

Session 1

Control Flow -- Part 1

if Statements

for Loops

Dictionaries

8.20.19

Link to Jupyter Notebooks:

<https://mybinder.org/v2/gh/data-voyage-solutions/oag-session-mats/master>



01

Schedule/ Topics

Discuss what to expect from planned training sessions.

02

Let's look ahead

Get an idea of what's to come in the next few weeks; gather info for project-based learning.

03

Progress Updates

Recap where we left off and any progress since the pilot workshop.

04

Jump in

- Control Flow Part 1: if statements and for loops
- Dictionaries

Meeting Date	Module	Sub-topic
8/20/19	1: Python Fundamentals	Control Flow Part 1 and Dictionaries
8/27/19		Control Flow Part 2
9/3/19		Versioning Control (Git)
9/10/19	2: Data Wrangling/Preparation	Loading data/Intro to Pandas
9/17/19		Common data cleaning tasks
9/24/19		Common errors encountered & solutions
10/1/19	3: EDA & Intro to Visualizations	Basic summary/descriptive statistics
10/8/19		How to choose the right/best chart
10/15/19		How to create different visuals in Python
10/22/19	4: Visualizations (e.g., Bokeh)	Design principles/Formatting
10/29/19		Interactive visuals
11/5/2019		Creating dashboards

Schedule/ Topics



Session 1

Control Flow Part 1 and
Dictionaries



Session 2

Dictionaries and
Control Flow Part 2



Session 3

Versioning Control (Git)



Session 4

Loading Data/Intro to Pandas

Does anyone on team
currently use Git?

Is Git used in team
workflows?

Or as individual
documentation?

Let's look ahead

Project-based learning

Typical Data Sets?
Sources--csv, SQL, API

Data loading
workflows?

Recap

- Syntax and Operators
- Simple Data Types
- Variables
- Data Structures: Lists, Tuples

**Progress
Updates**

Review

Data Structures (Containers)

lists	dictionaries
<code>[]</code>	<code>{ k : v }</code>

Control Flow

if statement	for loop	functions
--------------	----------	-----------

Control Flow

- for loops
- if Statements

Control Flow

A control structure is a block of programming that analyzes variables and chooses a direction in which to go based on given parameters.

- The term **control flow** details the direction the program takes (how program logic “flows”). It determines how a computer will respond when given certain conditions and parameters.

Control Structures

for loop

- Repeat operations
- Loop variable takes each value from list in turn
- Indentation controls end of loop
- Watch out for infinite loops!
 - Interrupt or restart kernel when this happens

```
1 | users = ["Jeff", "Jay", "Theresa"]
2 |
3 | for user in users:
4 |     print("Hello %s" % user)
```

Python

Control Flow: for loops

```
In [1]: total = 0
        for number in [1,2,3,4,5]:
            total += number
        print("total is:",total)
```

total is: 15

```
In [2]: import time

        counter = 0
        my_list = [1,2,3,4,5]
        for number in my_list:
            counter += 1
            my_list.append(counter)
            print("new_list:", my_list)
            time.sleep(1)
```

```
new_list: [1, 2, 3, 4, 5, 1]
new_list: [1, 2, 3, 4, 5, 1, 2]
new_list: [1, 2, 3, 4, 5, 1, 2, 3]
new_list: [1, 2, 3, 4, 5, 1, 2, 3, 4]
new_list: [1, 2, 3, 4, 5, 1, 2, 3, 4, 5]
new_list: [1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 6]
new_list: [1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 6, 7]
new_list: [1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 6, 7, 8]
new_list: [1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 6, 7, 8, 9]
new_list: [1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

2) Looping through an entire list using a for loop

Control Flow

- ~~for loops~~
- if Statements

Control Structures

if / elif / else

- Check conditions with boolean operators (i.e. <, >, ==)
- Execute a single code block depending on result
- If True: code runs
- Can be:
 - if alone
 - if/else
 - if/elif(s)/else
- Indentation controls end of if block
- Data controls which code runs

```
1 | if age_person > 18:  
2 |     return "They can drive"  
3 | else:  
4 |     return "They cannot drive"
```

Python



Control Flow: `if` Statements

```
In [ ]: A = 10  
        B = 100
```

```
In [ ]: if A == 10:  
        print("var is 10")
```

```
In [ ]: if B == 10:  
        print("var is 10")
```

```
In [ ]: if A == 10:  
        print("var is 10")  
        else:  
        print("var is not 10")
```

```
In [ ]: if B == 10:  
        print("var is 10")  
        else:
```

Control Structures - if / elif / else

Python

```
1 | A = 10
2 | B = 100
3 | if A>B:
4 |     print("A is larger than B")
5 | elif A==B:
6 |     print("A is equal to B")
7 | else:
8 |     print("A is smaller than B")
```

Deeper Dive: `if` statements

In this section, you will learn to:

- **write conditional tests**, which allow you to check any condition of interest.
- **write simple if statements**
- **create a more complex series of if statements** to identify when the exact conditions you want are present.
- **apply this concept to lists**, so you'll be able to write a for loop that handles most items in a list one way but handles certain items with specific values in a different way



Dictionary

Next time



Extra Review Questions (if needed)

Get through as many of the questions below.

Start!