Computing for Data Analytics
CPSC 4800

Python, Data Structures

Langara.
THE COLLEGE OF HIGHER LEARNING

Learning Objective

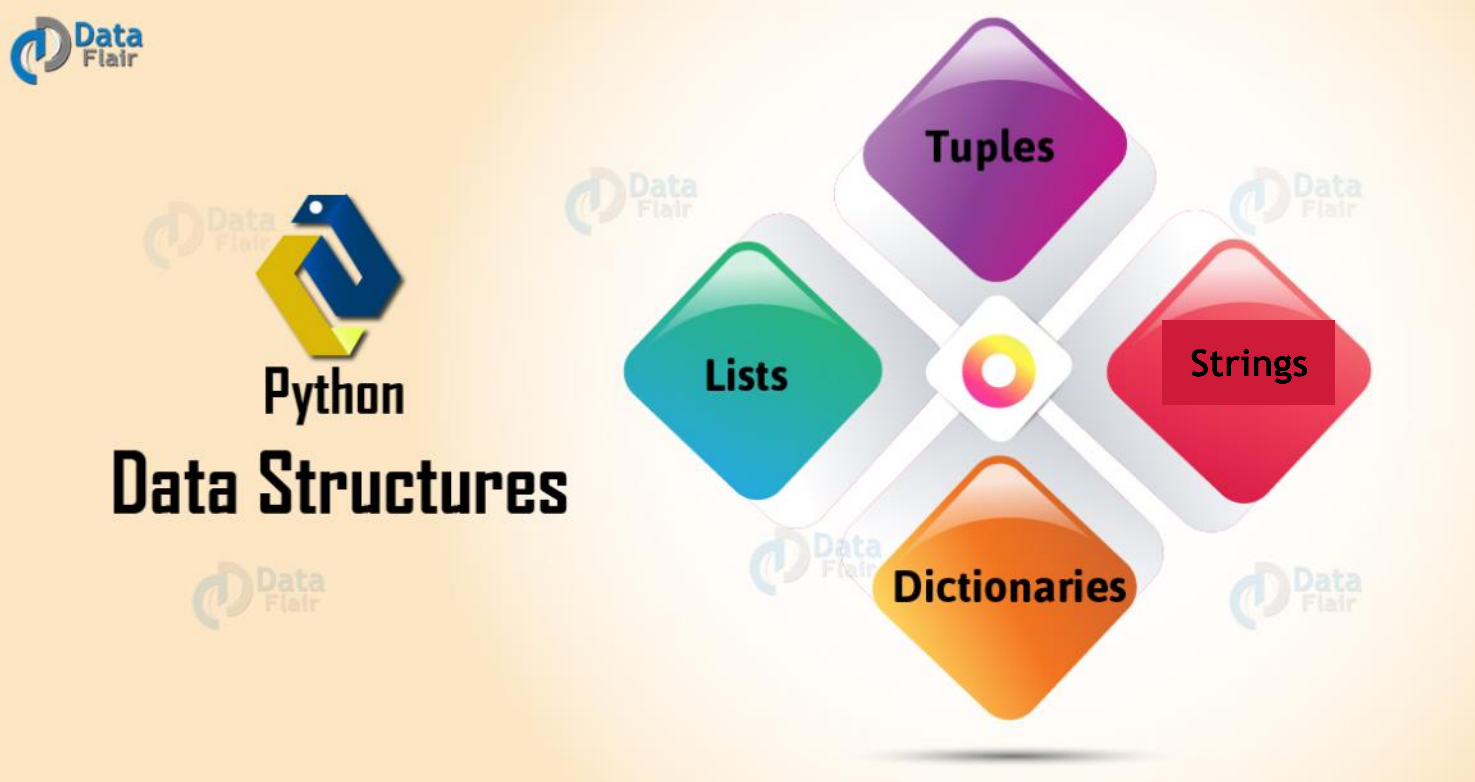
By the end of this lecture, students should:

- Learn Python important data structures
- List common operators for each data type and explain their outcomes

Python built in data structures

Python has set of built in data structures:

- Strings
- Lists
- Tuples
- Dictionaries



Strings

A string is a sequence of letters (called characters).

In Python, strings start and end with single or double quotes.

```
>>> "foo"
'foo'
>>> 'foo'
'foo'
```

Accessing single characters

You can access individual characters by using indices in square brackets.

```
>>> word = 'python'
>>> word [0] # character in position 0
'p'
>>> word [5] # character in position 5
'n'
```

Accessing substrings

>>> word [0:2] # characters from position 0 (included) to 2 (excluded)

'py'

>>> word [2:5] # characters from position 2 (included) to 5 (excluded)

'tho'

String functionality

Length

```
>>> len(word)
6
```

Concatenation

```
>>> "python" + " " +  "is a programming
   language"
'python is a programming language'
```

What are some string methods?

String methods

- In Python, a method is a function that is defined with respect to a particular object.

```
>>> word = 'python'

>>> word.find('t')

2
```

String methods

```
>>> word.find('p')
0
>>> word.upper()
'PYTHON'
>>> word.lower()
'python'
```

Mutable vs Immutable

Mutable: A mutable object can be changed after it is created

Immutable: An immutable object can’t change after it is created

Strings are immutable

Strings cannot be modified; instead, create a new one.

```
>>> message = "Welcome to the python
   course"
>>> message [0] = 'p'
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'str' object does not support
   item assignment
```

Strings are immutable

String methods do not modify the string; they return a new string.

```
>>> word = 'python'
>>> word.replace('p','j')
'jython'
>>> word
'python’

>>> new_word = word.replace('p','j')
>>> new_word
'jython'
```

Exercise

Learn more about the following methods for strings and provide couple of examples:

- Strings: Converts the first character of each word to upper case
- Strings: find the number of times the value occurs in a string
- Strings: Check if the string starts or ends with the specified value
- What is the difference between capitalize() and title() methods? Provide examples.

Lists

- An ordered collection of items
- List items shouldn’t be of same type
  - We can have numbers, strings, list, etc in the same list

```
>>> squares = [1, 4, 9, 16, 25]
>>> squares
[1, 4, 9, 16, 25]
```

Lists are mutable

```
>>> squares = [1, 4, 9, 16, 25]
>>> squares [0] = 2
>>> squares
[2, 4, 9, 16, 25]
```

Slicing

- Like strings, lists can be indexed and sliced:

```
>>> squares = [1, 4, 9, 16, 25]

>>> squares [0]
1
>>> squares [-1]
25
```