

Python Branching

Panchatcharam Mariappan

Assistant Professor

**Department of Mathematics and Statistics,
IIT Tirupati**

MORE ON LISTS

```
x=[1,2,3,4,5]
print('First Element: ',x[0]) #indexing
print('First Element: ',x[1])
print('Last Element: ',x[-1]) #indexing from last
print('Slicing: ',x[2:]) #slicing
print('Slicing 2: ',x[-2:]) #slicing
x=x+[2,3,4,5,6] #concatenation
print('Concatenation:',x)
x.append(7) #appending
print('Appended 7: ',x)
x.remove(2) #removing
print('Removed 2: ',x)
x[2]=7 #replacing
print('Replacment: ',x)
```

```
y=[0,0,0] #nested list
y[0]=[1,2,3] #replacing
y[1]=[3,4,5]
y[2]=[5,6,7]
print('Nested List :',y)
#can be mixture of all data types
A=['a','b','c','d','e','f',0,0.1,"Ram",True,3+4j,[1,2,3,4]]
print('Mixtures: ',A)
print('Slicing Again: ',A[2:7]) #slicing
#Remove item
A[2:6]=[]
print('Removing again: ',A)
del A[2]
print('Deleting: ',A)
#Remove all
A[:]=[]
print('Removed Everything: ',A)
```

```
#Length
A=['a','b','c','d','e','f']
print('A*2:',A*2)
print('New List:',A)
print('Length of List: ',len(A))
A.clear()
print('Cleared Again: ',A)
A=['a','b','c','d','e','f']
A.extend('g')
print('Extended',A)
```

```
x=x+[2,3,4,5,6]
print(x)
print('Number of occurrences of 3: ',x.count(3))
#number of occurrences
print('Index of 5 in the list: ',x.index(5))
print('Maximum and Minimum: ',max(x),min(x))
#minimum and maximum
x.reverse()
print('Reversed: ',x)
x.sort()
print('Sorted: ',x)
reversed(x)
print(x)
```

MORE ON TUPLES

```
heros=('Arthos','Porthos','Aramis','Romeo','Juliet')  
print(heros)  
print(len(heros))  
print(heros[1],heros[2],heros[-1],heros[-  
2],heros[2:],heros[-2:],sep=' & ')  
print(heros.index('Porthos'))
```

MORE ON SETS


```
emptyset=set()  
print(emptyset)  
print(x)  
numbers=set(x)  
print(numbers)  
y=list(numbers)  
print(y)
```

```
programming=set(['C','C++','Python','Ruby','Java','S  
cala','Swift','Perl'])  
print(programming)  
programming.add('Python')  
print(programming)  
programming.add('SQL')  
print(programming)  
compilers=set(['C','C++','Scala'])  
interpreters=set(['Python','Java'])  
programming.update(compilers)
```

```
intersect=compilers.intersection(interpreters)
print(intersect)
union=compilers.union(interpreters)
print(union)
union=union.union(programming)
print(union)
diff=programming.difference(compilers)
print(diff)
print(compilers.isdisjoint(interpreters))
print(compilers.issubset(programming))
print(programming.issuperset(compilers))
print(programming^compilers)#symmetric difference
print(programming.symmetric_difference(compilers))
```

MORE ON DICTIONARIES

```
course={'MA6024':'PDE','MA5191':'Programming','MA5023':'DE',  
        'MA6204':'Numerical'}  
print(course)  
print(course['MA6204'])  
print(course.get('MA5191'))  
print(course.keys())  
print(course.values())  
print(course.items())
```

```
course['MA1101']='Calculus'  
print(course)  
print(len(course))  
course.clear()  
print('cleared: ',course)
```

```
btechcourse={'MA2021':'Linear Algebra','MA2022':'Complex  
Methods','MA2023':'Probability'}  
print(btechcourse)  
course.update(btechcourse)  
print(course)  
herodictionary=dict.fromkeys(heros)  
print(herodictionary)  
herodictionary=dict.fromkeys(heros,[1,2,3,4,5])  
print(herodictionary)
```

- Write a Python program to display the first and last items from the following list.
 - **Unit Test:**
fruits=['Apple','Orange','Grapes','Banana','Papaya']
 - Write a python program to display the length of this list
 - Output: 5
- Create a list fruits as given above
 - Append the fruit 'kiwi' and 'lemon' to fruits list
 - Print the list again
 - Remove the 'Grapes' from the list
 - Sort the fruits in ascending order
 - Sort the fruits in descending order

- Write a Python program to create two sets called Bollywood and Hollywood movies
 - **Unit Test:**
 - hollywood={'Avatar', 'Avengers', 'Starwars', 'Gravity', 'Titanic', 'JurassicPark'}
 - Bollywood={'PK', '3idiots', 'Dhoom', 'chole', 'Dil Se'}
- Write a python program to display the length of the set
- Update another five movies in bollywood
- Update 'slum dog millionaire' in Bollywood and Hollywood
 - Find a similar movie and update in both
- Find the intersection of Hollywood and Bollywood
- Find the union, set difference, symmetric difference

- Write a Python program to create dictionary called fruits_price
 - **Unit Test:**
fruits_price={'Apple':150,'Orange':80,'Grapes':60,'Banana':40,'Papaya':100}
- Write a python program to display the length of the dictionary
- Create a list fruits as given above
 - Get at least another five fruit names from the user and its unit price, update the values in the list
 - Display the list of fruits available.
 - Get the number of fruits required by the customer and show its total price with 18%.

BRANCHING

- **if statement:** Executes a group of statements only if a certain condition is true. Otherwise, the statements are skipped.

- Syntax:

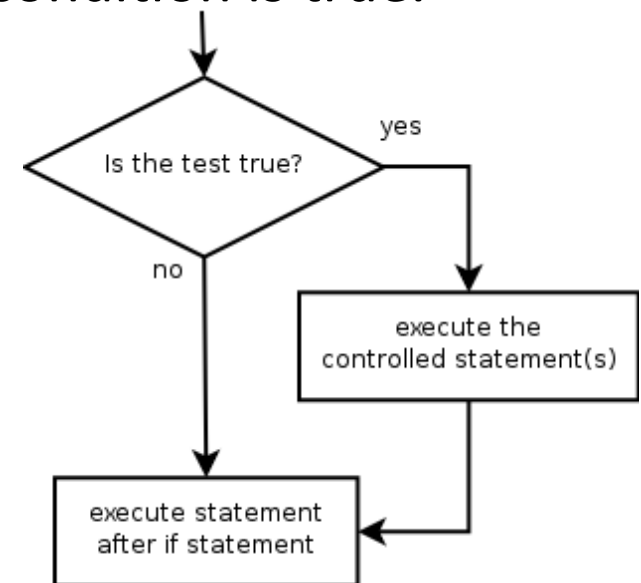
```
if condition :  
    statements
```

- Example:

```
gpa = 3.4
```

```
if gpa > 2.0:
```

```
    print("Your application is accepted.")
```



- **if/else statement:** Executes one block of statements if a certain condition is True, and a second block of statements if it is False.

- Syntax:

```
if condition:  
    statements
```

```
else:  
    statements
```

- Example:

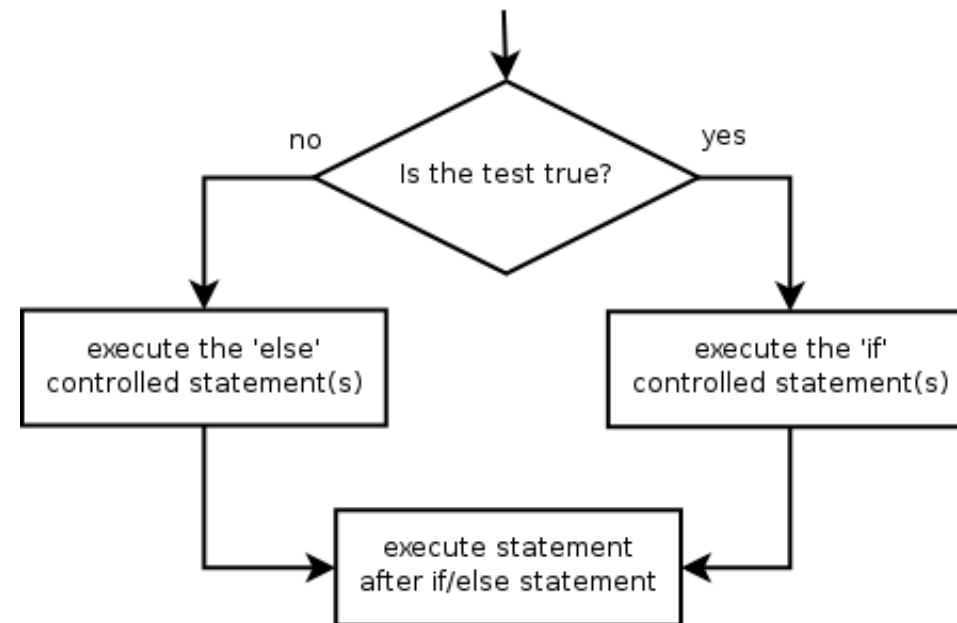
```
gpa = 8.4
```

```
if gpa > 7.0:
```

```
    print("Welcome to Kathmandu University!")
```

```
else:
```

```
    print("Your application is denied.")
```

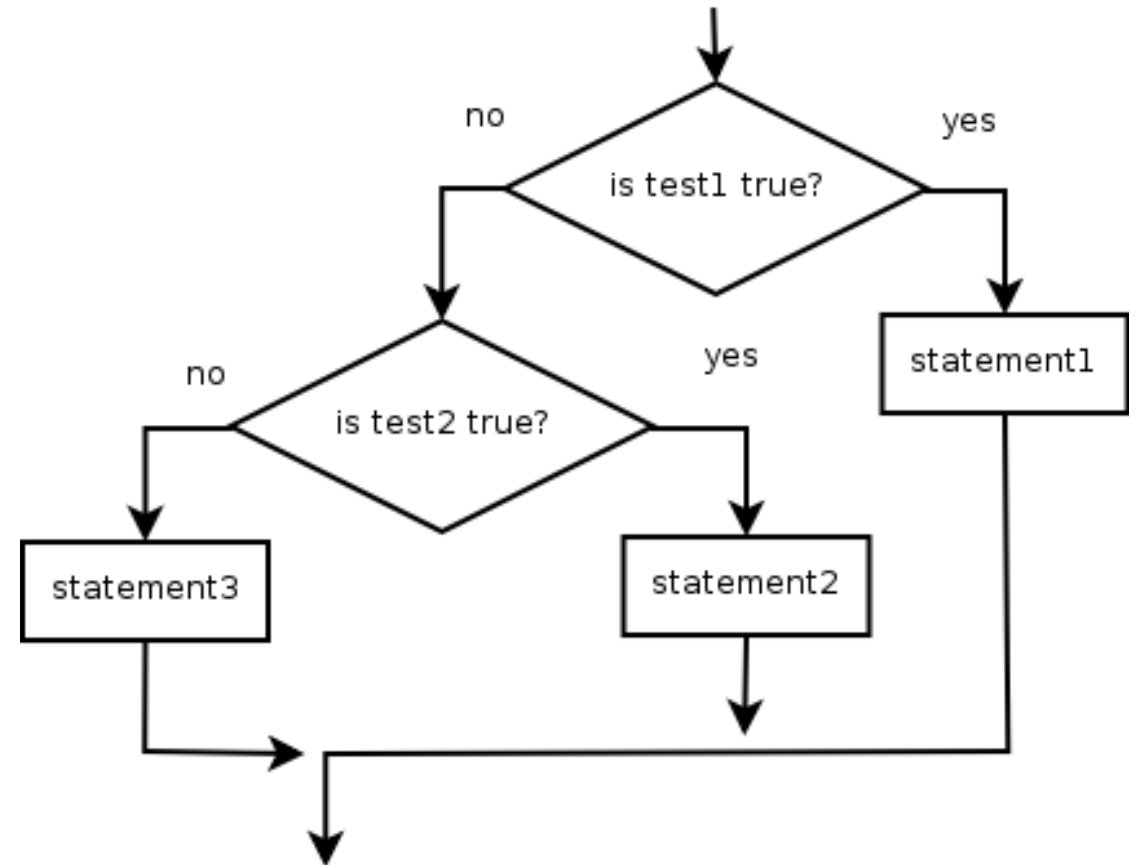


- Multiple conditions can be chained with `elif` ("else if"):

`if condition :`
 `statements`

`elif condition :`
 `statements`

`else :`
 `statements`



Operator	Meaning	Example	Result
==	equals	$1 + 1 == 2$	True
!=	does not equal	$3.2 != 2.5$	True
<	less than	$10 < 5$	False
>	greater than	$10 > 5$	True
<=	less than or equal to	$126 <= 100$	False
>=	greater than or equal to	$5.0 >= 5.0$	True

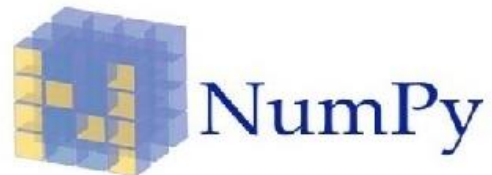
Operator	Example	Result
and	$9 \neq 6$ and $2 < 3$	True
or	$2 == 3$ or $-1 < 5$	True
not	not $7 > 0$	False

- For given two integers, find which one is smaller than the other
- Calculate the smallest of three integers

- Get four real numbers, find the smallest and the greatest of those four numbers. Calculate the difference between the smallest and the greatest.



End of Python Branching



IP[y]:
IPython

