



# Python-Modules and Packages

Dr. Sarwan Singh  
NIELIT Chandigarh

# Agenda

- Modules
- Packages
- Creating and using modules
- Creating and using packages

Artificial Intelligence

Machine Learning

Deep Learning

*"Success is more a  
function of consistent  
common sense than it is  
of genius"*

(An Wang, Computer engineer and  
inventor, 1920 - 1990)

*One guiding principle of Python code is that*  
**"explicit is better than implicit"**

# Python - Modules

*Every file, which has the file extension .py and consists of proper Python code, can be called as a module*



- **module** is a **Python** object with arbitrarily named attributes that programmer can bind and reference.
- **module** is a file consisting of **Python** code.
- **module** can define functions, classes and variables.
- **module** can also include runnable code.
- The objects, i.e. files, classes or attributes contained in a module can be accessed after **import**.

```
import math  
math.pi
```

3.141592653589793



# Package in Python

- **package** is a collection of **Python modules**
- **package** is a directory of **Python modules** containing an additional `__init__.py` file,
- Modules are used to break down large programs into small manageable and organized files.
- modules provide reusability of code.
- Python has a ton of standard modules available.
- User-defined modules can also be imported in same way as Standard modules are imported.

# Import with renaming

- `import math as m`
- `print("The value of pi is", m.pi)`

Import specific names from a module without importing the module as a whole.

- `from math import pi`
- `from math import pi, e`
- `from math import *`

```
print("The value of pi is ", pi)
```

# Creating own modules

- Create a file name myModule.py having

`def hiModule():`

`print('This is Module calling')`

`print('Inside hiModule')`

```
%run myModule.py
```

```
hiModule
```

```
<function __main__.hiModule>
```

```
hiModule()
```

```
This is Module calling  
Inside hiModule
```



# Open new notebook

- Try using hiModule()  
error is generated

```
import myModule  
myModule.hiModule()
```

Another way to use function :

```
import myModule import hiModule  
hiModule()
```

```
hiModule()
```

```
-----  
NameError                                Traceback (most recent call last)  
<ipython-input-1-4d63db864d68> in <module>()  
----> 1 hiModule()
```

```
NameError: name 'hiModule' is not defined
```

```
import myModule  
myModule.hiModule()
```

```
This is Module calling  
Inside hiModule
```

```
from myModule import hiModule  
hiModule()
```

```
This is Module calling  
Inside hiModule
```

# Creating own package

- Create a directory having new package's name.
- Put all classes in it.
- Create a **`__init__.py`** file in the directory. This is important to differentiate package directory from the ordinary directories.
- import statements to *import* classes from newly created package are written in this file

Animals directory contains

```
__init__.py
Birds.py
Mammals.py
test.py
```





- Birds.py file having Birds class
- Mammals.py file having Mammals class



\_\_init\_\_.py file contains

from Mammals import Mammals  
from Birds import Birds

```
class Birds:
```

```
    def __init__(self):  
        ''' Constructor for this class. '''  
        # Create some member animals  
        self.members = ['Sparrow', 'Robin', 'Duck']  
  
    def printMembers(self):  
        print('Printing members of the Birds class')  
        for member in self.members:  
            print('\t%s ' % member)
```

```
class Mammals:
```

```
    def __init__(self):  
        ''' Constructor for this class. '''  
        # Create some member animals  
        self.members = ['Tiger', 'Elephant', 'Wild Cat']  
  
    def printMembers(self):  
        print('Printing members of the Mammals class')  
        for member in self.members:  
            print('\t%s ' % member)
```

# Using classes from package

```
import Animals
#import classes from new Package
from Animals import Mammals
from Animals import Birds
newMammal = Mammals()
newBird = Birds()
newMammal.printMembers()
newBird.printMembers()
```

Printing members of the Mammals class

Tiger  
Elephant  
Wild Cat

Printing members of the Birds class

Sparrow  
Robin  
Duck

```
%run Animals/test.py
```

Printing members of the Mammals class

Tiger  
Elephant  
Wild Cat

Printing members of the Birds class

Sparrow  
Robin  
Duck

# Jupyter Notebook Link

- <https://colab.research.google.com/drive/16QAIOODvsa6C17bvjtn7MxAR63d6a4y-?usp=sharing>

# Github Repository Link

- <https://github.com/sarwansingh/Python/tree/master/ORD>

