



INTRO TO PYTHON

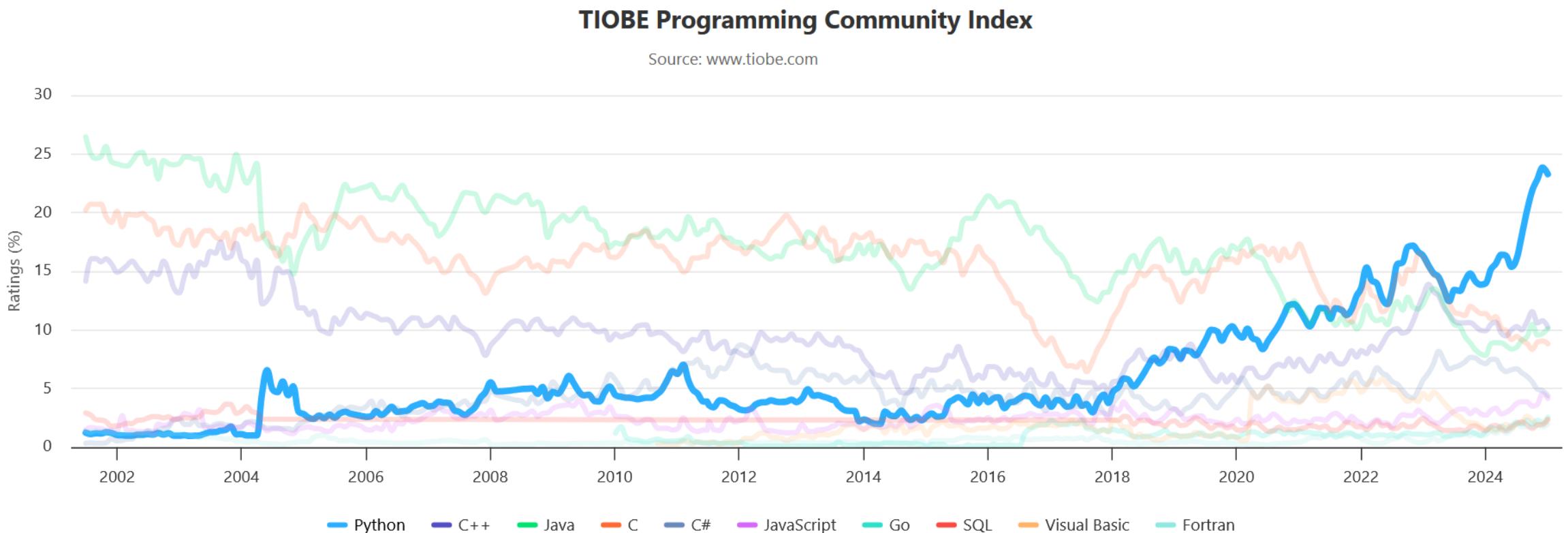
Basic Programming and Data Analysis

January 8, 2026 Prepared by Jay / MDAL (Built on Niti's past slides)
Website: <https://rmdal.github.io/python-intro-2026-winter/>



Rotman School of Management
UNIVERSITY OF TORONTO

Popularity of Python – Number 1



Ref. <https://www.tiobe.com/tiobe-index/>

What Can Python Do – Just About Anything

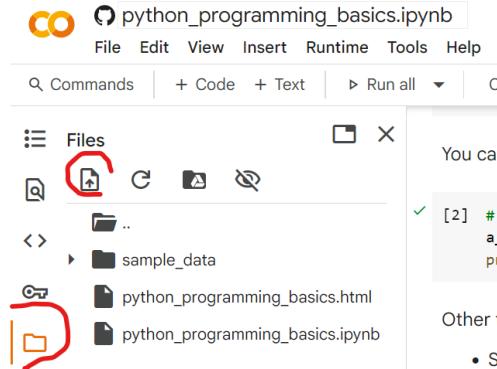
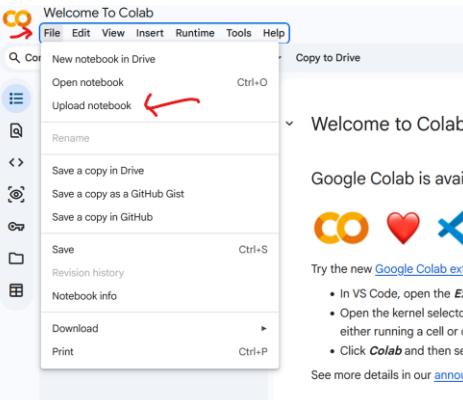
- Statistical analysis
- Machine learning / AI
- Data visualization
- Scientific computing
- Others
 - Scripting & automation
 - Web development
 - Systems testing & prototyping
 - Desktop applications

Install and Code in Python

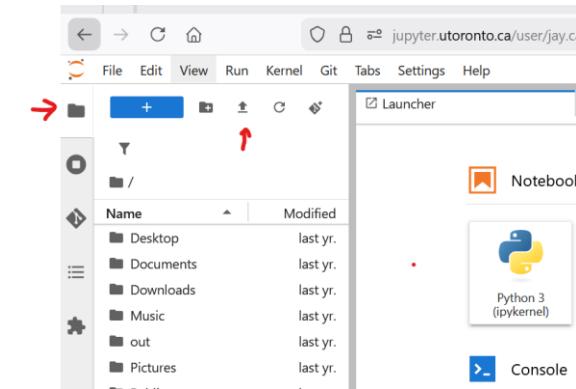
- Best Choice for Beginners: Notebook in the Cloud
 - [Google Colab \(our choice today\)](#)
 - [Uoft JupyterHub](#) (a good choice too)
 - use either *classic Jupyter Notebook* or *modern Jupyterlab* (both **2G memory limit**)
- Other Options
 - Local Notebook with [Anaconda Python Distribution](#)
 - Comes with a GUI launchpad, Jupyter Notebook, and many DS packages preinstalled
 - A large disk footprint (4.4G)
 - [Official Python Distribution](#) + a Code Editor (e.g. [VS Code](#))
 - Install needed packages/libraries on your own
 - Work well with both pure Python code and Python notebooks

More on Notebook in the Cloud

- Python Notebook (.ipynb) vs Python Script (.py)
- [Google Colab](#) or [Jupyterlab](#) at [UofT JupyterHub](#)
 - Upload a Notebook (e.g. assignment start-up code) and data



Colab



Jupyterlab

- Download a Notebook (as .ipynb, .html or .pdf for assignment submission)
 - *File -> Download (or Save and Export Notebook As)*
 - Note that Colab only support .ipynb download

Plan

- Programming basics (notebook on workshop site)
 - Data structures (string, list, dictionary, etc.)
 - Programming structures (conditional, loop, etc.)
 - Functions (functions and methods)
- Data analysis basics (notebook on workshop site)
 - Simple data processing (operations on 2D tables/dataframes)
 - Simple analysis on a stock price time series
 - Predict next-day stock price with a linear regression model

Basic Data Structure - 1

- What's data structure
 - a way of storing and organizing data/values of certain types
- Value, type, variable, and the assignment operator (=)



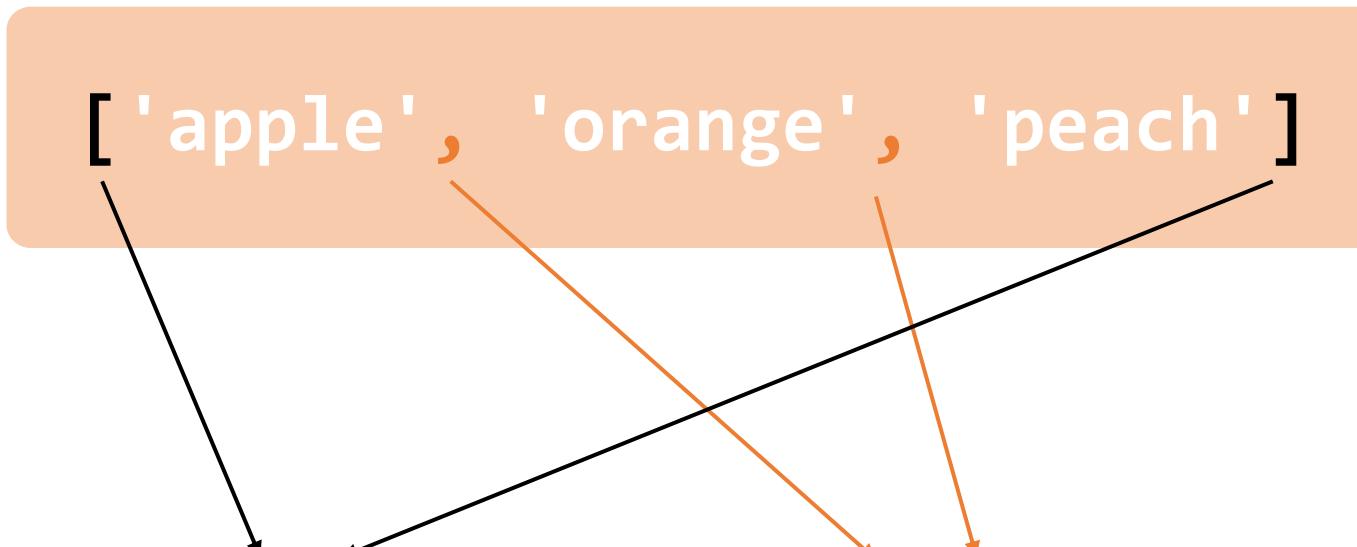
An assignment statement

Basic Data Structure - 2

- Basic numeric types: `int`, `float`, `complex`
- String type (`str`)
 - String index
 - Methods associated with string object

More DS Native to Python - List

- Mutable
- Ordered (Insertion order)
- Indexed



['apple', 'orange', 'peach']

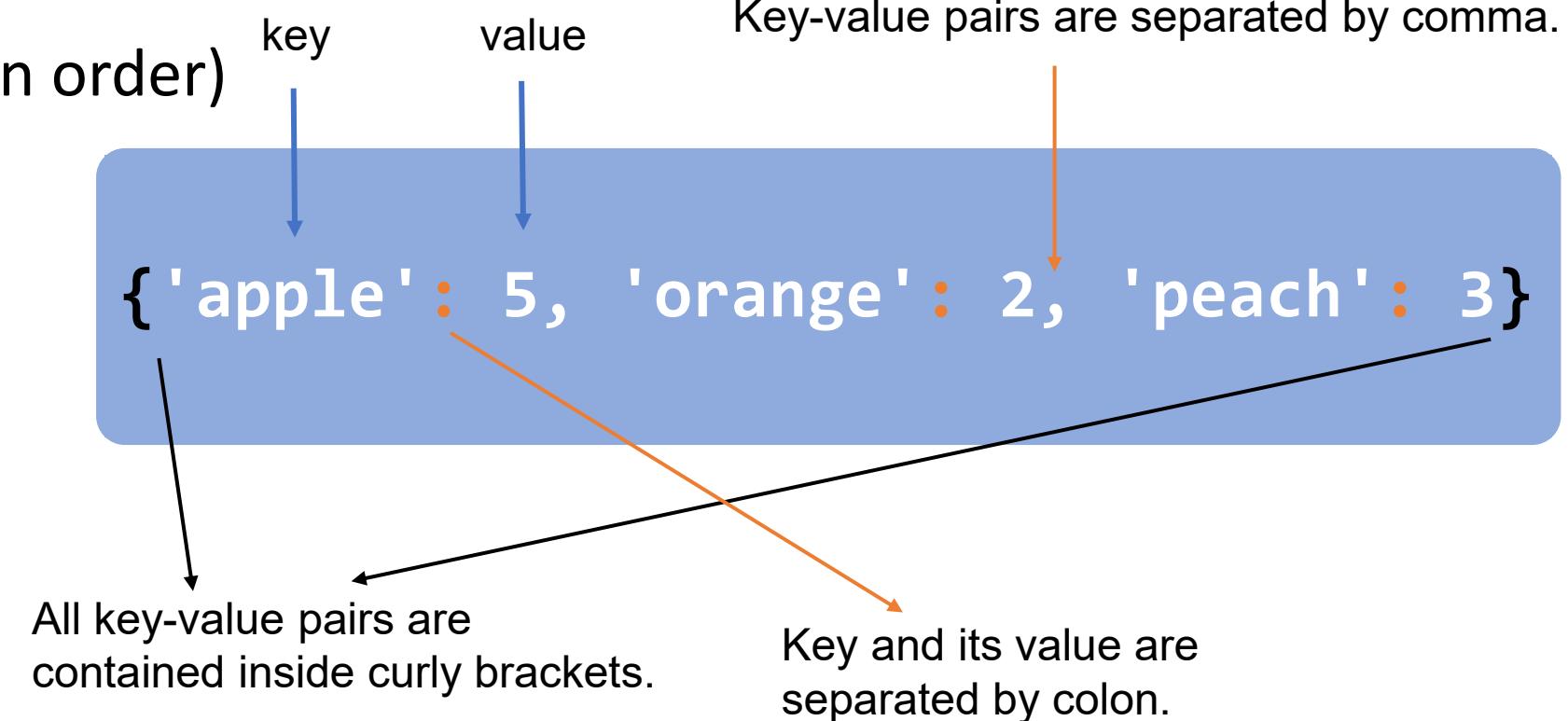
The diagram shows a list of three fruit names: 'apple', 'orange', and 'peach', enclosed in square brackets. Four arrows point from the text below to specific parts of the list: one arrow points to the left bracket, another to the first element 'apple', a third to the middle element 'orange', and a fourth to the right bracket.

All elements contained
inside square brackets.

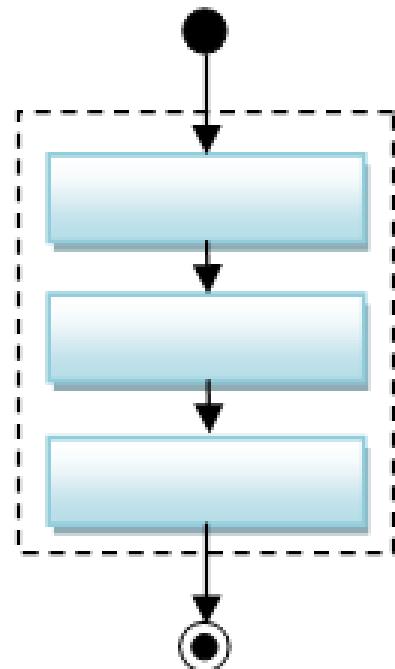
Elements are
separated by comma.

More DS Native to Python - Dictionary

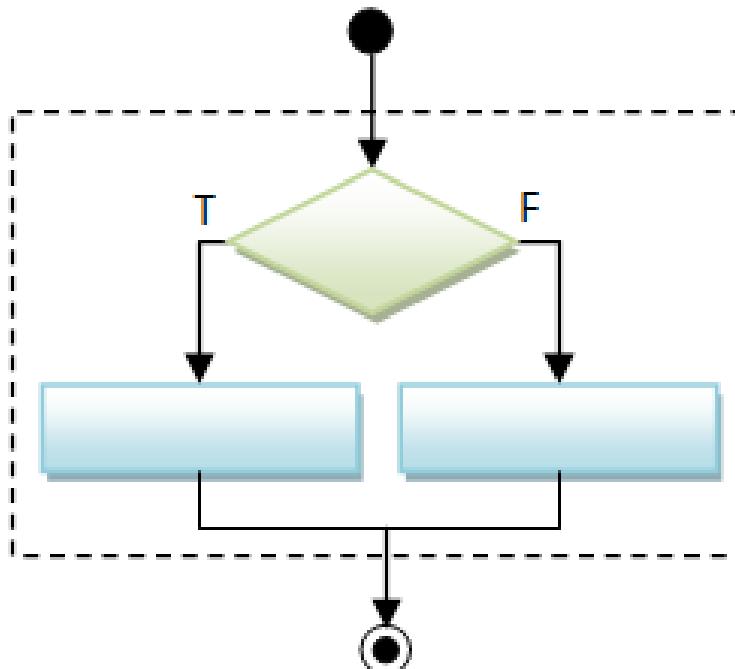
- Mutable
- Ordered (Insertion order)
- Key-value pair



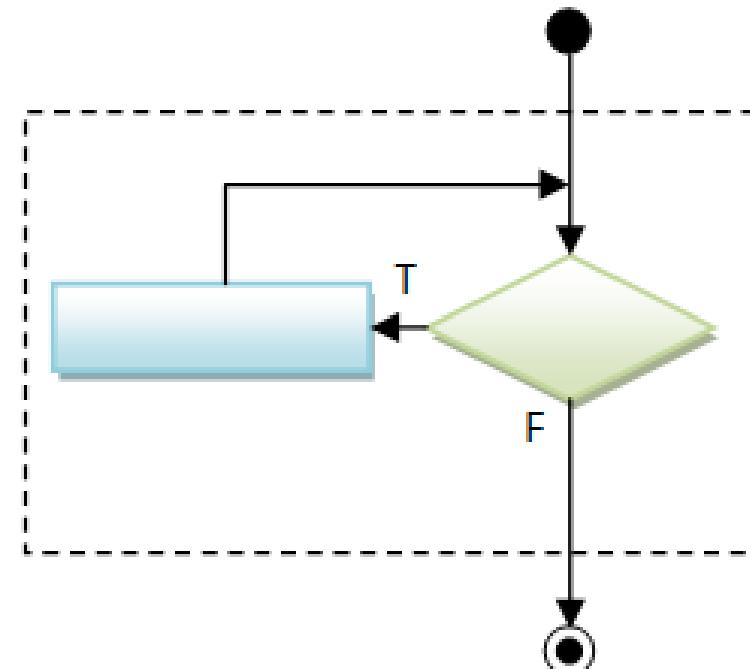
Programming Structures



Sequential



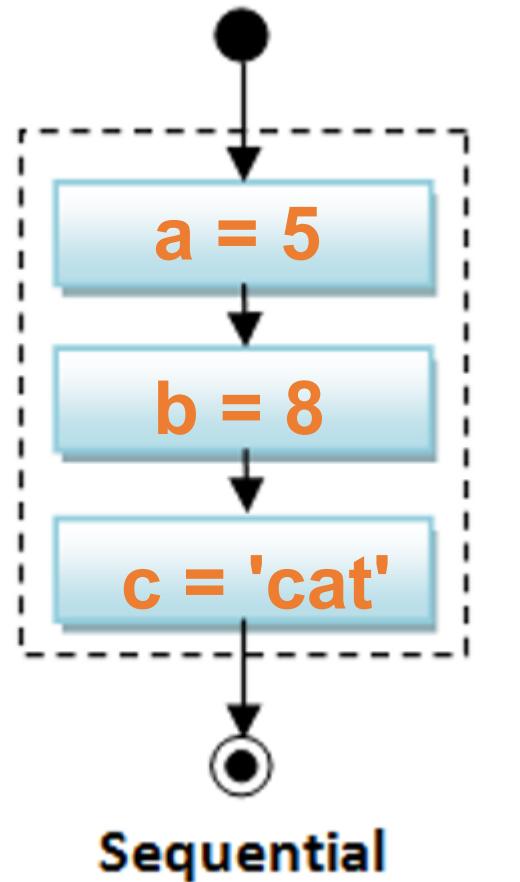
Conditional (Decision)



Loop (Iteration)

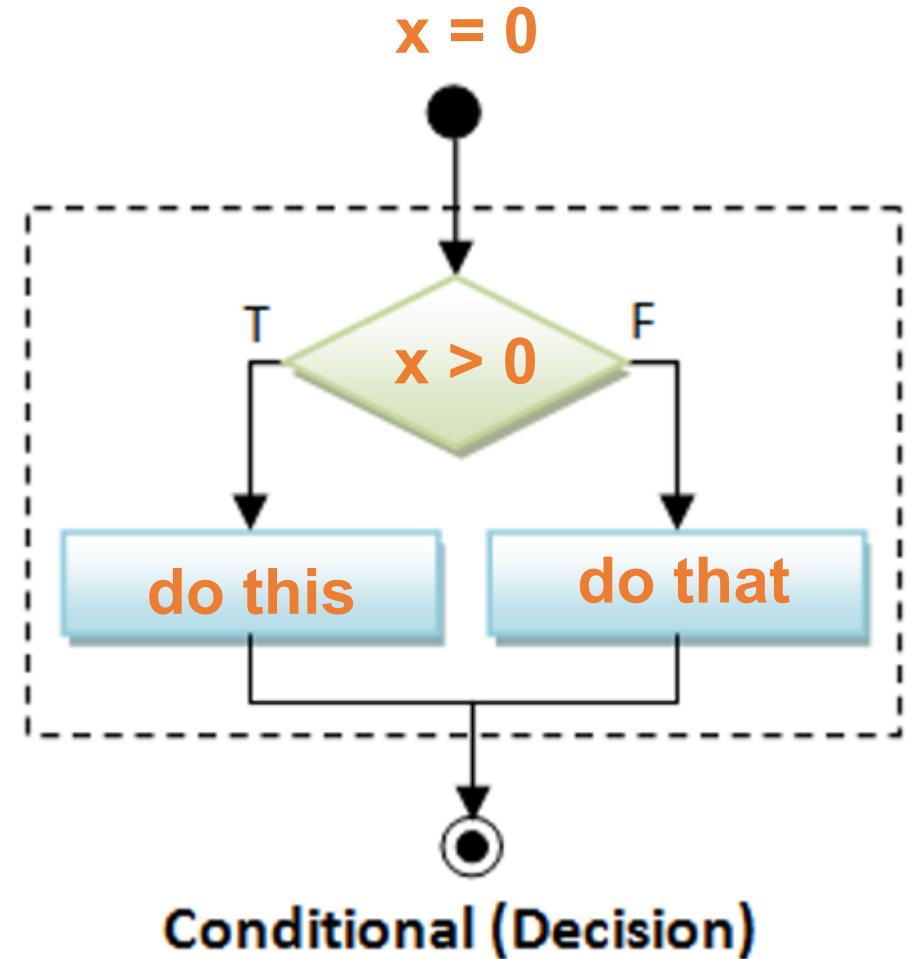
Sequential

- Code executes in sequence



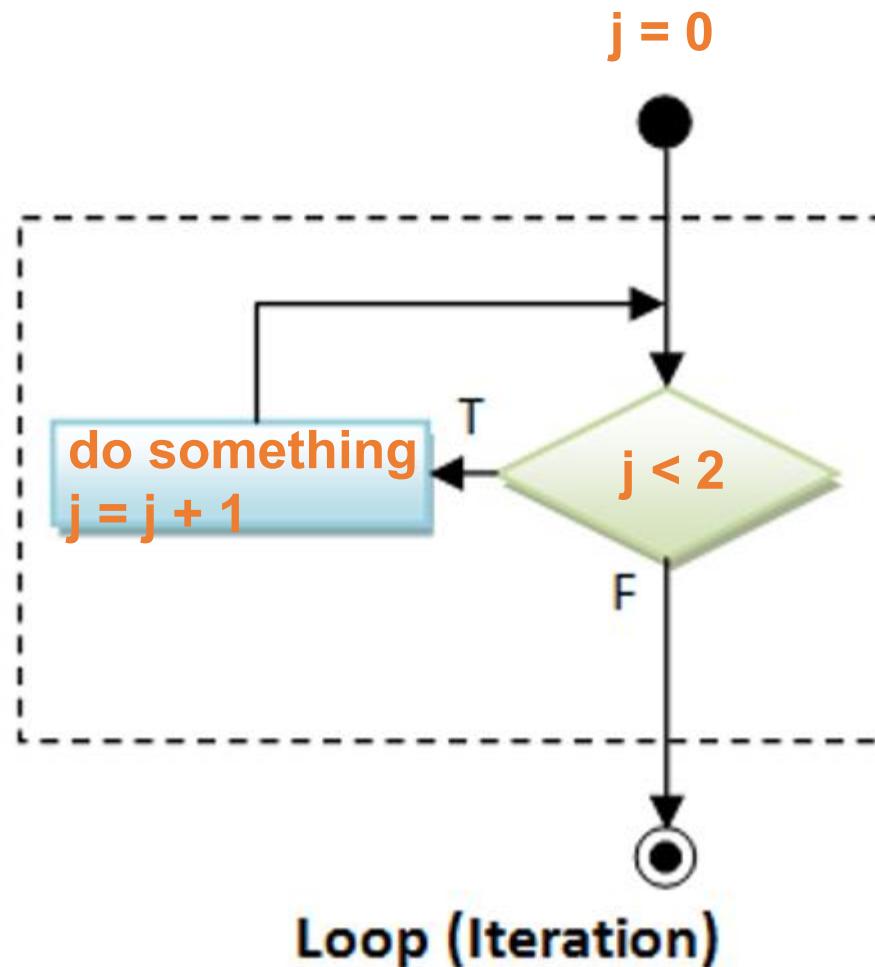
Conditional

- Whether a certain block of code is executed or not depends on whether a condition is satisfied.



Iterative

- A block of code repeatedly executed for either identical tasks or similar tasks



Function – Custom Functions

- What's a function
 - a logical block of code
 - input -> output
- Why write functions
 - Reusability
 - Abstraction
 - Maintainability

def celsius_to_fahr(celsius):
 fahr = 9/5 *celsius + 32
 return fahr

Annotations pointing to specific parts of the code:

- Declare a function with def keyword**: Points to the `def` keyword.
- Give it a meaningful name**: Points to the function name `celsius_to_fahr`.
- Parameters (input) on which the function is designed to work upon**: Points to the parameter `celsius`.
- Return statement**: Points to the `return fahr` statement.
- Value to return (output)**: Points to the value `fahr` being returned.
- Block of code that defines the operations to run**: A brace on the right side groups the entire function definition.

Other Functions

- Built-in functions
- A method is
 - A functions associated with an object (an instance of a class)
 - Accessed using the dot operator (.)

```
# create a list
num_list = [4, 8, 10, 15]
```

```
# print() and sum() are
# built-in functions
print(sum(num_list))
```

```
# remove is a method associated
# with a list object
num_list.remove(10)
print(num_list)
```

Functions/Methods From Other Packages

- Third-party packages/libraries offer functions for various of tasks
- Useful data science packages
 - [numpy](#): operating on vectors and matrices/arrays.
 - [pandas](#): processing 2D tables (dataframes).
 - [matplotlib](#): plotting.
 - [scikit-learn](#): machine learning.

```
# import the numpy module
import numpy as np

# create a 2x3 array
# array() is a function provided by numpy
ar = np.array([[1, 2, 3],
               [4, 5, 6]])

# print the array
print(ar)

# find the largest element in the array
# max() is a method associated with the array object
print(ar.max())

# find the array's shape
# shape is an attribute of the array object;
# it's not a method or function
print(ar.shape)
```

Data Analysis Basics

- Let's walk through the notebook together
 - Simple data processing (operations on 2D tables/dataframes)
 - Simple analysis on a stock price time series
 - Predict next-day stock price with a linear regression model