ITA6017
Python Programming

Dr. Arun Pandian J

Module 4: Python Strings & Regular Expressions

Strings: Understanding string in build methods and Operations [slicing], Regular Expressions: Powerful pattern matching and searching, Power of pattern searching using regex in python, Real time parsing of networking or system data using regex, Password, email, url validation using regular expression, Pattern finding programs using regular expression.

String Methods

- Strings, revisited
- Objects and their methods
- Indexing and slicing
- Some commonly used string methods

Remember: What is a string?

- A string is a sequence of zero or more characters
- A string is delimited (begins and ends) by single or double quotes

```
poem = 'Ode to a Nightingale'
lyric = "Roll on, Columbia, roll on"
exclamation = "That makes me !#? "
```

- The empty string has zero characters (" or "")

Quote characters in strings

- You can include a single quote in a double quoted string or a double quote in a single quoted string

```
will = "All the world's a stage"
ben = 'BF: "A penny saved is a penny earned"'
```

- To put a single quote in a single quoted string, precede it with the backslash ('\') or 'escape' character.

```
>>> will = 'All the world\'s a stage'
>>> print(will)
All the world's a stage
```

- The same goes for double quotes

```
>>> ben = "BF: \"A penny saved is a penny earned\""
>>> print(ben)
BF: "A penny saved is a penny earned"
```

Putting a format character in a string

- A format character is interpreted by the print() function to change the layout of text
- To put a format character in a string, precede it with the backslash ('\')
- A newline is represented by '\n'

```
>>> juliette = 'Good night, good night\nParting is such sweet sorrow'
>>> print(juliette)
Good night, good night
Parting is such sweet sorrow
```

- A tab is represented by '\t'

```
>>> tabs = 'col0\tcol1\tcol2'
>>> print(tabs)
col0	col1 col2
```

Index of string characters

- The first character of a string has index 0

```
>>> greeting = 'hello, world'
>>> greeting[0]
'h'
>>> 'hello, world'[0]
'h'
```

- You can also count back from the end of a string, beginning with -1

```
>>> greeting = 'hello, world'
>>> greeting[-1]
'd'
>>> 'hello, world'[-1]
'd'
```

Slicing a string

- You can use indexes to slice (extract a piece of) a string
- aStr[i:j] is the substring that begins with index i and ends with (but does not include) index j

```
>>> greeting = 'hello, world'
>>> greeting[1:3]
'el'
>>> greeting[-3:-1]
'rl'
```

- omit begin or end to mean 'as far as you can go'

```
>>> print(greeting[:4], greeting[7:])
hell world
```

- aStr[i:j:k] is the same, but takes only every k-th character

```
>>> greeting[3:10:2]
'l,wr'
```

Index/slice a string vs index/slice a list

How they're the same and how they're different

- SAME:
  - You can index a list or string by providing an integer index value, beginning with 0 from the left or -1 from the right [i].
  - You can slice a list or string by providing begin and end values ([i:j]) or begin, end and step values ([i:j:k])
  - You can omit begin or end ([:j] or [i:]) to mean 'as far as you can go'

List index vs string index (continued)

- DIFFERENT:
  - if you reference a single element of a list with the index operator ([i]), its type is the type of that element

```
>>> abc = ['a', 'b', 'c']
>>> abc[0]
'a'
>>> type(abc[0])
<class 'str'>
```

  - If you slice (extract a piece of) a list with begin and end ([i:j]) values, you get a sublist (type list)

```
>>> abc[0:2]
['a', 'b']
>>> type(abc[0:2])
<class 'list'>
```

String methods

- A method is a function that is bundled together with a particular type of object
- A string method is a function that works on a string
- This is the syntax of a method:

```
anObject.methodName(parameterList)
```

- For example,

```
>>> 'avocado'.index('a')
0
```

returns the index of the first 'a' in 'avocado'

- You can also use a variable of type string

```
>>> fruit = 'avocado'
>>> fruit.index('a')
0
```

Method parameters

- Like any function, a method has zero or more parameters
- Even if the parameter list is empty, the method still works on the 'calling' object:

```
>>> 's'.isupper()
False
```

- Here is a string method that takes two parameters:

```
>>> aStr = 'my cat is catatonic'
>>> aStr.replace('cat', 'dog')
'my dog is dogatonic'
```

Strings are immutable

- A string is immutable -- once created it can not be modified
- When a string method returns a string, it is a different object; the original string is not changed

```
>>> aStr = 'my cat is catatonic'
>>> newStr = aStr.replace('cat', 'dog')
>>> newStr
'my dog is dogatonic'
>>> aStr
'my cat is catatonic'
```

- However, you can associate the old string name with the new object

```
>>> aStr = 'my cat is catatonic'
>>> aStr = aStr.replace('cat', 'dog')
>>> aStr
'my dog is dogatonic'
```

String Methods

len()	min()	max()	isalnum()	isalpha()
isdigit()	islower()	isuppe()	isspace()	isidentifier()
endswith()	startswith()	find()	count()	capitalize()
title()	lower()	upper()	swapcase()	replace()
center()	ljust()	rjust()	center()	isstrip()
rstrip()	strip()			

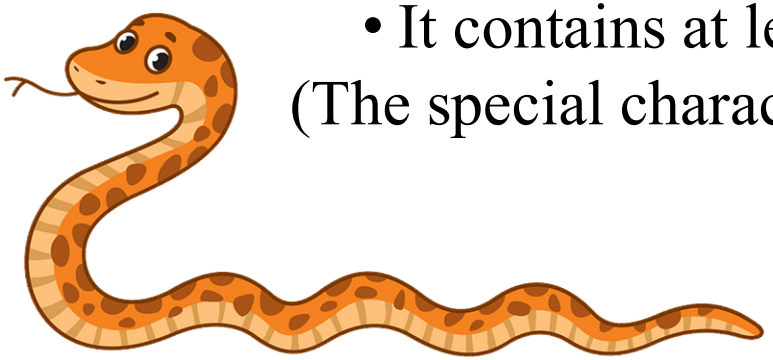
Task -1:

Validate the user entered password using following constraints:

Its length is at least 6

- It contains at least one digit.
- It contains at least one lowercase English character.
- It contains at least one uppercase English character.
- It contains at least one special character.

(The special characters are: !@#$%^&*()-+)



Task 2:

- Find the given string is duplicate shuffled string of original string

Original String = “abc”
Input = “bac”
Output = Duplicate String

Input = “cab”
Output = Duplicate String

Input = “ccab”
Output = Unique string

Input = “ab”
Output = Unique string

Task -3:

Read a entire paragraph as a string and separate each sentence of the paragraph.

Input:

Python Programming can be used to process text data for the requirements in various textual data analysis. A very important area of application of such text processing ability of python is for NLP. NLP is used in search engines, newspaper feed analysis and more recently for voice -based applications like Siri and Alexa. Python's NLTK is a group of libraries that can be used for creating such Text Processing systems.

Output:

Python Programming can be used to process text data for the requirements in various textual data analysis.

A very important area of application of such text processing ability of python is for NLP.

NLP is used in search engines, newspaper feed analysis and more recently for voice -based applications like Siri and Alexa.

Python's NLTK is a group of libraries that can be used for creating such Text Processing systems.