



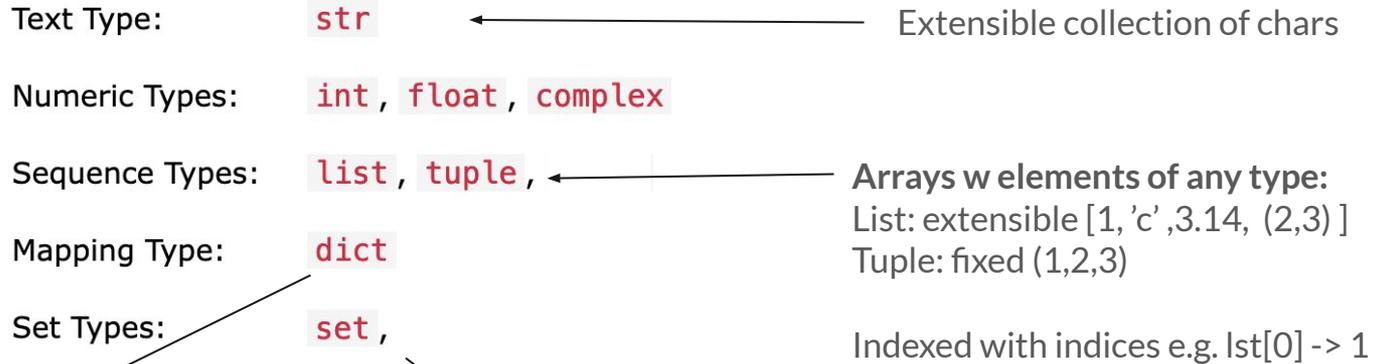
MS&E 233 Python Tutorial

Gordon Martinez-Piedra
4/5/2024

Python Basics



Basic Python Types



Key Value mapping key and value can be any type:

```
map = {'a': 1, 'b': 2}
Indexed with keys e.g. map['a'] -> 1
```

Set with only unique value i.e. if 'a' is already in set and you insert it, set remains unchanged.
Can't index



for loops

Note: iterable items share a lot of semantics e.g. indexing and lengths.

```
# Iterable Items are basically anything with multiple values:  
# string, list, tuple, string, set, dictionary
```

```
numbers = [1,2,3,4,5]  
for x in numbers:  
    # Do something with element x  
    print(x)
```

```
for i in range(len(numbers)):  
    # Do something with index i  
    print(numbers[i])
```

```
.....
```

Output of both:

```
1  
2  
3  
4  
5
```

```
.....
```



Reference

<https://www.w3schools.com/python/default.asp>

Teaches you about each data type and operations you can do e..g map insertion

Teaches you different semantics e..g *for* loops or *while* loops (I imagine you'll only be using *for* loops in this class).

```
map = dict()
print(map)
map['a'] = 1
print(map)
```

```
"""
```

```
Output
```

```
{}
```

```
{'a' : 1}
```

```
"""
```

Numpy Basics



What is numpy?

“**NumPy** is the fundamental package for N-dimensional arrays, mathematical functions, and numerical computing with Python. It is open source, interoperable, performant, and easy to use, and supports a wide range of domains and applications.”

- <https://numpy.org>





Using packages

```
import numpy as np  
  
np.array([1,2,3,4,5])
```



Creating n-dimensional arrays

```
# Each sub list is a row  
twoByTwo = np.array([[1,2], [3,4]])  
# Access with typical matrix notation a(i,j) = ith row, jth column e.g.  
print(twoByTwo[0][1])  
print(twoByTwo[1][0])
```

.....

Output

2

3

.....

N-dimensional arrays

```
N, M, O = 3, 4, 5
```

```
# Create a vector size N all zeros
```

```
zeroVec = np.zeros((N))
```

```
print(zeroVec)
```

```
# Create N x M matrix all initialized to 1
```

```
ones = np.ones((N, M))
```

```
print(ones)
```

```
# Create a N x M x O matrix all initialized to 0
```

```
zeros = np.zeros((N, M, O))
```

```
print(zeros)
```

```
"""
```

```
Output:
```

```
[0., 0., 0.]
```

```
[[1. 1. 1. 1.]
```

```
[1. 1. 1. 1.]
```

```
[1. 1. 1. 1.]]
```

```
[[[0. 0. 0. 0. 0.]
```

```
[0. 0. 0. 0. 0.]
```

```
[0. 0. 0. 0. 0.]
```

```
[0. 0. 0. 0. 0.]]
```

```
[[0. 0. 0. 0. 0.]
```

```
[0. 0. 0. 0. 0.]
```

```
[0. 0. 0. 0. 0.]
```

```
[0. 0. 0. 0. 0.]]
```

```
[[0. 0. 0. 0. 0.]
```

```
[0. 0. 0. 0. 0.]
```

```
[0. 0. 0. 0. 0.]
```

```
[0. 0. 0. 0. 0.]]]
```



Matrix Operations

```
vec1 = np.array([1,2])
vec2 = np.array([3,4])

# Matrix elemnt wise operations
print (vec1 * 2)
print(vec1 * vec2)

# Dot product
vec3 = vec1.dot(vec2)
print(vec3)
```

```
[2., 4.]
[3., 8.]
11
```

```
# Matrix multiplication
N = 2
ones = np.ones((N, N))
matmul = ones @ vec1
print(matmul)

# Matrix transpose
A = np.array([[1,2], [3,4]])
print(A.shape)
transpose = A.T
print(transpose)
```

```
[3., 3.]
(2,2)

[[1,3],
 [2, 4]]
''''''
```



1-Dim Array Slicing

```
# 1D Slicing  
vec = np.array([1,2,3])  
  
# Uses [a,b) - inclusive of first index, exclusive of second  
vec[0:2]  
# [1,2]  
  
# Blank is used as beginning or end of matrix  
vec[:2]  
# [1,2]  
vec[:]  
# [1,2,3]  
  
# negative indexing refers to index N - i where N is size of array  
vec[:-1]  
# [1,2]
```



N-Dim Array Slicing

```
A = np.array([[1,2], [3,4]])  
A[:, 0]
```

[1,3]



Non-exhaustive ideas of other functions

```
# Find the maximum  
np.max(vec1)  
# Find the index of the maximum  
np.argmax(vec1)  
  
# Find the mean value  
np.mean(vec1)
```

In short if its used in class, there is probably an np method that does it for you.



Distribution Sampling

```
# Sampling from distributions  
# Random float in [0, 1)  
np.random.rand()  
# Random float from uniform  
np.random.uniform(low, high)  
# Random int from binomial  
np.random.binomial(n, p)  
# Random float from normal  
np.random.normal(mean, std_dev)
```

Matplotlib Basics





What is matplotlib?

“**Matplotlib** is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.”

- <https://matplotlib.org/>





Plots types

```
import matplotlib.pyplot as plt
```

```
# Scatter plot
```

```
plt.scatter()
```

```
# Line graph
```

```
plt.plot()
```

```
# Show all plots you've made up until now
```

```
plt.show()
```

In general we will give you a lot of help with these e.g. setting up labels etc. and will try to only be making you do work related to the class. We will also give very explicitly instructions for what to be plotting.

E.g. we will set up graph labels and all you will have to do is choose the correct x & y labels.