



Python – Math Functions

Function	Description	Example
<code>max(x1, x2,...)</code>	The largest of its arguments: the value closest to positive infinity.	<code>max(80, 100, 1000) : 1000</code>
<code>min(x1, x2,...)</code>	The smallest of its arguments: the value closest to negative infinity.	<code>min(80, 100, 1000) : 80</code>
<code>pow(x, y)</code>	The value of $x^{**}y$.	Import math <code>math.pow(100, 2) : 10000.0</code>
<code>round(x [,n])</code>	<code>x</code> rounded to <code>n</code> digits from the decimal point. Python rounds away from zero as a tie-breaker: <code>round(0.5)</code> is <code>1.0</code> and <code>round(-0.5)</code> is <code>-1.0</code> .	



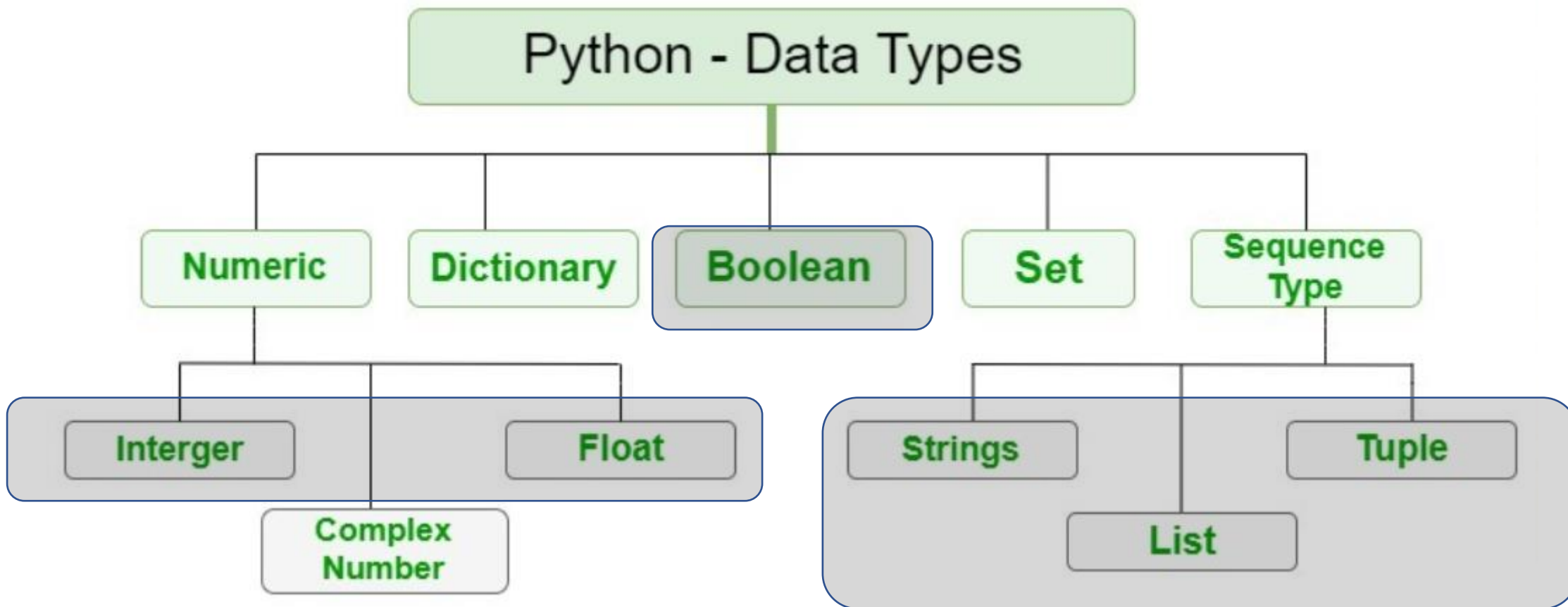
Python – Math Functions

Function	Description	Example
<code>sqrt(x)</code>	The square root of x for <code>x > 0</code> .	<code>math.sqrt(100)</code> : 10.0
<code>exp(x)</code>	The <code>exp()</code> method returns exponential of x: ex.	<code>exp(-45.17)</code> : 2.4150062132629406e-20
<code>abs(x)</code>	This method returns the absolute value of x.	<code>abs(-45)</code>
<code>ceil(x)</code>	This method returns the smallest integer not less than x.	<code>math.ceil(-45.17)</code> : -45 <i>+45.17 → 46 next digit</i>



Python Data Types

Following are the standard or built-in data type of Python:





Python – Lists

A list contains items separated by commas and enclosed within square brackets ([]). To some extent, lists are similar to arrays in C. One of the differences between them is that all the items belonging to a list can be of different data type.

```
list = [ 'abcd', 786 , 2.23, 'john', 70.2 ]
tinylist = [123, 'john']
print (list) # Prints complete list
print (list[0]) # Prints first element of the list
print (list[1:3]) # Prints elements starting from 2nd till 3rd
print (list[2:]) # Prints elements starting from 3rd element
print (tinylist * 2) # Prints list two times
print (list + tinylist) # Prints concatenated lists
```



Python – List Slicing and Indexing

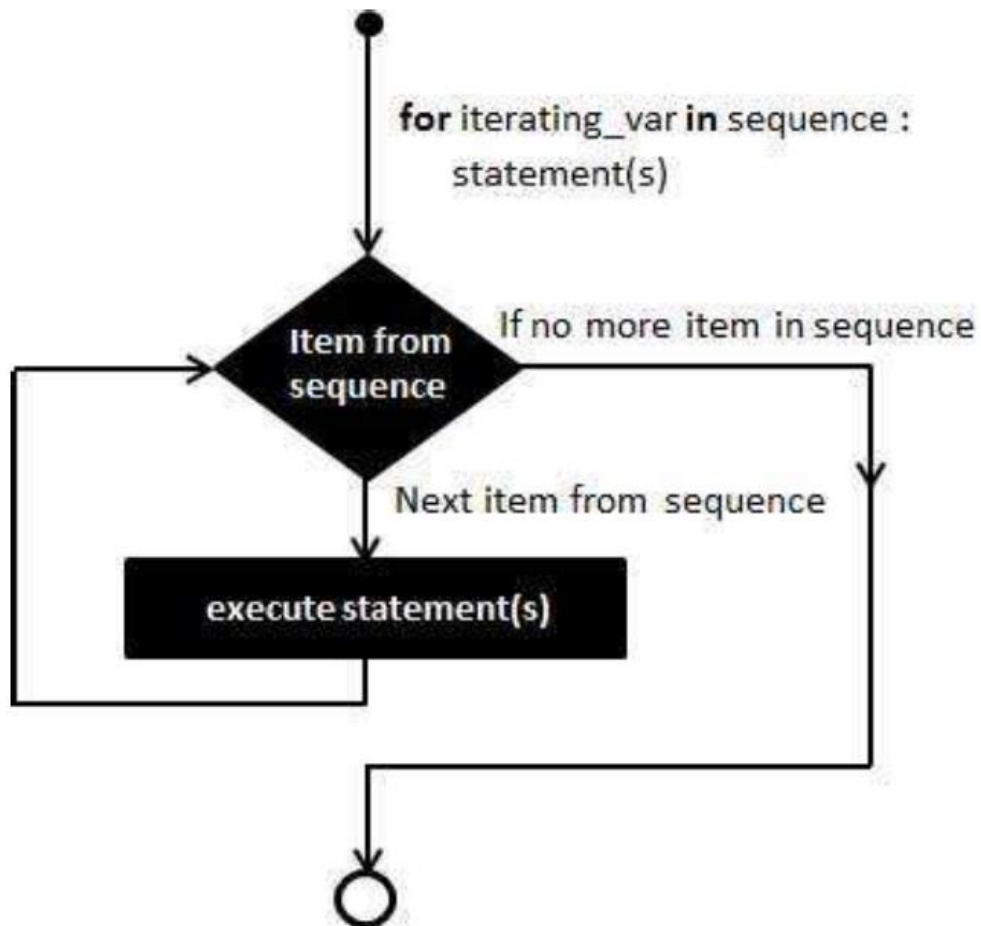
0 1 2
L = ['C++', 'Java', 'Python']
-3 -2 -1

Python Expression	Results	Description
L[2]	'Python'	Offsets start at zero
L[-2]	'Java'	Negative: count from the right
L[1:]	['Java', 'Python']	Slicing fetches sections



Python – for Loop

The **for** statement in Python has the ability to **iterate over the items of any sequence**, such as a **list or a string**.



for iterating_var **in** sequence:
statements(s)