Python Programming Basics

Numeric Functions and Operators in Python

Contents

1. Introduction to Functions
2. General Functions
3. Introduction to Operators
4. Arithmetic Operators
5. Relational Operators
6. Miscellaneous Operators

General Functions

Note: ‘x’ here is a numeral or a vector of numerals.

```
#Absolute value of ‘x’
abs(-4)
4
x = [-4,4.5,-10.5,6]
[abs(i) for i in x]
[4, 4.5, 10.5, 6]`
```

```
#Square Root of ‘x’
import math
math.sqrt(81)
9.0
```

```
#Rounds to the nearest integer that’s larger than x
math.ceil(445.67)
446
```

```
#Rounds to the nearest integer that’s smaller than x
math.floor(445.67)
445
```

General Functions

Note: ‘x’ here is a numeral or a vector of numerals.

```
#Rounds to the nearest integer toward 0.
math.trunc(445.67)
445
```

```
#Rounds to the nearest possible value after mentioning how many digits
#to keep after decimal point.
```

```
round(44.5682,2)
44.57
```

General Functions

Note: ‘x’ here is a numeral or a vector of numerals.

```
#Computes natural logarithms.
import numpy as np
np.log(50)
3.912023005428146
np.log([44,55])
array([3.78418963, 4.00733319])
```

```
math.log(50)
3.912023005428146
```

Natural log can be calculated using **numpy** as well as **math** library.

```
#Computes binary (base 2) logarithm.
math.log(8,2)
3.0
```

```
#Computes logarithm to the base 10.
math.log10(55)
1.7403626894942439
```

```
#Computes the exponential value , ex.
math.exp(6)
403.4287934927351
```

Operators

- Assignment Operators
- Arithmetic Operators
- Relational Operators
- Miscellaneous Operators

Arithmetic Operators

```
x = 7
y = 18
```

```
#Addition
x+y
25
```

```
#Subtraction
x-y
-11
```

```
#Multiplication
x*y
126
```

Arithmetic Operators

```
#Division
y/x
2.5714285714285716
```

```
#Exponentiation
y**x
612220032
```

```
#Integer Division to get Remainder
y%x
4
```

```
#Integer Division to get Quotient
y//x
2
```

Relational Operators

These operators are used to compare values. The result of the comparison is the Boolean (True or False) value. Following table shows the relational operators available in Python

```
x = 7
y = 18
```

```
#Less than
x<y
True
```

```
#Greater than
x>y
False
```

```
#Less than or equal to
x<=5
False
```

Relational Operators

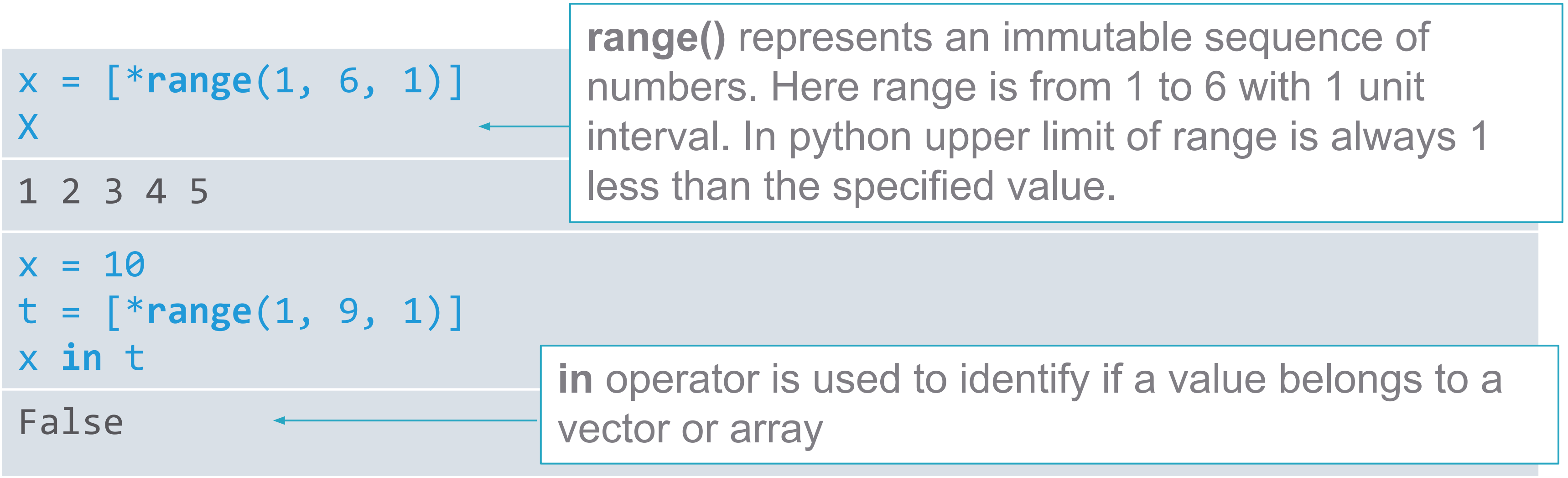
```
#Greater than or equal to
y>=20
False
```

```
#Equal to
y==16
False
```

```
#Not equal to
x!=5
True
```

Miscellaneous Operators

These operators are used for specific purpose and not general mathematical operations.



```
x = [*range(1, 6, 1)]
X
```

```
1 2 3 4 5
```

range() represents an immutable sequence of numbers. Here range is from 1 to 6 with 1 unit interval. In python upper limit of range is always 1 less than the specified value.

```
x = 10
t = [*range(1, 9, 1)]
x in t
```

```
False
```

in operator is used to identify if a value belongs to a vector or array

Quick Recap

In this session, we learnt different types of Functions and Operators in Python. Here is a quick recap:

General Functions	• `abs(), sqrt() ceil(), floor(), trunc(), log(), log2(), log10(), exp()`
Operators	• Assignment Operators: `=` • Arithmetic Operators: `+, -, *, /, ^, %, //` • Relational Operators: `<, >, <=, >=, ==, !=` • Miscellaneous Operators: `in`