

**Python (Ranges & Sets)**

Range is a kind of data type in python which is an immutable. Range will be used in for loops for number of iterations. Range is a constructor which takes arguments and those must be integers. Below is the syntax.

**Syntax:**

```
class range(stop)
class range(start, stop[, step])
```

stop: the value of stop parameter.

step: the value of step parameter. If the value is omitted, it defaults to 1.

start: the value of start parameter. If the value is omitted, it defaults to zero.

**Examples:**

```
>>>list(range(5))
[0, 1, 2, 3, 4]
>>>list(range(10,20))
[10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
>>>list(range(10,20,5))
[10, 15]
>>>list(range(10,20,2))
[10, 12, 14, 16, 18]
# It will not take the float numbers
>>>list(range(0,0.1))
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'float' object cannot be interpreted as an integer
```

```
>>>list(range(0,2))  
[0, 1]  
>>>list(range(0,1))  
[0]  
>>>list(range(0,10,5))  
[0, 5]
```

### Python – Sets

In this tutorial, we will learn about sets in python. A set is a datatype which holds an unordered collection with immutable and no duplicate elements. By the name, Set can be used for various mathematical operations. Mathematical operation may be union, intersection or difference, etc. Let us see the example of using the Set below.

#### Example:

```
>>> set1={'html','c','java','python','sql'}  
>>>print(set1)  
{'c', 'python', 'sql', 'html', 'java'}  
  
# Below we have given duplicates  
>>> set1={'html','c','java','python','sql','java'}  
  
# we can observe that duplicates are ignored  
>>>print(set1)  
{'c', 'python', 'java', 'html', 'sql'}  
>>> set1  
{'c', 'python', 'java', 'html', 'sql'}
```



**Membership testing in Sets:**

```
>>> set1={'html','java','python','sql','java'}
>>> set1
{'python', 'java', 'html', 'sql'}
>>> print(set1)
{'python', 'java', 'html', 'sql'}
>>> 'c' in set1
False
>>> 'java' in set1
True
```

**Sets Operations in Python:**

```
>>> set1={'html','java','python','sql','java'}
>>> set2={'html','oracle','ruby'}
# Unique words in set1
>>> set1
{'python', 'java', 'html', 'sql'}
>>> set2
{'ruby', 'html', 'oracle'}
# words in set1 but not in set2
>>> set1-set2
{'python', 'java', 'sql'}
# Words in set1 or set2 or both
>>> set1 | set2
{'ruby', 'html', 'oracle', 'python', 'java', 'sql'}
```

```
# Words in both set1 and set2
>>> set1 & set2
{'html'}

# Words in set1 or set2 but not both
>>> set1 ^ set2
{'oracle', 'python', 'sql', 'ruby', 'java'}
```

### Python - Operators

Operators in Python helps us to perform the mathematical operations with numbers. There are different operators in Python as below.

Operators	Description
//	Integer division
+	addition
-	subtraction
*	multiplication
/	Float division
%	Provide remainder after division (Modulus)
**	Perform exponent (raise to power)

Let us try implementing every operator now.

#### 1. Addition: symbol used (+)

```
>>> 10+10
20
>>> 20+30
50
```



```
>>>50+50
```

```
100
```

**2. Substraction: symbol used (-)**

```
>>>20-10
```

```
10
```

```
>>>50-40
```

```
10
```

```
>>>100-30
```

```
70
```

**3. multiplication: Symbol used (\*)**

```
>>>5*2
```

```
10
```

```
>>>10*2
```

```
20
```

```
>>>20*2
```

```
40
```

**4. Float Division: This will divide and provide the result in floating value and the symbol used (/)**

```
>>>5/2
```

```
2.5
```

```
>>>10/2
```

```
5.0
```

**5. Integer Division: This will divide and truncate the decimal and provide the Integer value and the symbol used (//)**

```
>>>5//2
```

```
2
```

```
>>>7//2
```

```
3
```

**6. Exponentiation Operator:** This will help us to calculate a power b and return the result

```
>>>10**3 # This mean 10*10*10
```

```
1000
```

**7. Modulus Operator:** This will provide the remainder after the calucation and symbol used (%)

```
>>>10%3
```

```
1
```

What if we want to work with multiple operators at a time. Here comes the Operator precedence in Python.

Operator	Description
lambda	Lambda expression
if – else	Conditional expression
or	Boolean OR
and	Boolean AND
not x	Boolean NOT
in, not in, is, is not, <, <=, >, >=, !=, ==	Comparisons, including membership tests and identity tests
	Bitwise OR
^	Bitwise XOR
&	Bitwise AND
<<, >>	Shifts
+, -	Addition and subtraction



<code>*, @, /, //, %</code>	Multiplication, matrix multiplication division, remainder [5]
<code>+x, -x, ~x</code>	Positive, negative, bitwise NOT
<code>**</code>	Exponentiation [6]
<code>await x</code>	Await expression
<code>x[index], x[index:index], x(arguments...), x.attribute</code>	Subscription, slicing, call, attribute reference
<code>(expressions...), [expressions...], {key: valu e...}, {expressions...}</code>	Binding or tuple display, list display, dictionary display, set display

### Our Other Courses:

1. Artificial Intelligence
2. Data Science
3. Machine Learning
4. DevOps
5. AWS / Azure
6. IOT
7. Selenium
8. ETL Testing
9. Webservices Testing
10. Manual Testing
11. Blueprism
12. Uipath

**For Any Queries Fell free to Call: 7730997544 (Teja)**