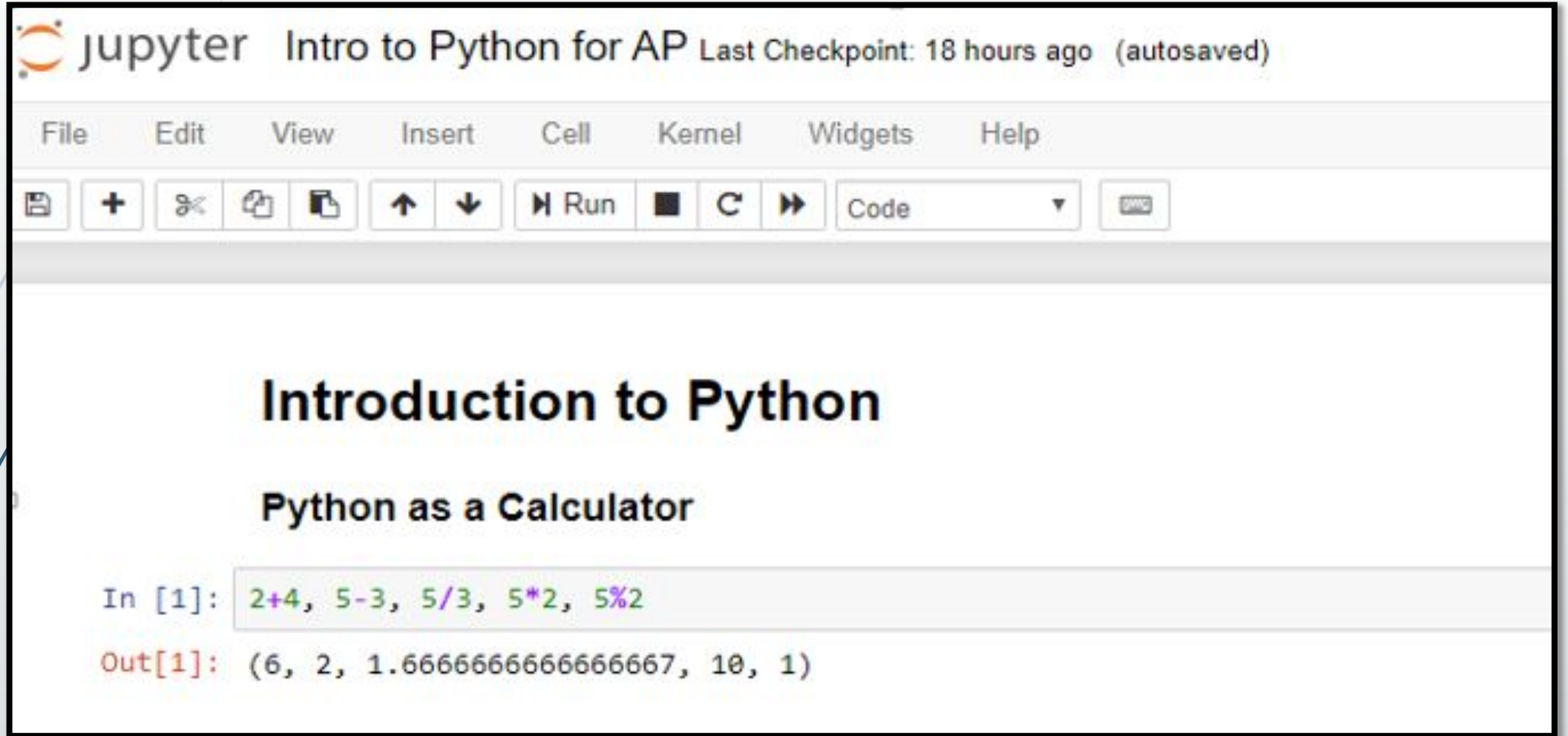# Python for Physics

By

Rabia Tabassum

Assistant Professor

Key topics Covered

- Introduction to Jupiter Notebook .
- Introduction to Basics of Python.
- Python libraries (Numpy and Matplotlib)
- Programming the main topics of Physics:
    - Vectors
    - Motion & Free Fall Motion
    - Projectile motion
    - Simple Harmonic Motion & Damped Oscillation
    - Circular Motion &SHM
    - Wave Motion
    - Electrostatics Force and Field
    - Gravitational Field

Introduction to Python

Variables

```
In [2]: print('Hello world')

        Hello world

In [3]: a = 4
        b= 3.5
        c = 'Physics'
        list = [1,2,3,4]
        print (a,' ,', b , ',', c , ',' ,list)
        print (type(a), type(b), type(c), type(list))

        4  , 3.5 , Physics , [1, 2, 3, 4]
        <class 'int'> <class 'float'> <class 'str'> <class 'list'>
```

Reserved Words

You cannot use reserved words as variable names / identifiers

```
False    class    return is        finally
None     if    for   lambda  continue
True     def  from     while   nonlocal
and   del  global  not  with
as    elif    try     or      yield
assert else    import pass
break    except in       raise
```

String Operations:

```
In [4]: s1 = "Applied"
        s2 = "Physics"
        s1+s2

Out[4]: 'AppliedPhysics'
```

```
In [5]: print(s1 + " " + s2) # for space b/w s1 and s2

        Applied Physics
```

```
In [6]: s1[0] , s1[1]

Out[6]: ('A', 'p')
```

```
In [7]: s1[0:2], s1[3:]

Out[7]: ('Ap', 'lied')
```

String Operations:

```
In [8]: s1[0::+3] , s2[0::+2]

Out[8]: ('Ald', 'Pyis')
```

```
In [9]: s1[::-1] , s2[::-1]

Out[9]: ('deilppA', 'scisyhP')
```

```
In [10]: s3 = 'Applied'
         s1 == s2 , s1 == s3 , s2 ==s3

Out[10]: (False, True, False)
```

Boolean data type

```
In [11]: b1 = True
         b2 = False
         type(b1) , type(b2)

Out[11]: (bool, bool)
```

```
In [12]: zero_int = 0   #An int, float or complex number set to zero returns as False. An integer,
                        #float or complex number set to any other number, positive or negative, returns as True.
         bool(zero_int)

Out[12]: False
```

```
In [13]: pos_int = 1
         f = -0
         neg = -2.3
         bool(pos_int) , bool(s1) , bool(b1), bool(b2), bool(f), bool(neg)

Out[13]: (True, True, True, False, False, True)
```

Boolean data type

```
In [14]: f = 0.0
         fr = 0.22
         bool(f) , bool(fr)
```

Out[14]: (False, True)

```
In [15]: b1 or b2 , b1 and b2 ,  not b1 , b1 == b2 , b1 != b2
```

Out[15]: (True, False, False, False, True)

```
In [16]: name = "Anaya"
         empty = ""

         bool(name),  bool(empty)
```

Out[16]: (True, False)

List

```
In [17]: list1 = ["physics", "Chemistry", "Math", "Statistics"]   # indexing strat from  0 and then , 1, 2, 3
         list1[0] , list1[3], list1[3]
```

```
Out[17]: ('physics', 'Statistics', 'Statistics')
```

```
In [18]: list1[2:] , list1[:2] , list1[:], list1[-3:], list1[:-3]
```

```
Out[18]: (['Math', 'Statistics'],
         ['physics', 'Chemistry'],
         ['physics', 'Chemistry', 'Math', 'Statistics'],
         ['Chemistry', 'Math', 'Statistics'],
         ['physics'])
```

Lists are mutable

```
In [19]: list1[2] = 'Computer Science'
         list1
```

```
Out[19]: ['physics', 'Chemistry', 'Computer Science', 'Statistics']
```

Appending to a list using " append and extend"

```
In [20]: list1.append('Islamiat')
         list1
```

```
Out[20]: ['physics', 'Chemistry', 'Computer Science', 'Statistics', 'Islamiat']
```

```
In [21]: list2 = [1,2,3,4,5]
         list1.extend(list2)
         list1
```

```
Out[21]: ['physics',
          'Chemistry',
          'Computer Science',
          'Statistics',
          'Islamiat',
          1,
          2,
          3,
          4,
          5]
```

Deleting from a list using " remove and pop "

```
In [22]:  list1.remove('Islamiat')
          list1

Out[22]:  ['physics', 'Chemistry', 'Computer Science', 'Statistics', 1, 2, 3, 4, 5]

In [23]:  list1.pop(0)
          list1

Out[23]:  ['Chemistry', 'Computer Science', 'Statistics', 1, 2, 3, 4, 5]
```

Tuples in Python

Tuples are immutable

```
In [24]:  tuple1 = ('AP', 'PF','Eng')
          tuple1[2]

Out[24]:  'Eng'
```