Working with Strings and Files

HORT 530

Lab 10

Instructor: Kranthi Varala

Today's pairs

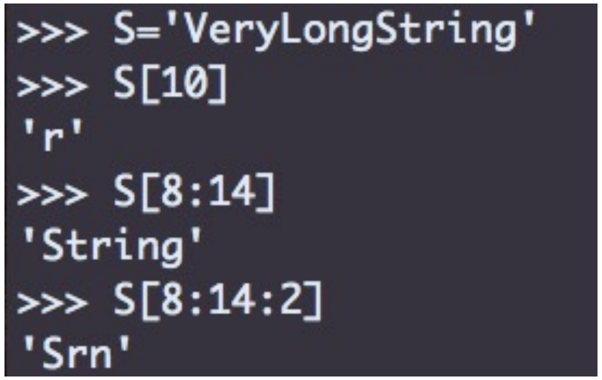
Chris, Sharlene	Emily, Meredith	Scott, Freddie	Xiaohui, Hui	Mithila, Maria	Rachel, Brenden

Strings: Special characters

- String special characters are defined by a preceding backslash, also called the escape character.
- '\n' = newline, '\r' = carriage return, '\t' = tab
- Unix: '\n'   DOS: '\r\n'
- The escape character and the character after it are interpreted as one character.
    - len('my\tworld\n') : 9
- Valid escape characters are interpreted by print method.

Strings: Operations

- Slice: S[i:j] returns the characters from i to j.
- Slice with step: S[i:j:k] returns all characters from i to j but moves by k characters instead of 1.

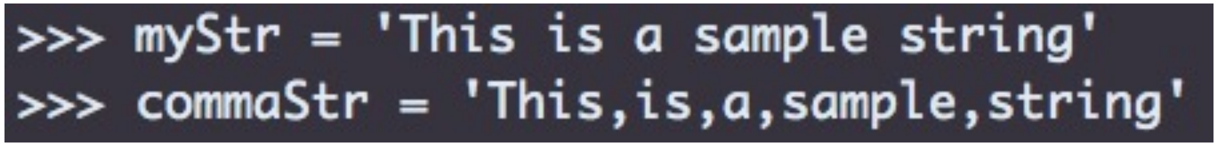


```
>>> S='VeryLongString'
>>> S[10]
'r'
>>> S[8:14]
'String'
>>> S[8:14:2]
'Srn'
```

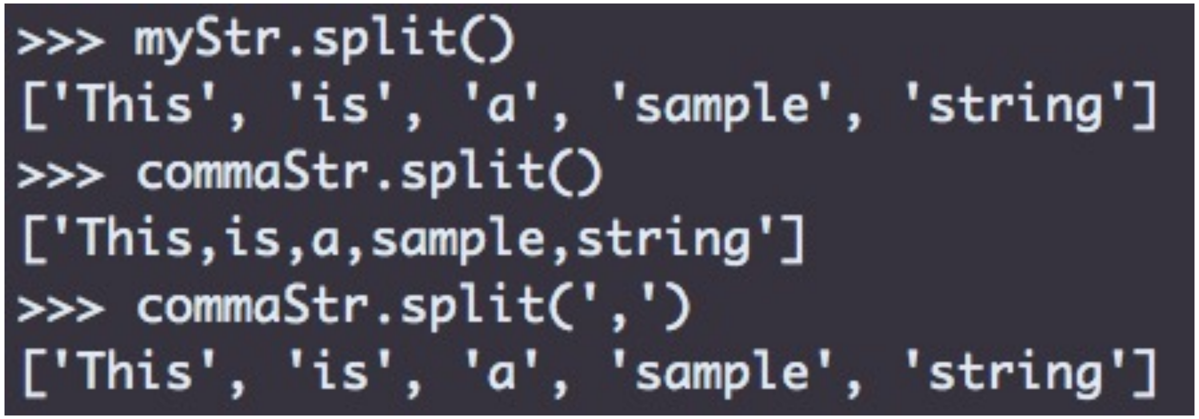
- S[:] returns a copy of the string S.
  - Alternate way to create a copy of a string, instead of using copy.copy()
  - Also works with lists

Splitting a string

- str.split('sep'): split the string based on the separator given and return the substrings as a list. Default separator is space.



```
>>> myStr = 'This is a sample string'
>>> commaStr = 'This,is,a,sample,string'
```



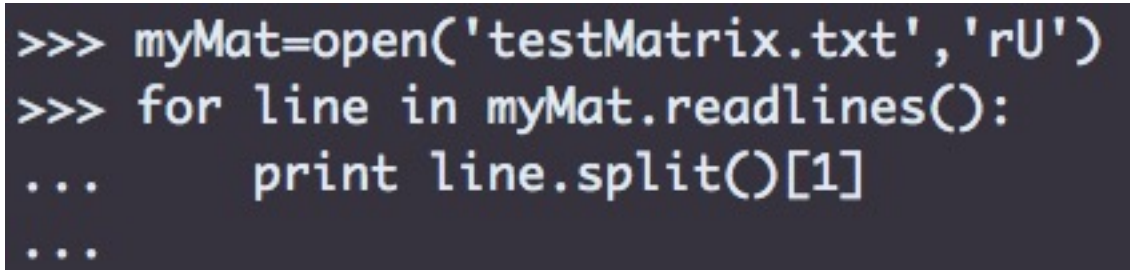
```
>>> myStr.split()
['This', 'is', 'a', 'sample', 'string']
>>> commaStr.split()
['This,is,a,sample,string']
>>> commaStr.split(',')
['This', 'is', 'a', 'sample', 'string']
```

Files

- File objects are a reference to the file on the storage device.
- At initiation, File objects need to be told 2 things:
  - Path to file
  - Processing Mode: read, write, append
- Additional options to mode:
  - ‘b’ : open file in binary mode
  - ‘+’ : open file for reading and writing

Reading from files

- Default options for file objects are: a. read b. text mode.
- Primary read methods for File object:
  - read() : Reads till given offset, or end of file (EOF)
  - readline(): reads the next line
  - readlines(): when used in a loop reads one line at a time until EOF
- All read methods return a String object.



```
>>> myMat=open('testMatrix.txt','rU')
>>> for line in myMat.readlines():
...     print line.split()[1]
...
```

Writing to files

- Primary write method is: write(str) Note that it only accepts a string object.
- String objects can be converted to other data types as required.
- Convert all your data to string before writing to file.
- Writing uses a buffer that minimizes number of write events.
- Force writing by using the file.flush() method

Common file operations

```
inFile = open('data.txt','r')
```

Option 1. `fileContents = inFile.read()`
Option 2. `oneLine = inFile.readline()`
Option 3. `fileAsList = inFile.readlines()`

```
inFile.close()
```

```
outFile = open('out.txt','w')
outFile.write(<string>)
outFile.writelines(<list>)
outFile.flush()
outFile.close()
```

Working with files

- Using the input file: "/scratch/scholar/kvarala/ICB/Week10/GSE49418_series_matrix.txt", print the average WT expression and average mutant expression values for each gene.

- Output should look as follows:

```
"Gene"              Avg.WT              Avg.MT
```

NOTES:

Handle the header line differently.

Convert line (string) to list of strings.

Convert strings to floats when needed.