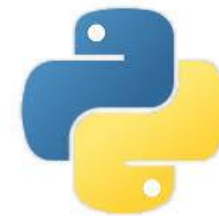


CMPT 120: Introduction to Computing Science and Programming 1

Strings, Lists, and Control Structures: while



python™

Copyright © 2018, Liaqat Ali. Based on [CMPT 120 Study Guide](#) and [Think Python - How to Think Like a Computer Scientist](#), mainly. Some content may have been adapted from earlier course offerings by Diana Cukierman, Anne Lavergn, and Angelica Lim. Copyrights © to respective instructors. Icons copyright © to their respective owners.

Reminders

Liaqat Ali, Summer 2018.

One-Stop Access To Course Information

- **Course website**: One-stop access to all course information.

<http://www2.cs.sfu.ca/CourseCentral/120/liaqata/WebSite/index.html>

- Course Outline
- Exam Schedule
- Python Info
- **CourSys/Canvas** link
- Learning Outcomes
- Office Hours
- Textbook links
- and more...
- Grading Scheme
- Lab/Tutorial Info
- Assignments

- **Canvas**: Discussions forum - <https://canvas.sfu.ca/courses/39187>

- **CourSys**: Assignments submission, grades - www.coursys.sfu.ca

How to Learn in This Course?



- A** **Attend** Lectures & Labs
- R** **Read** / review Textbook/Slides/Notes
- R** **Reflect** and ask Questions
- O** **Organize** – your learning activities on weekly basis,
and finally...
- W** **Write** Code, **Write Code**, and **Write Code**.

Deliverables

1. Deliverables are due by the given date and time.
2. For the course, we are using IDLE to write and run our Python code.
3. You can use the CSIL lab computers outside your lab hours.
4. Plan ahead your assignments and other deliverables. Computer crash, network problems etc. are not acceptable excuses for delays in deliverables.
5. You may use online Python interpreters for running and testing your codes, such as:

<https://repl.it/languages/Python3>

Labs

1. Each lab has an assigned TA.
2. Attend your assigned lab and show your work to your TA for the participation marks.
3. Class enrolments and lab swaps are closed now.

Course Topics

1. General introduction
2. Algorithms, flow charts and pseudocode
3. Procedural programming in Python
4. **Data types and Control Structures**
5. Fundamental algorithms
6. Binary encodings
7. Basics of computability and complexity
8. Basics of Recursion
9. Subject to time availability:
 - Basics of Data File management

Today's Topics

1. Strings

- String Methods

2. Lists

3. Practice

1

Strings and List

String Method split()

```
>>> sentence = "My name is Peter Robert Jackson!"
```

- **split()** function is used to parse or break a long string into words or parts.

- **Example:**

```
>>> 'My name is Peter Robert Jackson!.split()
```

```
['My', 'name', 'is', 'Peter', 'Robert', 'Jackson!']
```

```
>>> words = 'My name is Peter Robert Jackson!.split()
```

words	My	name	is	Peter	Robert	Jackson!
	0	1	2	3	4	5

```
print(words) # ['My', 'name', 'is', 'Peter', 'Robert', 'Jackson!']
```

```
len(words) # 6
```

```
print(words[3]) # 'Peter'
```

String Method `split()` - 2

```
words = ['My', 'name', 'is', 'Peter', 'Robert', 'Jackson!']
```

- The variable `words` in this case refer to group of values (i.e., not a single value.)
- So, the `split()` function creates a **list**.
- **List**: is an object that contains multiple data items.
- Values in a list can be accessed using an index.
 - **Examples:**

```
>>> words[1]
'name'
>>> words[3]
'Peter'
```

String Method `split()`: Separator

- By default, the `split` method uses **spaces** as separators (**that is, it returns a list of the words in the string that are separated by spaces**).
- You can specify a different separator by passing it as an argument.

- **Examples:**

```
>>> date_string = '11/26/2018'
```

```
>>> date_list = date_string.split('/')
```

```
>>> date_list
```

```
['11', '26', '2018']
```

2

Practice

Liaqat Ali, Summer 2018.

Class Participation: Add A While Loop

- Modify the Simple Calculator program you wrote in the class today to add a **while loop**.
- Exit the loop when the user enters **0** to end the program.
- Post your code on the **Canvas by tonight Friday, 11:59pm.**