

DOC STRING

Docstrings provide a convenient way of associating documentation with functions, classes, or modules.

```
In [1]: def my_function(a,b):  
        '''This function about adding two numbers'''  
        return a+b  
        my_function(10,5)
```

Out[1]: 15

```
In [2]: my_function.__doc__
```

Out[2]: 'This function about adding two numbers'

DATA TYPES

```
In [3]: # Numeric  
        int_=10  
        float_=2.4  
        complex_=3+8j  
  
        # Sequence  
        list_[1,2,3,4]  
        tuple_=(1,2,3,4)  
  
        #Mapping  
        dict_={'a':1,'b':2}  
  
        # set  
        set_={1,2,3,4}  
  
        #bool  
        bool_=True  
  
        # string  
        string_="Hello"
```

CONDITIONAL AND LOOP STATEMENT

```
In [4]: # if .. else  
        if(True):  
            print("hi")  
        else:  
            print("hello")  
        # short hand    print("A") if a > b else print("B")
```

hi

```
In [5]: # if ... elif ... else
flag=1
if(flag==0):
    print("if")
elif(flag==1):
    print("elif")
else:
    print("else")
```

elif

```
In [6]: #Nested if
flag =1
if(flag>0):
    if(flag==1):
        print("inside if")
    else:
        print("inside else")
else:
    pass
```

inside if

```
In [7]: # Looping
# for and while
i = 1
while i < 5:
    print(i)
    i += 1
else:
    print("i is no longer less than 5")
```

1
2
3
4
i is no longer less than 5

```
In [8]: # break
i = 1
while i < 5:
    print(i)
    i += 1
    if(i==3):
        break

else:
    print("i is no longer less than 5")
```

1
2

```
In [9]: # continue
# break
i = 1
while i < 5:
    print(i)
    i += 1
    if(i==3):
        continue

else:
    print("i is no longer less than 5")
```

```
1
2
3
4
i is no longer less than 5
```

```
In [10]: # for loop
for x in range(5):
    print(x, end=" ")
```

```
0 1 2 3 4
```