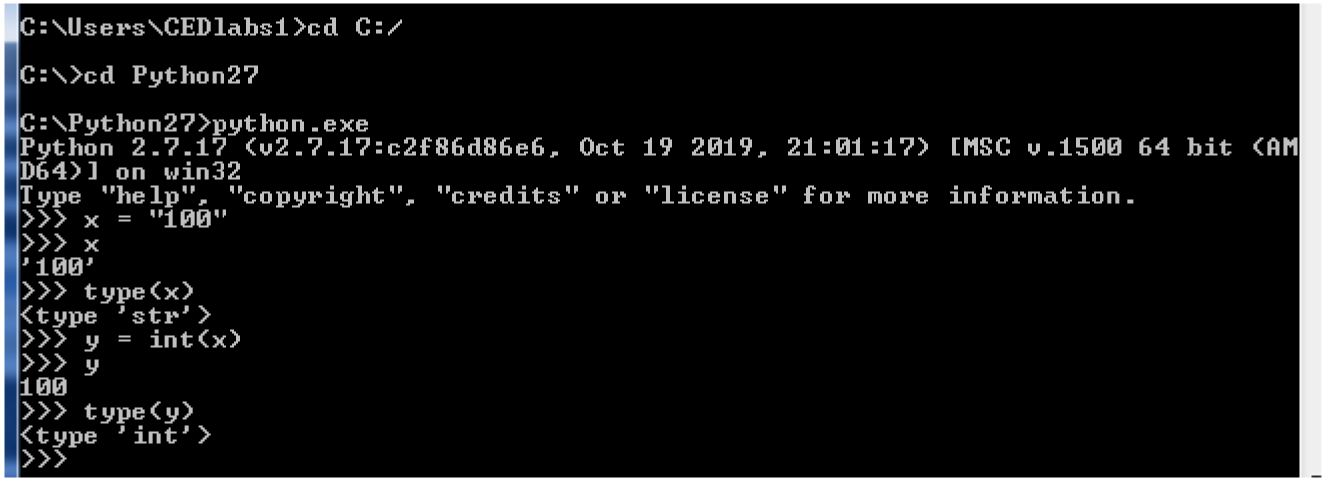
- Few Examples Contd.:

Length of string:



```
>>> s = "hello world"
>>> s
'hello world'
>>> type(s)
<type 'str'>
>>> t = "This is simple program"
>>> r = s+t
>>> r
'hello worldThis is simple program'
>>> type(r)
<type 'str'>
>>> len(s)
11
>>>
```

Convert string to integer:



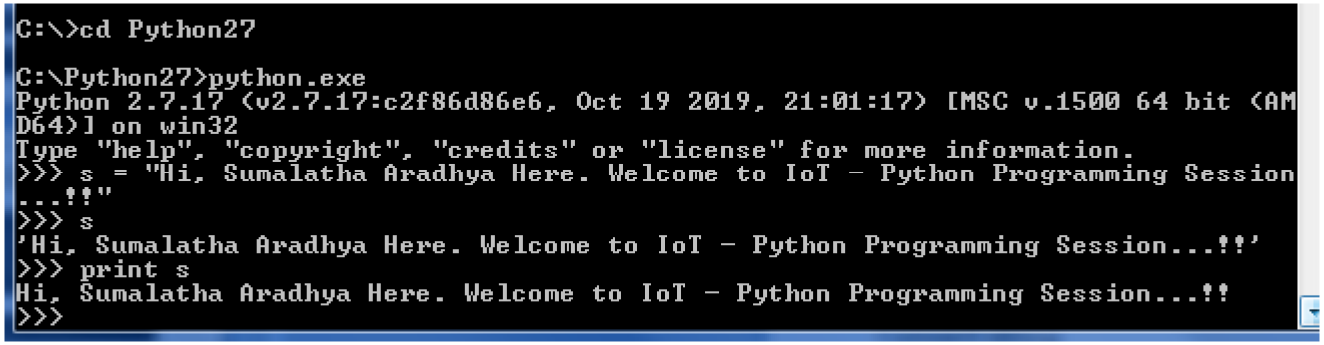
```
C:\Users\CEDlabs1>cd C:/

C:\>cd Python27

C:\Python27>python.exe
Python 2.7.17 (v2.7.17:c2f86d86e6, Oct 19 2019, 21:01:17) [MSC v.1500 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> x = "100"
>>> x
'100'
>>> type(x)
<type 'str'>
>>> y = int(x)
>>> y
100
>>> type(y)
<type 'int'>
>>>
```

Python Data Types and Data Structures:

Working with Strings in Python:

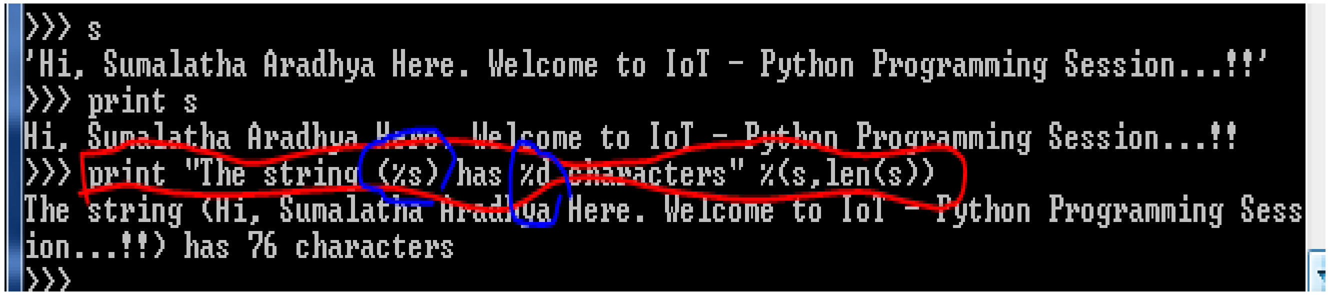
➢Few Examples Contd.:

Print string:



```
C:\>cd Python27

C:\Python27>python.exe
Python 2.7.17 (v2.7.17:c2f86d86e6, Oct 19 2019, 21:01:17) [MSC v.1500 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> s = "Hi, Sumalatha Aradhya Here. Welcome to IoT - Python Programming Session
...!!"
>>> s
'Hi, Sumalatha Aradhya Here. Welcome to IoT - Python Programming Session...!!'
>>> print s
Hi, Sumalatha Aradhya Here. Welcome to IoT - Python Programming Session...!!
>>>
```

Formatting Output:



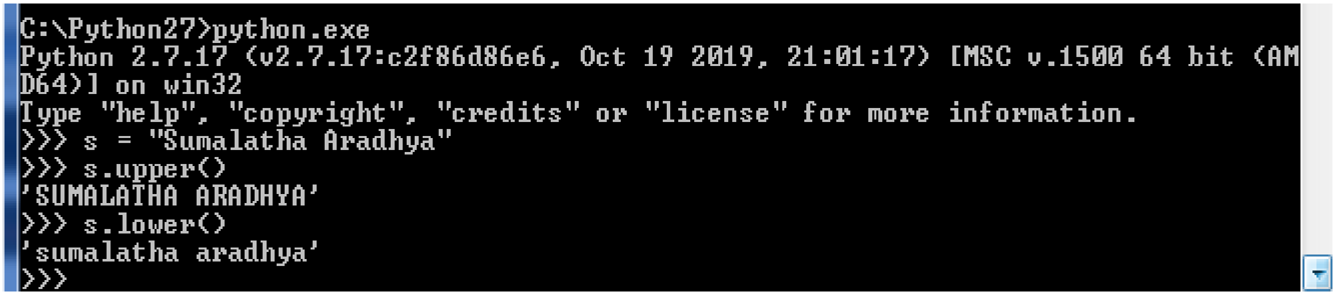
```
>>> s
'Hi, Sumalatha Aradhya Here. Welcome to IoT - Python Programming Session...!!'
>>> print s
Hi, Sumalatha Aradhya Here. Welcome to IoT - Python Programming Session...!!
>>> print "The string (%s) has %d characters" %(s,len(s))
The string (Hi, Sumalatha Aradhya Here. Welcome to IoT - Python Programming Sess
ion...!!) has 76 characters
>>>
```

Python Data Types and Data Structures:

Working with Strings in Python:

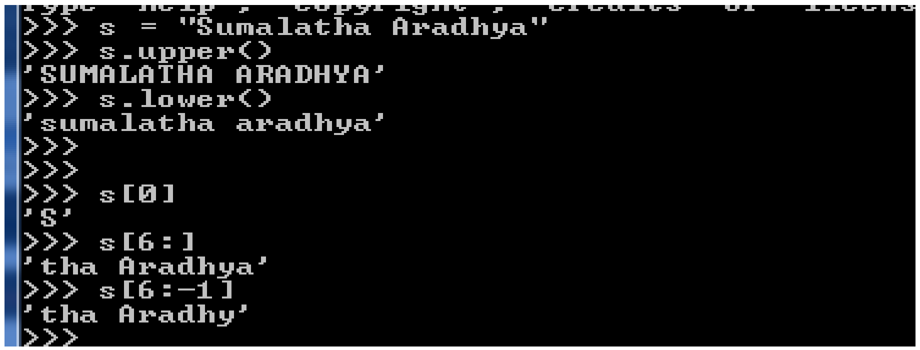
➢Few Examples Contd.:

Convert to upper
Or lower case:



```
C:\Python27>python.exe
Python 2.7.17 (v2.7.17:c2f86d86e6, Oct 19 2019, 21:01:17) [MSC v.1500 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> s = "Sumalatha Aradhya"
>>> s.upper()
'SUMALATHA ARADHYA'
>>> s.lower()
'sumalatha aradhya'
>>>
```

Accessing the
Substring:



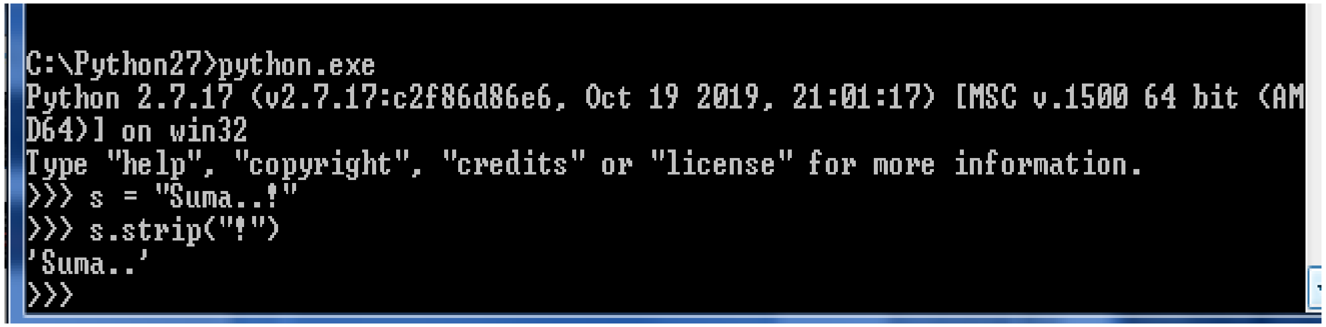
```
>>> s = "Sumalatha Aradhya"
>>> s.upper()
'SUMALATHA ARADHYA'
>>> s.lower()
'sumalatha aradhya'
>>>
>>>
>>> s[0]
'S'
>>> s[6:]
'tha Aradhya'
>>> s[6:-1]
'tha Aradhy'
>>>
```

Python Data Types and Data Structures:

Working with Strings in Python:

- Few Examples Contd.:

Stripping a string:

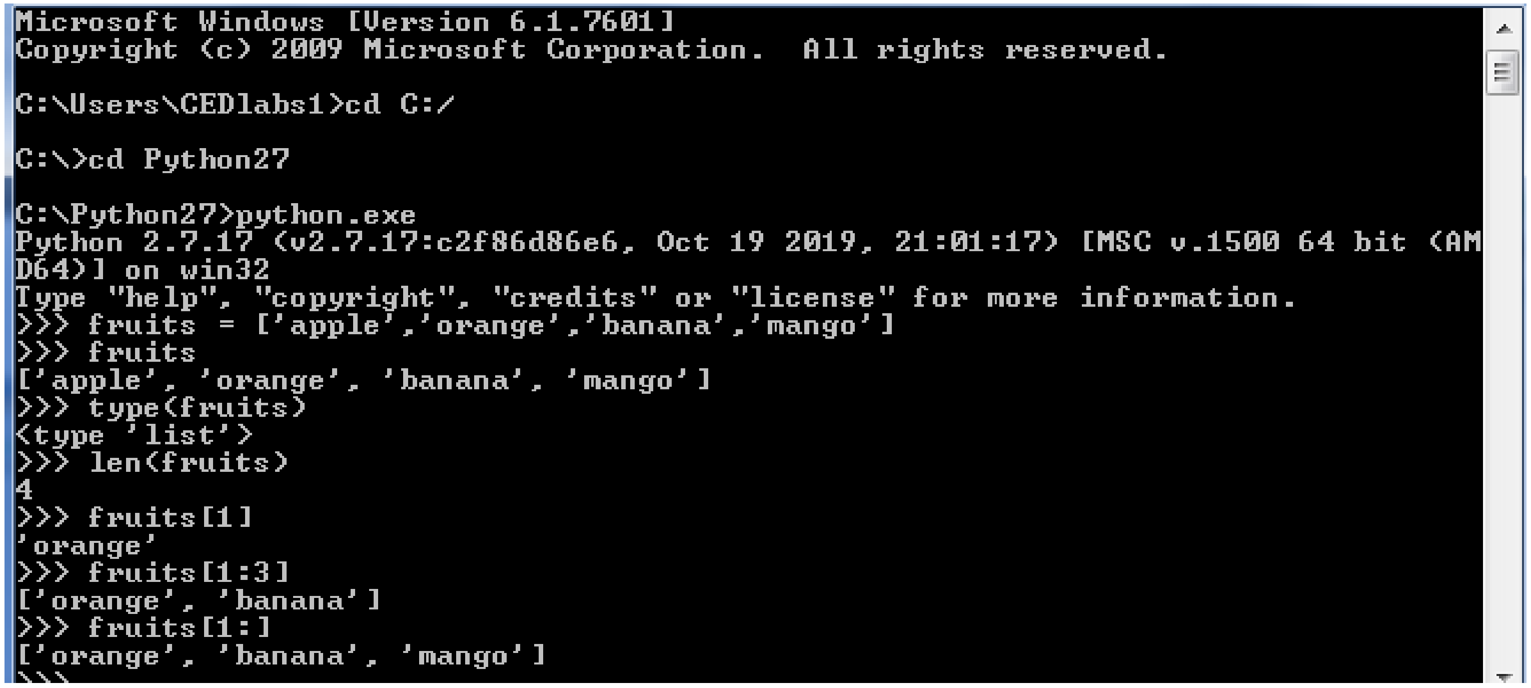


```
C:\Python27>python.exe
Python 2.7.17 (v2.7.17:c2f86d86e6, Oct 19 2019, 21:01:17) [MSC v.1500 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> s = "Suma..!"
>>> s.strip("!")
'Suma..'
>>>
```

Working with Lists in Python:

- List is a compound data type used to group together other values
- List items need not all have the same type
- A list contains items separated by commas and enclosed within square brackets

- Few Examples for list:



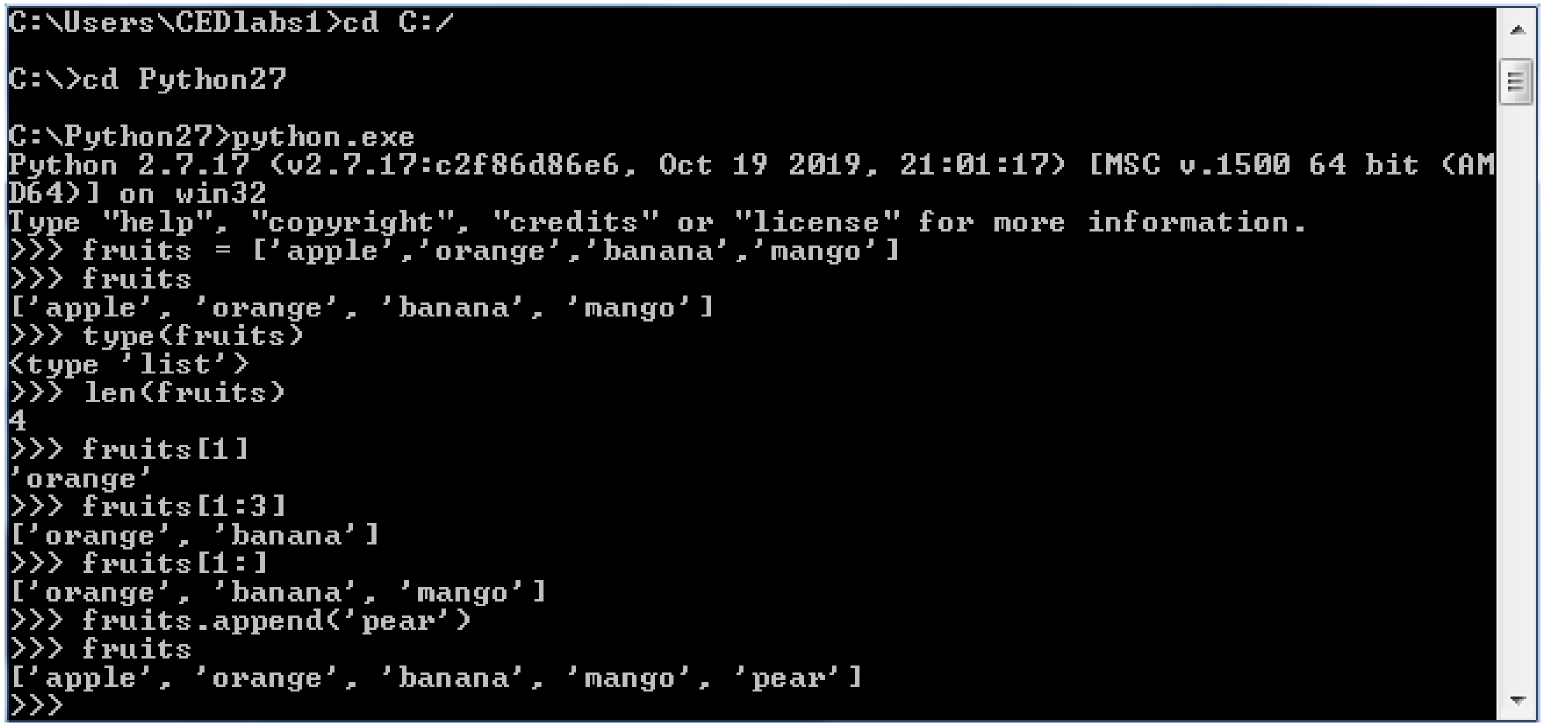
```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\CEDlabs1>cd C:/

C:\>cd Python27

C:\Python27>python.exe
Python 2.7.17 (v2.7.17:c2f86d86e6, Oct 19 2019, 21:01:17) [MSC v.1500 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> fruits = ['apple','orange','banana','mango']
>>> fruits
['apple', 'orange', 'banana', 'mango']
>>> type(fruits)
<type 'list'>
>>> len(fruits)
4
>>> fruits[1]
'orange'
>>> fruits[1:3]
['orange', 'banana']
>>> fruits[1:]
['orange', 'banana', 'mango']
>>>
```

Python Data Types and Data Structures:

Siddaganga Institute of Technology

Working with Lists in Python:

- Few Examples for list contd.:
  - Appending an item to the list:



```
C:\Users\CEDlabs1>cd C:/

C:\>cd Python27

C:\Python27>python.exe
Python 2.7.17 (v2.7.17:c2f86d86e6, Oct 19 2019, 21:01:17) [MSC v.1500 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> fruits = ['apple','orange','banana','mango']
>>> fruits
['apple', 'orange', 'banana', 'mango']
>>> type(fruits)
<type 'list'>
>>> len(fruits)
4
>>> fruits[1]
'orange'
>>> fruits[1:3]
['orange', 'banana']
>>> fruits[1:]
['orange', 'banana', 'mango']
>>> fruits.append('pear')
>>> fruits
['apple', 'orange', 'banana', 'mango', 'pear']
>>>
```

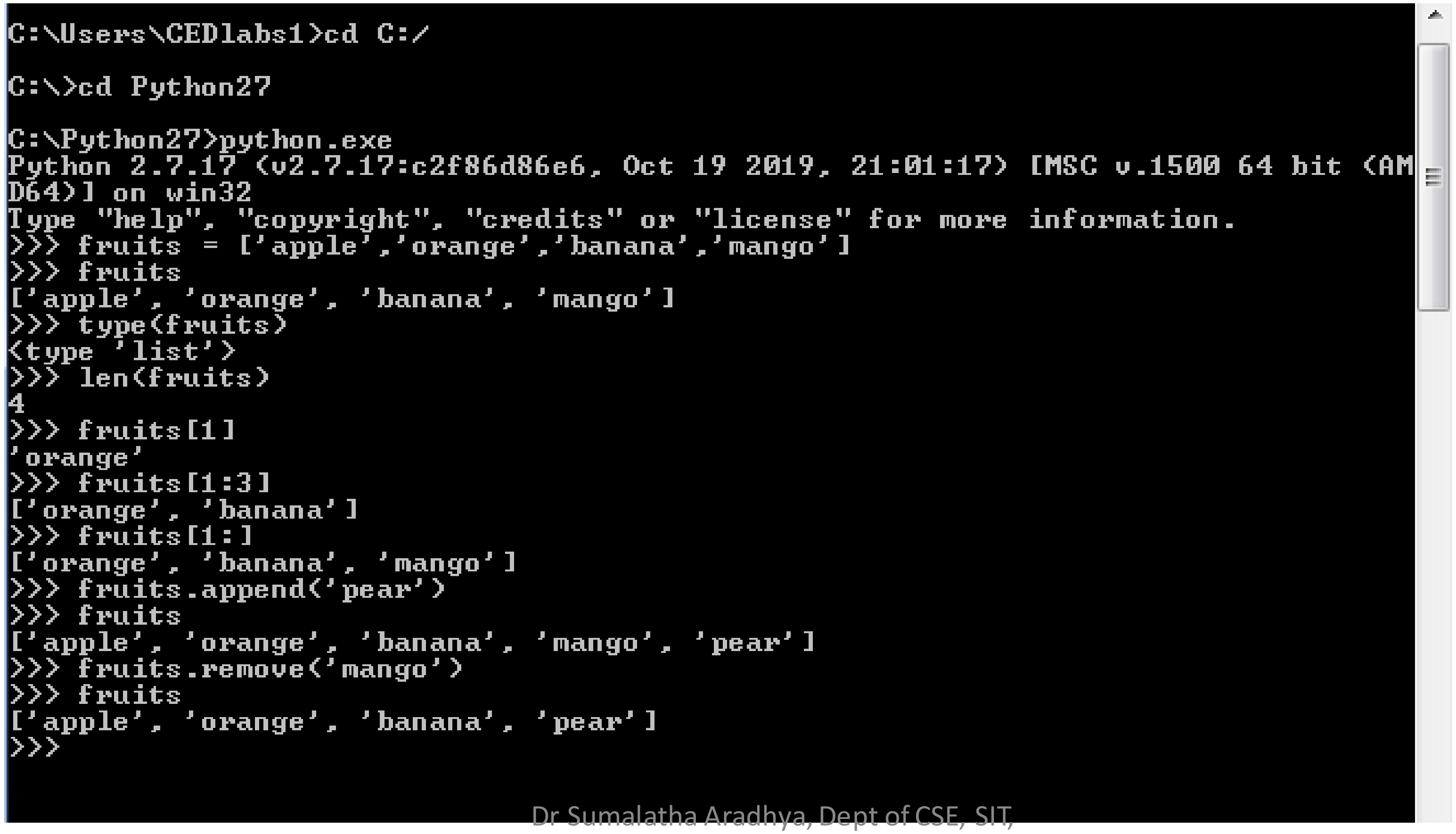
Python Data Types and Data Structures:

Siddaganga Institute of Technology

Working with Lists in Python:

- Few Examples for list contd.:

  - Removing an item from the list:



```
C:\Users\CEDlabs1>cd C:/

C:\>cd Python27

C:\Python27>python.exe
Python 2.7.17 (v2.7.17:c2f86d86e6, Oct 19 2019, 21:01:17) [MSC v.1500 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> fruits = ['apple','orange','banana','mango']
>>> fruits
['apple', 'orange', 'banana', 'mango']
>>> type(fruits)
<type 'list'>
>>> len(fruits)
4
>>> fruits[1]
'orange'
>>> fruits[1:3]
['orange', 'banana']
>>> fruits[1:]
['orange', 'banana', 'mango']
>>> fruits.append('pear')
>>> fruits
['apple', 'orange', 'banana', 'mango', 'pear']
>>> fruits.remove('mango')
>>> fruits
['apple', 'orange', 'banana', 'pear']
>>>
```

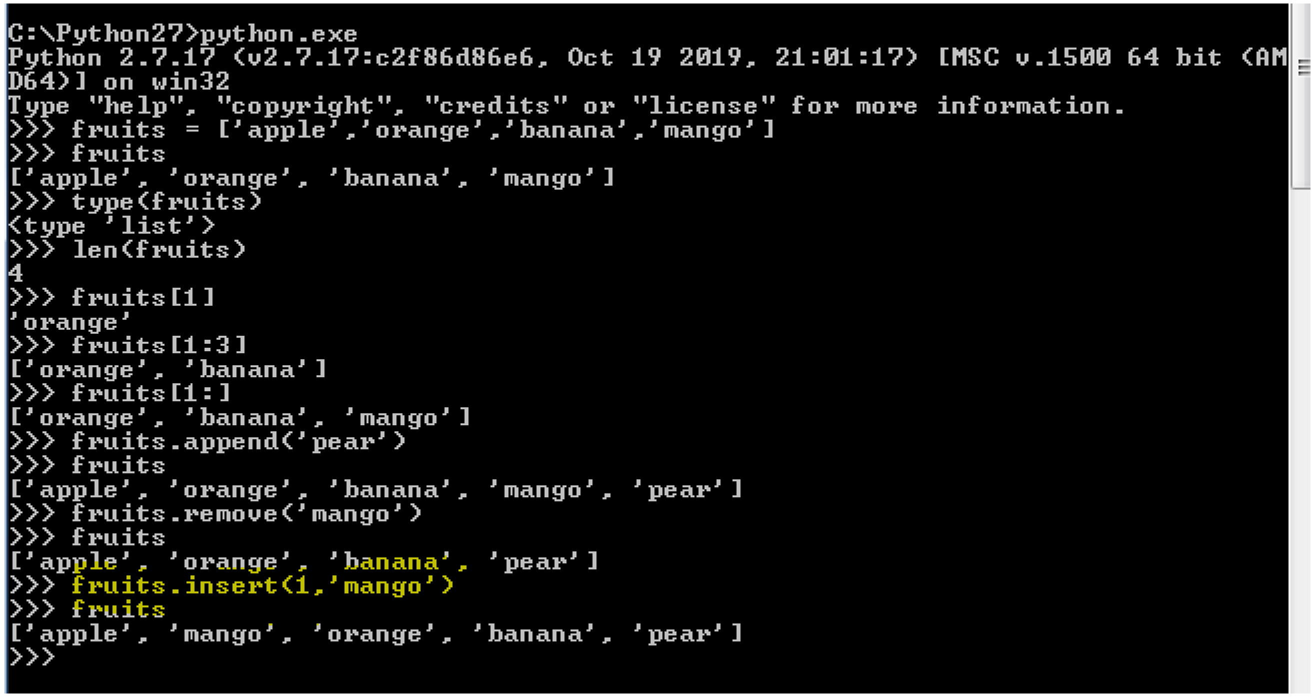
Python Data Types and Data Structures:

Siddaganga Institute of Technology

Working with Lists in Python:

➢Few Examples for list contd.:

▪Inserting an item to the list:



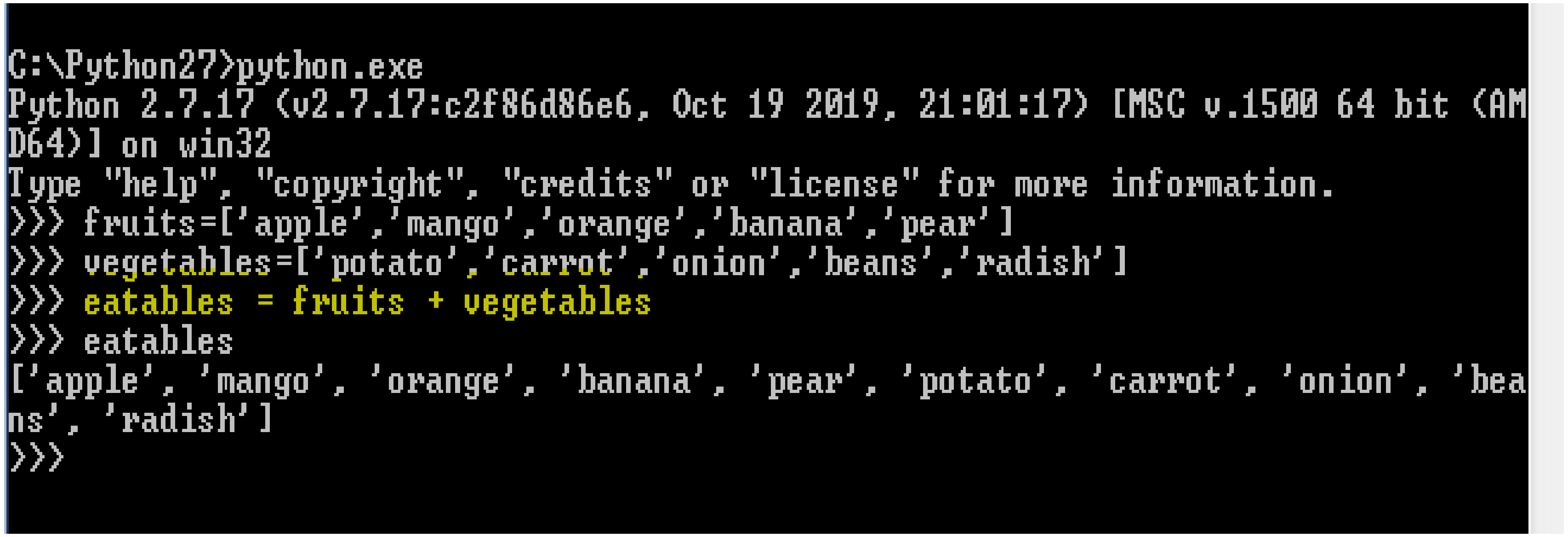
```
C:\Python27>python.exe
Python 2.7.17 (v2.7.17:c2f86d86e6, Oct 19 2019, 21:01:17) [MSC v.1500 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> fruits = ['apple','orange','banana','mango']
>>> fruits
['apple', 'orange', 'banana', 'mango']
>>> type(fruits)
<type 'list'>
>>> len(fruits)
4
>>> fruits[1]
'orange'
>>> fruits[1:3]
['orange', 'banana']
>>> fruits[1:]
['orange', 'banana', 'mango']
>>> fruits.append('pear')
>>> fruits
['apple', 'orange', 'banana', 'mango', 'pear']
>>> fruits.remove('mango')
>>> fruits
['apple', 'orange', 'banana', 'pear']
>>> fruits.insert(1,'mango')
>>> fruits
['apple', 'mango', 'orange', 'banana', 'pear']
>>>
```

Python Data Types and Data Structures:

Working with Lists in Python:

➢Few Examples for list contd.:

▪Combining lists

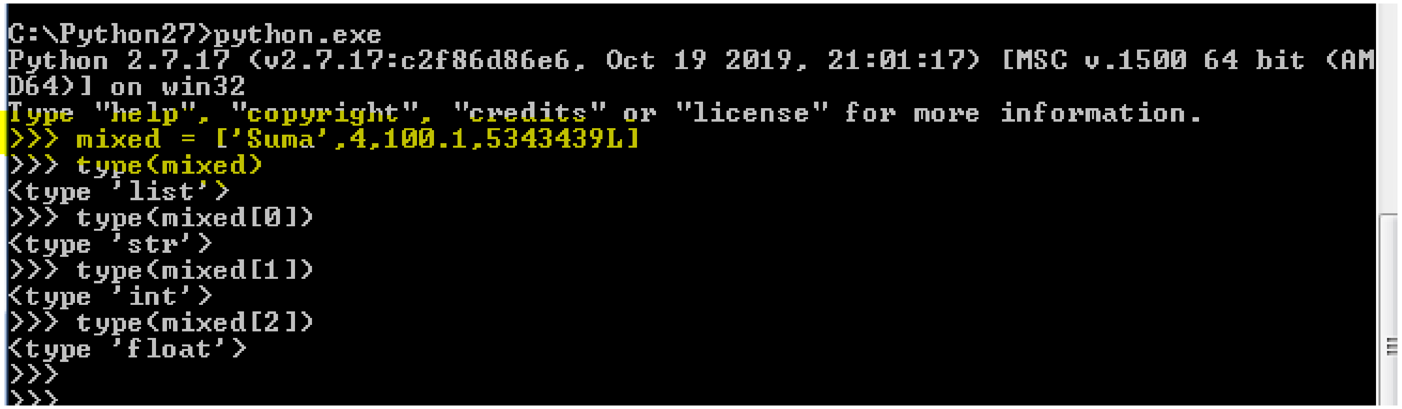


```
C:\Python27>python.exe
Python 2.7.17 (v2.7.17:c2f86d86e6, Oct 19 2019, 21:01:17) [MSC v.1500 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> fruits=['apple','mango','orange','banana','pear']
>>> vegetables=['potato','carrot','onion','beans','radish']
>>> eatables = fruits + vegetables
>>> eatables
['apple', 'mango', 'orange', 'banana', 'pear', 'potato', 'carrot', 'onion', 'bea
ns', 'radish']
>>>
```

Working with Lists in Python:

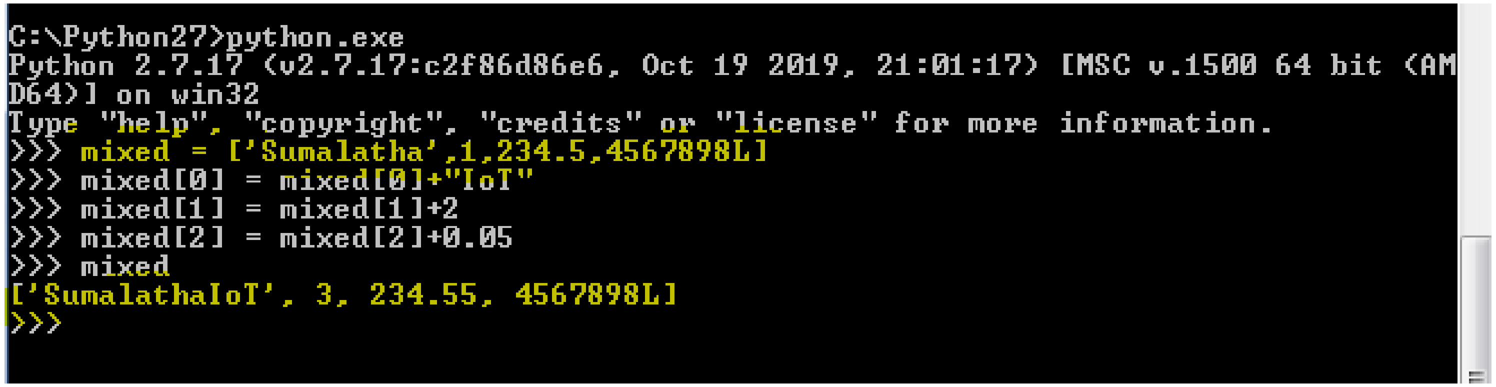
➢Few Examples for list contd.:

- Mixed data types in a list:



```
C:\Python27>python.exe
Python 2.7.17 (v2.7.17:c2f86d86e6, Oct 19 2019, 21:01:17) [MSC v.1500 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> mixed = ['Suma',4,100.1,5343439L]
>>> type(mixed)
<type 'list'>
>>> type(mixed[0])
<type 'str'>
>>> type(mixed[1])
<type 'int'>
>>> type(mixed[2])
<type 'float'>
>>>
>>>
```

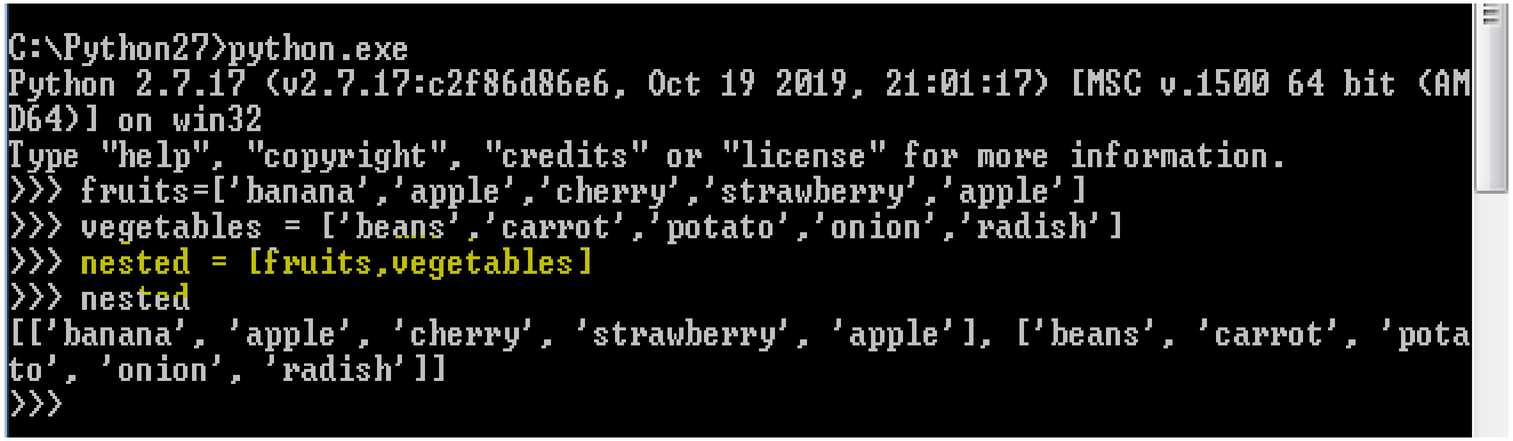
- Change the individual elements in a list:



```
C:\Python27>python.exe
Python 2.7.17 (v2.7.17:c2f86d86e6, Oct 19 2019, 21:01:17) [MSC v.1500 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> mixed = ['Sumalatha',1,234.5,4567898L]
>>> mixed[0] = mixed[0]+"IoT"
>>> mixed[1] = mixed[1]+2
>>> mixed[2] = mixed[2]+0.05
>>> mixed
['SumalathaIoT', 3, 234.55, 4567898L]
>>>
```

Working with Lists in Python:

- Few Examples for list contd.:
  - Lists can be nested:



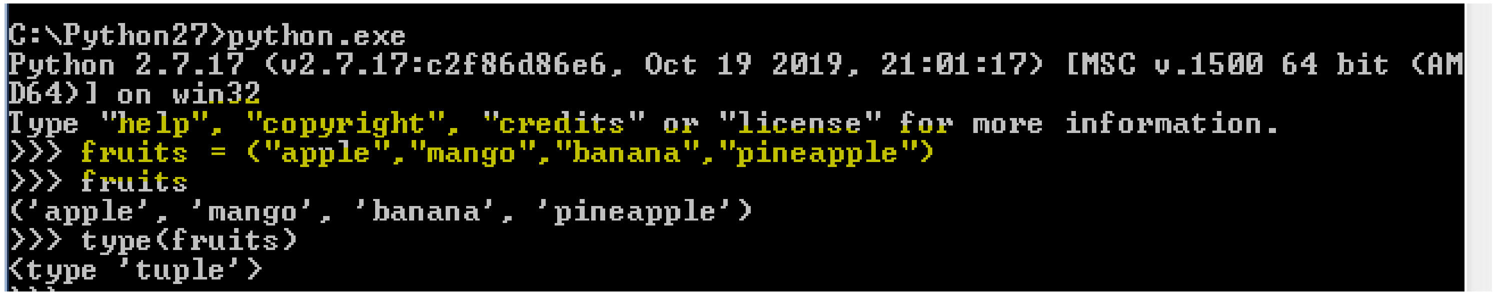
```
C:\Python27>python.exe
Python 2.7.17 (v2.7.17:c2f86d86e6, Oct 19 2019, 21:01:17) [MSC v.1500 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> fruits=['banana','apple','cherry','strawberry','apple']
>>> vegetables = ['beans','carrot','potato','onion','radish']
>>> nested = [fruits,vegetables]
>>> nested
[['banana', 'apple', 'cherry', 'strawberry', 'apple'], ['beans', 'carrot', 'pota
to', 'onion', 'radish']]
>>>
```

Python Data Types and Data Structures:

Working with Tuples in Python:

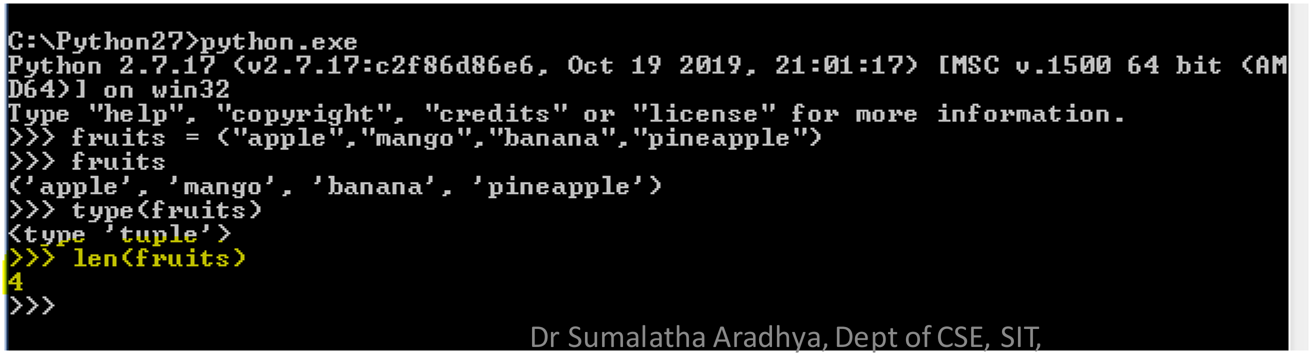
- Tuple is a sequence data type that is similar to the list
- A tuple consists of a number of values separated by commas and enclosed within **parenthesis**.
- List Vs Tuples:
  - The elements of tuple can not be changed
  - Tuples are read only lists

- Few Examples for tuple:



```
C:\Python27>python.exe
Python 2.7.17 (v2.7.17:c2f86d86e6, Oct 19 2019, 21:01:17) [MSC v.1500 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> fruits = ("apple","mango","banana","pineapple")
>>> fruits
('apple', 'mango', 'banana', 'pineapple')
>>> type(fruits)
<type 'tuple'>
```

- Get a length of tuple:

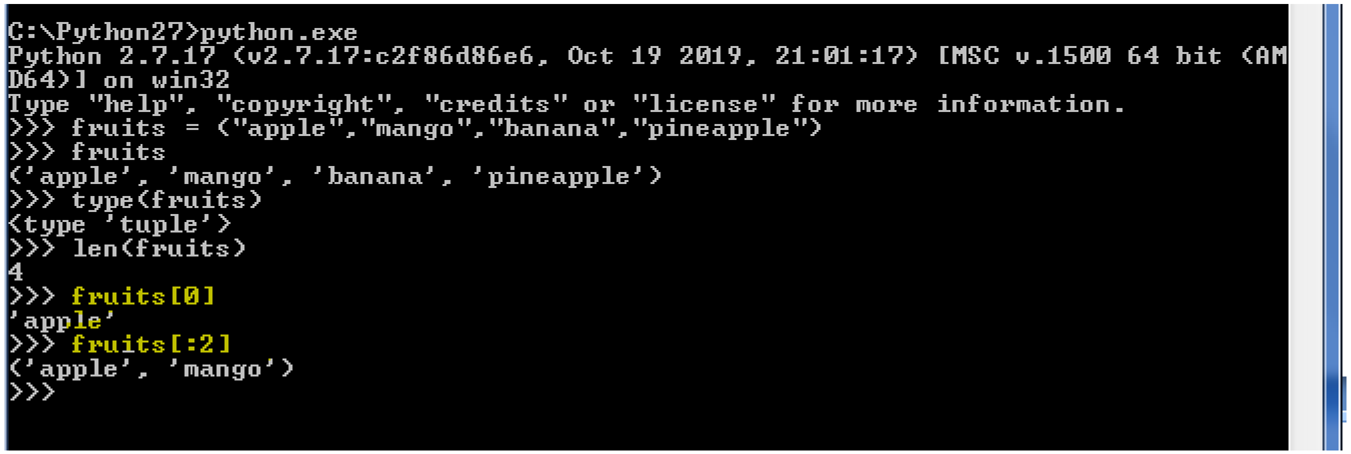


```
C:\Python27>python.exe
Python 2.7.17 (v2.7.17:c2f86d86e6, Oct 19 2019, 21:01:17) [MSC v.1500 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> fruits = ("apple","mango","banana","pineapple")
>>> fruits
('apple', 'mango', 'banana', 'pineapple')
>>> type(fruits)
<type 'tuple'>
>>> len(fruits)
4
>>>
```

Working with Tuples in Python:

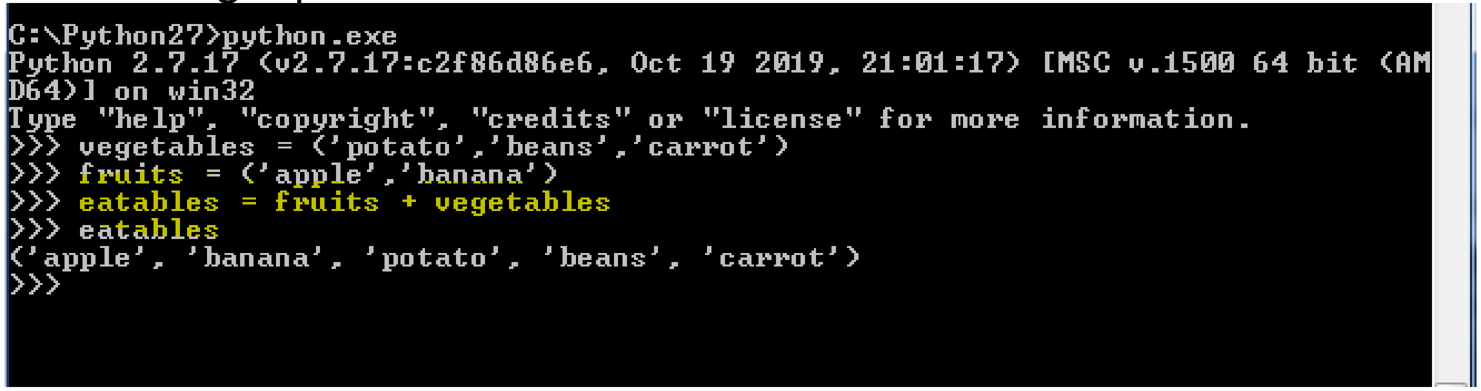
➢Few Examples for tuple contd.:

▪Get an element from tuple:



```
C:\Python27>python.exe
Python 2.7.17 (v2.7.17:c2f86d86e6, Oct 19 2019, 21:01:17) [MSC v.1500 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> fruits = ("apple","mango","banana","pineapple")
>>> fruits
('apple', 'mango', 'banana', 'pineapple')
>>> type(fruits)
<type 'tuple'>
>>> len(fruits)
4
>>> fruits[0]
'apple'
>>> fruits[:2]
('apple', 'mango')
>>>
```

▪Combining tuple:

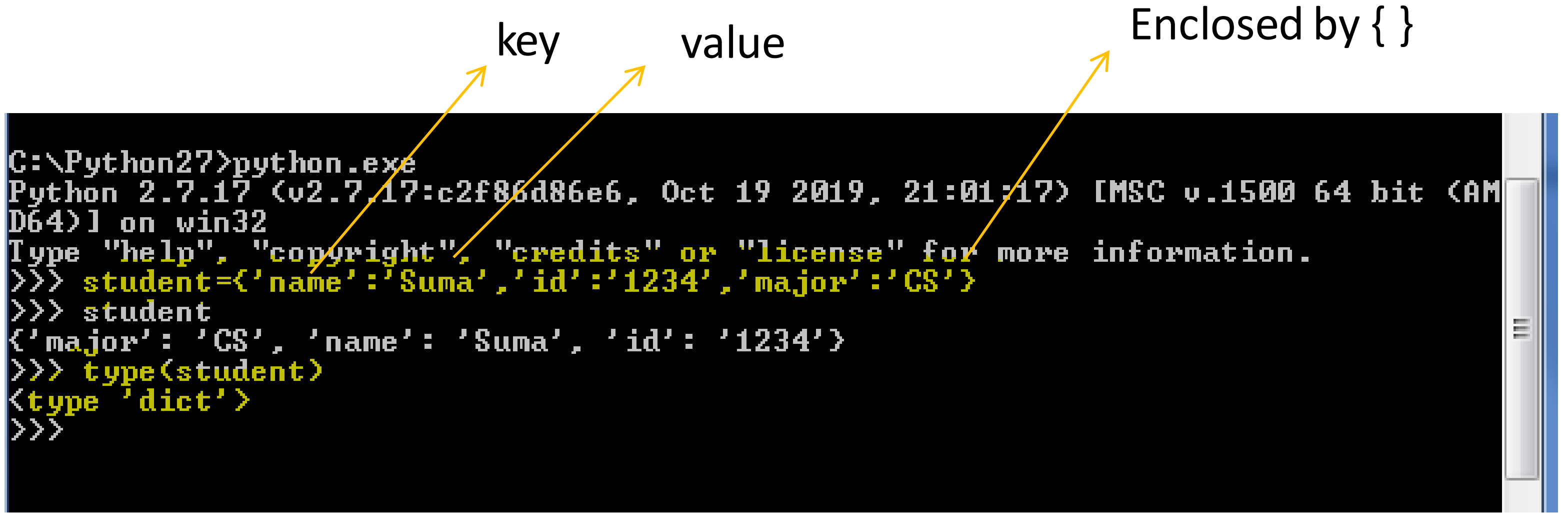


```
C:\Python27>python.exe
Python 2.7.17 (v2.7.17:c2f86d86e6, Oct 19 2019, 21:01:17) [MSC v.1500 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> vegetables = ('potato','beans','carrot')
>>> fruits = ('apple','banana')
>>> eatables = fruits + vegetables
>>> eatables
('apple', 'banana', 'potato', 'beans', 'carrot')
>>>
```

Working with Dictionaries in Python:

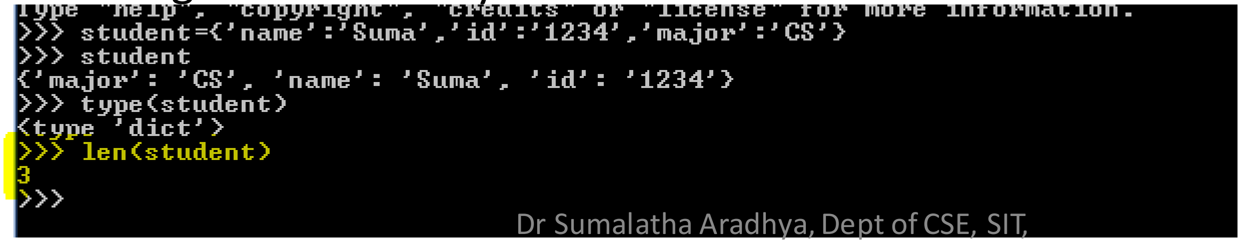
- Dictionary is a mapping data type or a kind of hash table that maps keys to values
- Keys in a dictionary can be of any data type, though members and strings are commonly used for keys
- Values in a dictionary can be of any data type or object.

- Few Examples for dictionary:



```
C:\Python27>python.exe
Python 2.7.17 (v2.7.17:c2f86d86e6, Oct 19 2019, 21:01:17) [MSC v.1500 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> student={'name':'Suma','id':'1234','major':'CS'}
>>> student
{'major': 'CS', 'name': 'Suma', 'id': '1234'}
>>> type(student)
<type 'dict'>
>>>
```

- Get a length of dictionary



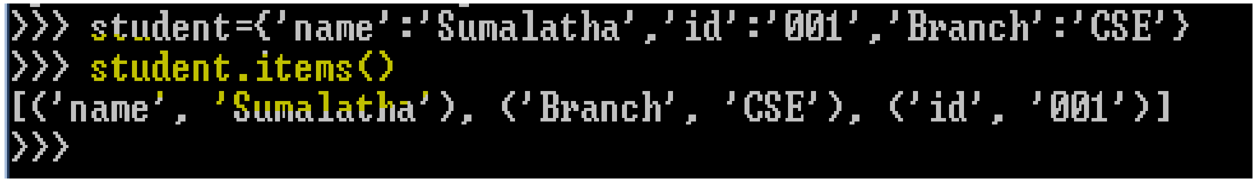
```
Type "help", "copyright", "credits" or "license" for more information.
>>> student={'name':'Suma','id':'1234','major':'CS'}
>>> student
{'major': 'CS', 'name': 'Suma', 'id': '1234'}
>>> type(student)
<type 'dict'>
>>> len(student)
3
>>>
```

Python Data Types and Data Structures:

Working with Dictionaries in Python:

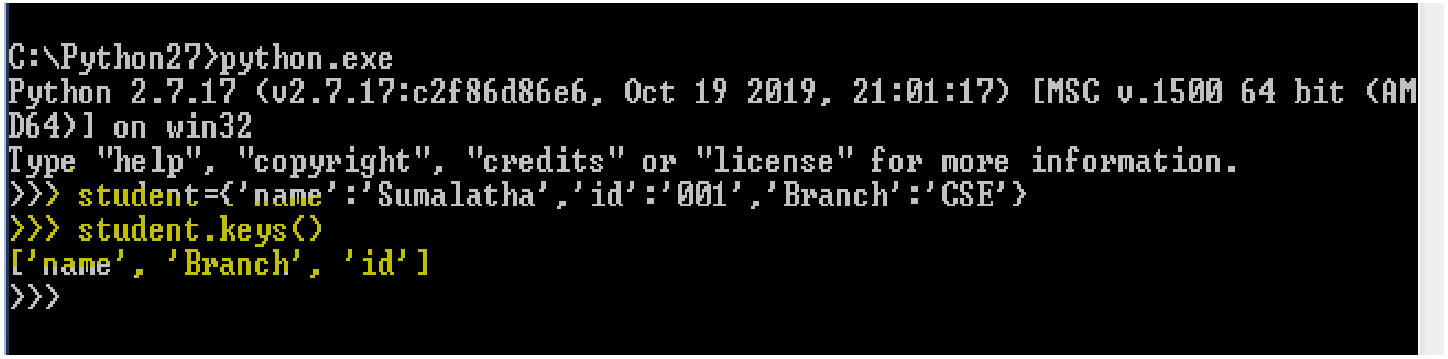
➢Few Examples for dictionary contd.:

- Get all items in a dictionary



```
>>> student={'name':'Sumalatha','id':'001','Branch':'CSE'}
>>> student.items()
[('name', 'Sumalatha'), ('Branch', 'CSE'), ('id', '001')]
>>>
```

- Get all keys in a dictionary



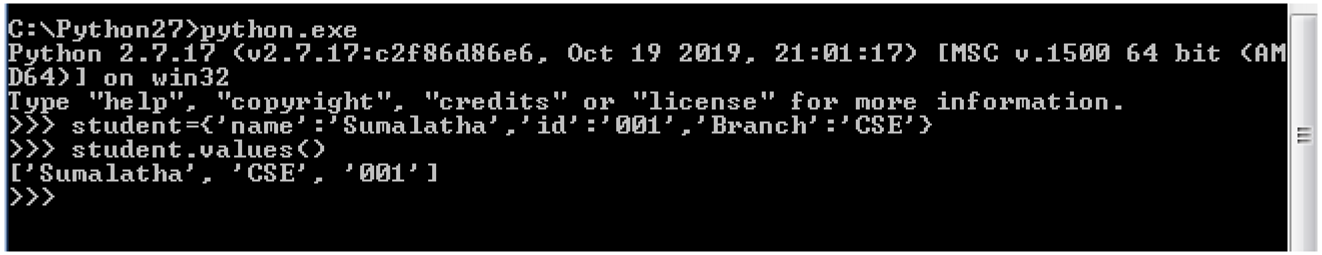
```
C:\Python27>python.exe
Python 2.7.17 (v2.7.17:c2f86d86e6, Oct 19 2019, 21:01:17) [MSC v.1500 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> student={'name':'Sumalatha','id':'001','Branch':'CSE'}
>>> student.keys()
['name', 'Branch', 'id']
>>>
```

Python Data Types and Data Structures:

Working with Dictionaries in Python:

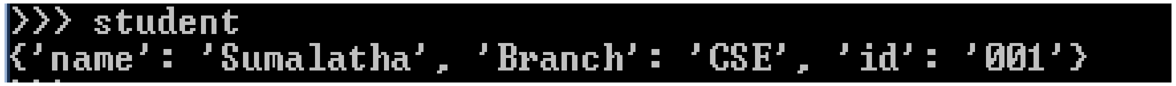
➢Few Examples for dictionary contd.:

- Get all values in a dictionary



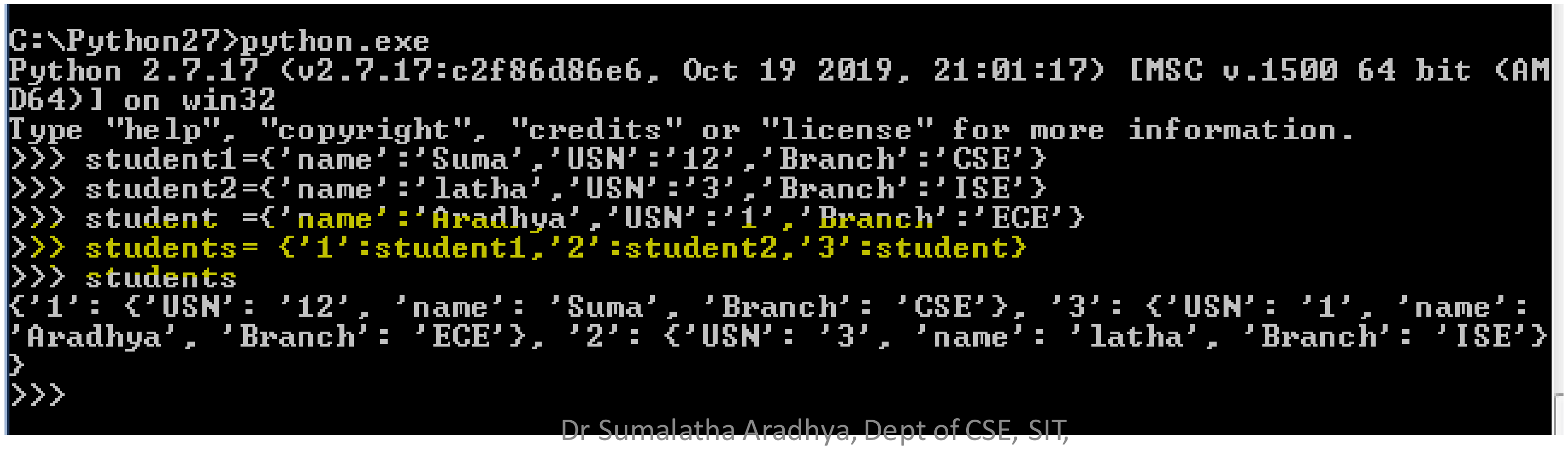
```
C:\Python27>python.exe
Python 2.7.17 (v2.7.17:c2f86d86e6, Oct 19 2019, 21:01:17) [MSC v.1500 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> student={'name':'Sumalatha','id':'001','Branch':'CSE'}
>>> student.values()
['Sumalatha', 'CSE', '001']
>>>
```

- @interpreter – get entire dictionary



```
>>> student
{'name': 'Sumalatha', 'Branch': 'CSE', 'id': '001'}
```

- A value in a dictionary can be in another dictionary

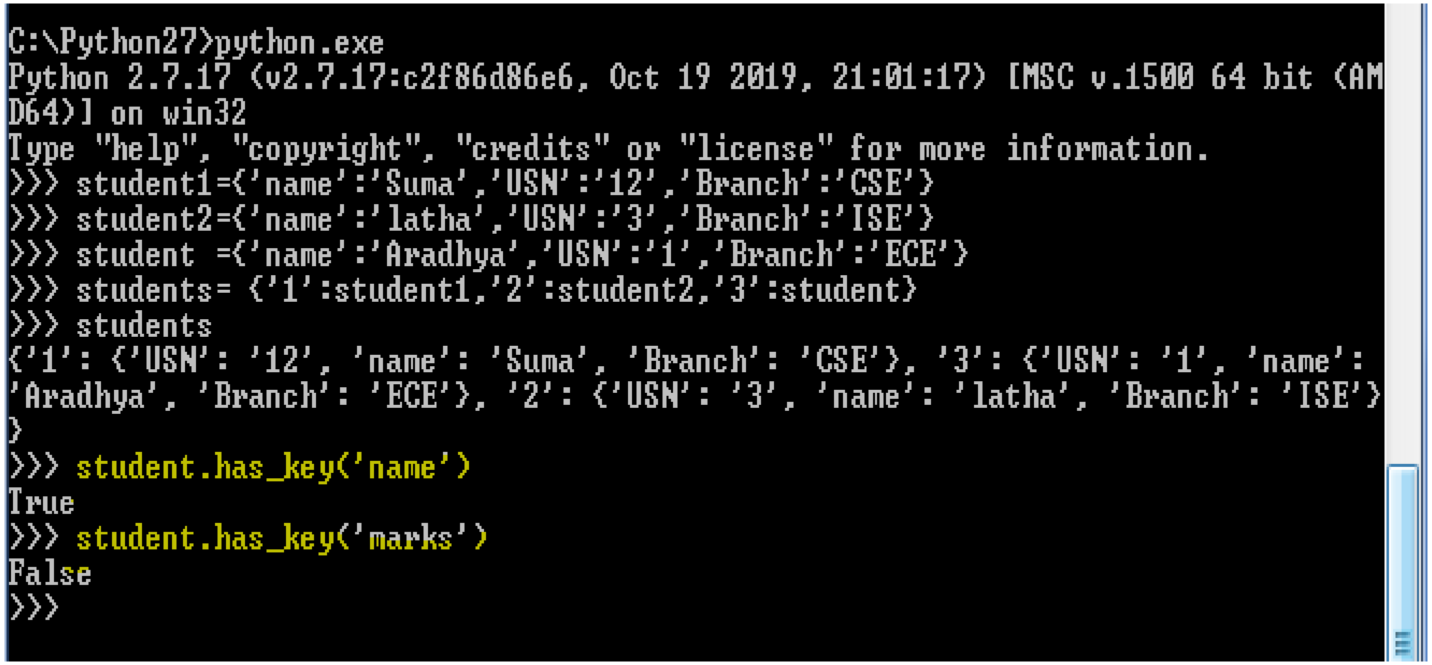


```
C:\Python27>python.exe
Python 2.7.17 (v2.7.17:c2f86d86e6, Oct 19 2019, 21:01:17) [MSC v.1500 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> student1={'name':'Suma','USN':'12','Branch':'CSE'}
>>> student2={'name':'latha','USN':'3','Branch':'ISE'}
>>> student ={'name':'Aradhya','USN':'1','Branch':'ECE'}
>>> students= {'1':student1,'2':student2,'3':student}
>>> students
{'1': {'USN': '12', 'name': 'Suma', 'Branch': 'CSE'}, '3': {'USN': '1', 'name':
'Aradhya', 'Branch': 'ECE'}, '2': {'USN': '3', 'name': 'latha', 'Branch': 'ISE'}
}
>>>
```

Python Data Types and Data Structures:

Working with Dictionaries in Python:

- Few Examples for dictionary contd.:
  - Check if dictionary has a key



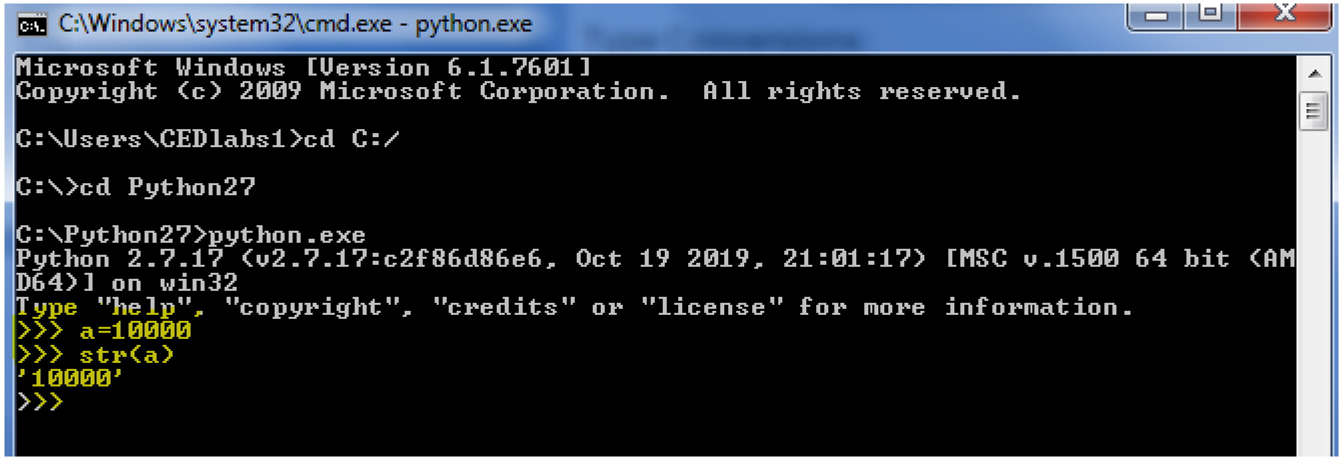
```
C:\Python27>python.exe
Python 2.7.17 (v2.7.17:c2f86d86e6, Oct 19 2019, 21:01:17) [MSC v.1500 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> student1={'name':'Suma','USN':'12','Branch':'CSE'}
>>> student2={'name':'latha','USN':'3','Branch':'ISE'}
>>> student ={'name':'Aradhya','USN':'1','Branch':'ECE'}
>>> students= {'1':student1,'2':student2,'3':student}
>>> students
{'1': {'USN': '12', 'name': 'Suma', 'Branch': 'CSE'}, '3': {'USN': '1', 'name':
'Aradhya', 'Branch': 'ECE'}, '2': {'USN': '3', 'name': 'latha', 'Branch': 'ISE'}
}
>>> student.has_key('name')
True
>>> student.has_key('marks')
False
>>>
```

Python Data Types and Data Structures:

Type Conversions:

➢Few Examples of Type Conversions:

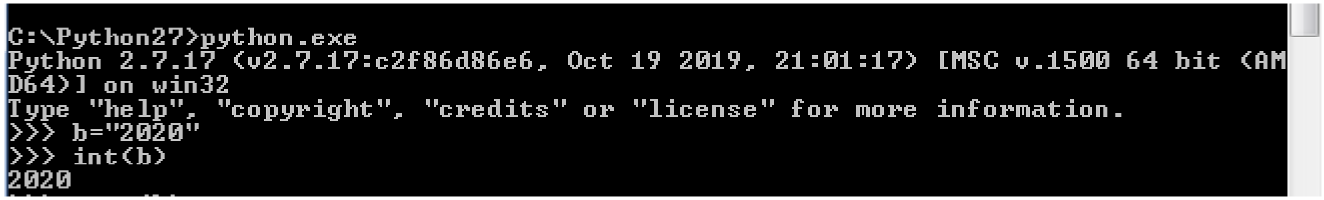
▪Convert to a string:



C:\Windows\system32\cmd.exe - python.exe

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\CEDlabs1>cd C:/

C:\>cd Python27

C:\Python27>python.exe
Python 2.7.17 (v2.7.17:c2f86d86e6, Oct 19 2019, 21:01:17) [MSC v.1500 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> a=10000
>>> str(a)
'10000'
>>>
```

▪Convert to an int:

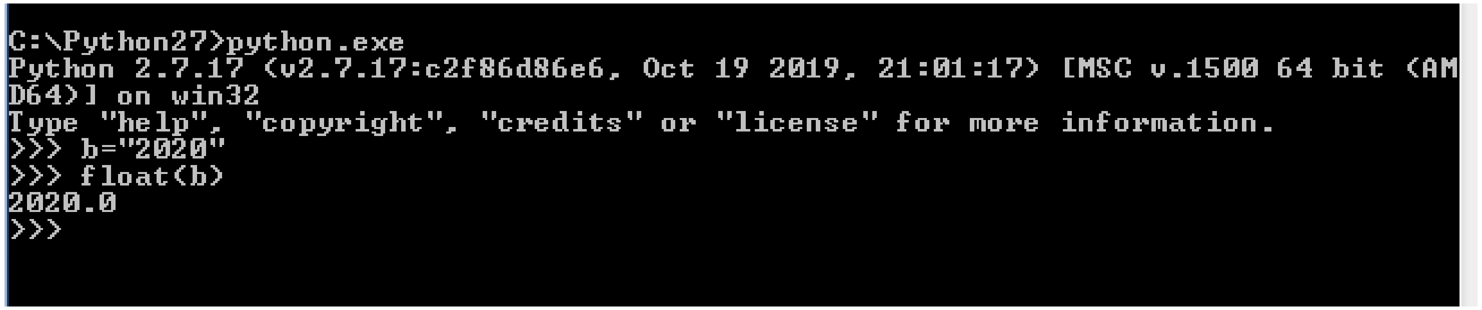


```
C:\Python27>python.exe
Python 2.7.17 (v2.7.17:c2f86d86e6, Oct 19 2019, 21:01:17) [MSC v.1500 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> b="2020"
>>> int(b)
2020
```

Python Data Types and Data Structures:

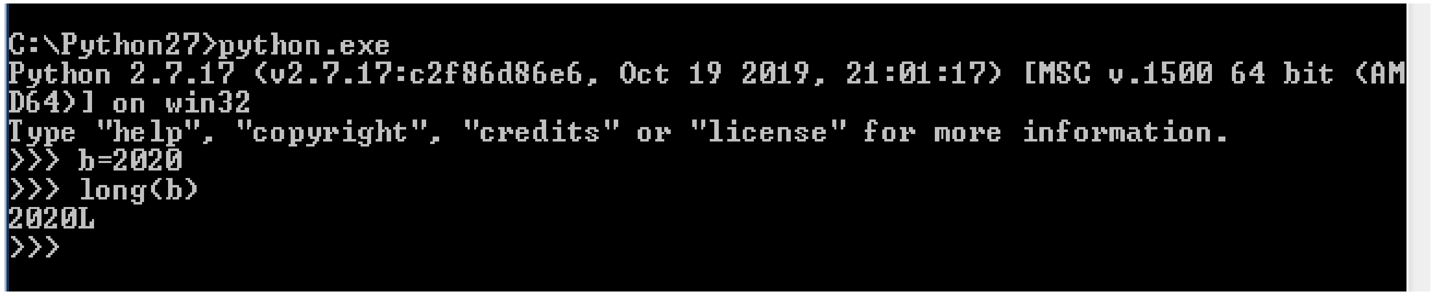
Type Conversions:

- Few Examples of Type Conversions:
  - Convert to a float:



```
C:\Python27>python.exe
Python 2.7.17 (v2.7.17:c2f86d86e6, Oct 19 2019, 21:01:17) [MSC v.1500 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> b="2020"
>>> float(b)
2020.0
>>>
```

  - Convert to long:



```
C:\Python27>python.exe
Python 2.7.17 (v2.7.17:c2f86d86e6, Oct 19 2019, 21:01:17) [MSC v.1500 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> b=2020
>>> long(b)
2020L
>>>
```

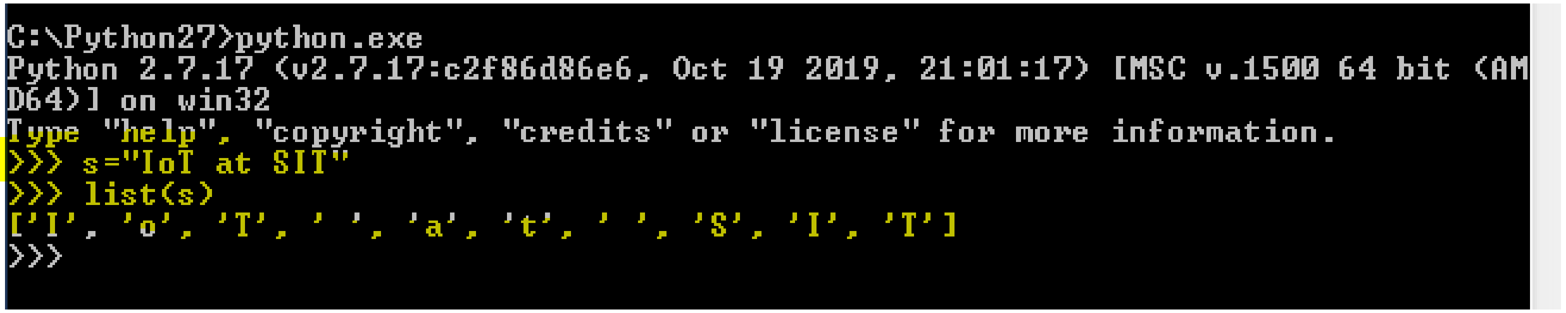
Python Data Types and Data Structures:

Type Conversions:

Siddaganga Institute of Technology

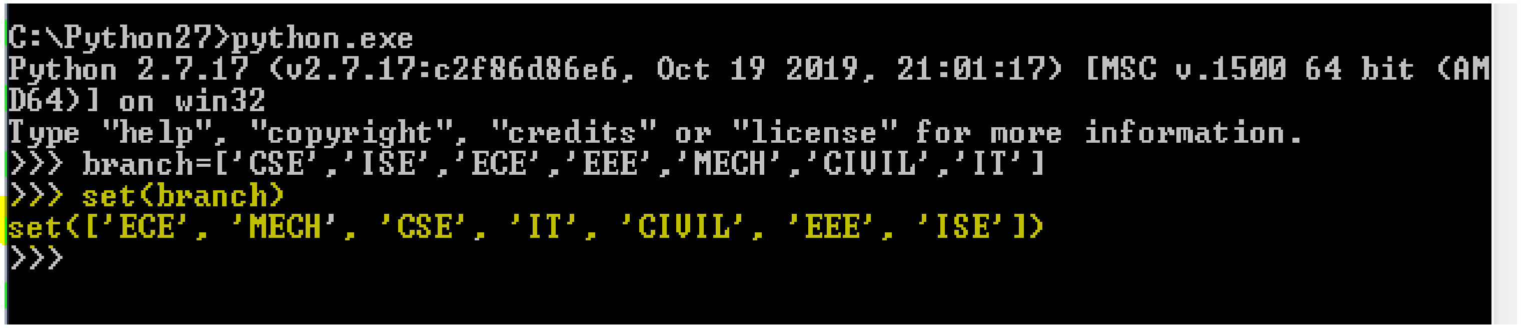
➢Few Examples of Type Conversions:

▪Convert to list:



```
C:\Python27>python.exe
Python 2.7.17 (v2.7.17:c2f86d86e6, Oct 19 2019, 21:01:17) [MSC v.1500 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> s="IoT at SIT"
>>> list(s)
['I', 'o', 'T', ' ', 'a', 't', ' ', 'S', 'I', 'T']
>>>
```

▪Convert to set:



```
C:\Python27>python.exe
Python 2.7.17 (v2.7.17:c2f86d86e6, Oct 19 2019, 21:01:17) [MSC v.1500 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> branch=['CSE','ISE','ECE','EEE','MECH','CIVIL','IT']
>>> set(branch)
set(['ECE', 'MECH', 'CSE', 'IT', 'CIVIL', 'EEE', 'ISE'])
>>>
```

Refer to my next video lecture for the following:

❖Python programming concepts:

- ❑ Control Flow
- ❑ Functions
- ❑ Modules
- ❑ Packages
- ❑ Classes
- ❑ Python packages of Interest for IoT
- ❑ Exercises
- ❑ Other references