

(1) String Representation

```
In [ ]: ...  
        Print out following four messages:  
  
        1) I am sick  
        2) I'm sick  
        3) I'm "sick"  
        4) I'm ...  
           ... "sick"  
        ...
```

```
In [1]: print('I am sick')
```

I am sick

```
In [2]: print("I'm sick")
```

I'm sick

```
In [3]: print('I\'m "sick"')
```

I'm "sick"

```
In [4]: print('I\'m ...\n... "sick"')
```

I'm ...
... "sick"

(2) Indexing (Slice)

```
In [5]: ...  
        The indexing operator can also be used to obtain slices of a list as well.  
  
        Let list lst refer to list  
  
        -> ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h']  
  
        Write Python expressions using list lst and the indexing operator that evaluate to:  
        ...  
  
        lst = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h']
```

```
In [ ]: # ['a', 'b', 'c', 'd']
```

```
In [6]: lst[0:4]
```

Out[6]: ['a', 'b', 'c', 'd']

In [7]: `lst[:4]`

Out[7]: ['a', 'b', 'c', 'd']

In [8]: `# ['d', 'e', 'f']`

In [9]: `lst[3:6]`

Out[9]: ['d', 'e', 'f']

In [10]: `# ['d']`

In [11]: `lst[3:4]`

Out[11]: ['d']

In [12]: `# ['f', 'g']`

In [13]: `lst[-3:-1]`

Out[13]: ['f', 'g']

In [14]: `# ['d', 'e', 'f', 'g', 'h']`

In [15]: `lst[3:]`

Out[15]: ['d', 'e', 'f', 'g', 'h']

In [16]: `# ['f', 'g', 'h']`

In [17]: `lst[-3:]`

Out[17]: ['f', 'g', 'h']

In [18]: `lst[:] # Prints the entire list`

Out[18]: ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h']

(3) String Methods

```
In [19]: # String Link Describes the URL of a Webpage  
  
link = 'http://www.main.com/smith/index.html'  
print(link)
```

http://www.main.com/smith/index.html

```
In [20]: # Return 'http' From the Link
```

```
In [21]: link[:4]
```

Out[21]: 'http'

```
In [22]: # Return 'HTTP' From the Link
```

```
In [23]: link[:4].upper()
```

Out[23]: 'HTTP'

```
In [24]: # Return the Index of the First Occurrence of 'smith' In the Link
```

```
In [25]: link.find('smith')
```

Out[25]: 20

```
In [26]: # Return 'smith' From the Link
```

```
In [27]: link[20:25]
```

Out[27]: 'smith'

```
In [28]: link[link.find('smith'):link.find('smith') + len('smith')]
```

Out[28]: 'smith'

```
In [29]: def sl(s):  
    return link[link.find(s):link.find(s) + len(s)]
```

```
In [30]: sl('index')
```

Out[30]: 'index'

```
In [31]: # Return 'Smith' From the Link
```

```
In [32]: link[link.find('smith'):link.find('smith') + len('smith')].capitalize()
```

```
Out[32]: 'Smith'
```

```
In [33]: link[20:25].capitalize()
```

```
Out[33]: 'Smith'
```

```
In [34]: # Count the Total Number of '/' In the Link
```

```
In [35]: link.count('/')
```

```
Out[35]: 4
```

```
In [36]: # Return a List of Substrings of the Link Delimited by '/'
```

```
In [37]: link.split('/')
```

```
Out[37]: ['http:', '', 'www.main.com', 'smith', 'index.html']
```

```
In [38]: # (2) String Events Describe the Schedule of 4 Events Spread Across 3 Days
```

```
events = '9/13 2:30 PM\n9/14 11:15 AM\n9/14 1:00 PM\n9/15 9:00 AM'\nprint(events)
```

```
9/13 2:30 PM
9/14 11:15 AM
9/14 1:00 PM
9/15 9:00 AM
```

```
In [39]: # The Number of Events on 9/14
```

```
In [40]: events.count('9/14')
```

```
Out[40]: 2
```

```
In [41]: # The Index of the Substring Describing the First Event on 9/14
```

```
In [42]: events.find('9/14')
```

```
Out[42]: 13
```

```
In [43]: # The Index Just Past the Substring Describing the Last Event on 9/14
```

```
In [44]: events.find('9/15')
```

```
Out[44]: 40
```

```
In [45]: # The List of Substrings Describing the Events on 9/14
```

```
In [46]: events[13:39].split('\n')
```

```
Out[46]: ['9/14 11:15 AM', '9/14 1:00 PM']
```

```
In [47]: events[events.find('9/14'):events.find('9/15') - 1].split('\n')
```

```
Out[47]: ['9/14 11:15 AM', '9/14 1:00 PM']
```

(4) Print () Revisited: General Formatting

```
In [48]: '''
There is a list of pets (i.e. boa, cat, and dog)
Print out followings
'''

pets = ['boa', 'cat', 'dog']
```

```
In [ ]: # (1)

'''
boa
cat
dog
'''
```

```
In [50]: for i in pets:
          print(i)
```

```
boa
cat
dog
```

```
In [51]: for i in pets:
          print(i, end = '\n')
```

```
boa
cat
dog
```

```
In [ ]: # (2)

'''
boa? cat? dog?
'''
```

```
In [52]: for i in pets:
          print(i, end = '? ')
```

boa? cat? dog?

```
In [53]: '''
          Suppose we have following string variables: weekday, month, day, year, hour, minute, and second.
          Please print out following:
          '''

          weekday = 'Monday'
          month = 'Feburary'
          day = 4
          year = 2019
          hour = 17
          minute = 13
          second = 32
```

```
In [ ]: # (1)

'''
17:13:32
'''
```

```
In [54]: print(str(hour) + ':' + str(minute) + ':' + str(second))
```

17:13:32

```
In [55]: print('{}: {}: {}'.format(hour, minute, second))
```

17:13:32

```
In [ ]: # (2)

'''
Monday, Feburary 4, 2019 at 17:13:32
'''
```

```
In [56]: print('{} , {} {}, {} at {}: {}: {}'.format(weekday, month, day, year, hour, minute, second))
```

Monday, Feburary 4, 2019 at 17:13:32

(5) Reading a Text File "example.txt"

```
In [57]: # Read the File Contents Into a List of Words

infile = open('example.txt')
contents = infile.read()
infile.close()

contents.split() # Creates a List of "example.txt"
```

```
Out[57]: ['1',
          'The',
          '3',
          'lines',
          'in',
          'this',
          'file',
          'end',
          'with',
          'the',
          'new',
          'line',
          'character.',
          '2',
          '3',
          'There',
          'is',
          'a',
          'blank',
          'line',
          'above',
          'this',
          'line.',
          'stuffstuffstuffwhateverand',
          'more',
          'whatever']
```

```
In [58]: # Read the File Contents Into a List of Lines

infile = open('example.txt')
contents = infile.readlines()
infile.close()

contents
```

```
Out[58]: ['1 The 3 lines in this file end with the new line character.\n',
          '2 \n',
          '3 There is a blank line above this line.\n',
          '\n',
          'stuffstuffstuffwhateverand more whatever']
```

```
In [59]: # Define a Function num_char(file_name), Which Returns the Number of Characters of the

def num_char(file_name):
    infile = open(file_name)
    contents = infile.read()
    infile.close()
    return len(contents)
```

```
In [60]: num_char('example.txt')
```

Out[60]: 145

```
In [61]: # Define a Function num_words(file_name), Which Returns the Number of Words of the Input

def num_words(file_name):
    infile = open(file_name)
    contents = infile.read().split()
    infile.close()
    return len(contents)
```

```
In [62]: num_words('example.txt')
```

Out[62]: 26

```
In [63]: # Define a Function num_lines(file_name), Which Returns the Number of Lines of the Input

def num_lines(file_name):
    infile = open(file_name)
    contents = infile.readlines()
    infile.close()
    return len(contents)
```

```
In [64]: num_lines('example.txt')
```

Out[64]: 5

(6) Writing to a Text File "new_file.txt"

Hello, World!

I love Python programming.

Really?

```
In [ ]: # Write to a New Text File

outfile = open('new_file.txt', 'w')
outfile.write('Hello, World!\n\n')
outfile.write('I love Python programming.\n\n')
outfile.write('Really?')
outfile.close()

'''
outfile = open('new_file.txt', 'w')
outfile.write('Hello, World!\n\nI love Python programming.\n\nReally?')
outfile.close()
'''
```