

***Rotman***

# INTRO TO PYTHON

February 3, 2020 Prepared by Niti / TDMDAL



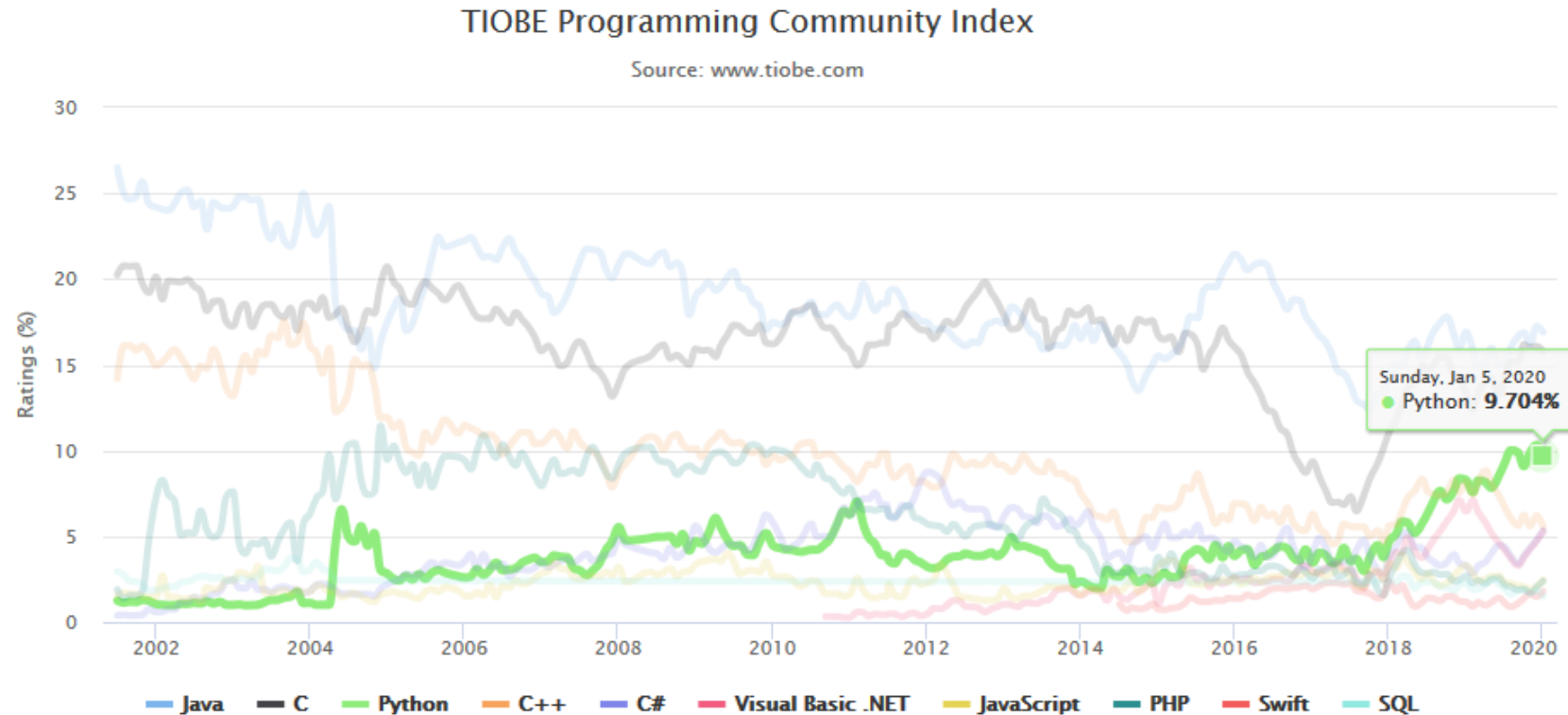
Rotman School of Management  
UNIVERSITY OF TORONTO

# Agenda

1. Python's Popularity
2. Data Structures
3. Programming Structures
4. Functions
5. Third Party Modules for Data Science

# Python's Popularity

In 2019, Python became the biggest gainer in the Tiobe index of language popularity, again!



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

# Python's Popularity

1. Statistical analysis
2. Scientific computing
3. Machine learning
4. Data visualization
5. Artificial intelligence
6. Others:
  - a) Scripting & automation
  - b) Web development
  - c) Systems testing & prototyping
  - d) Desktop & mobile applications
  - e) Education!

# Data Structures

# Data Structures

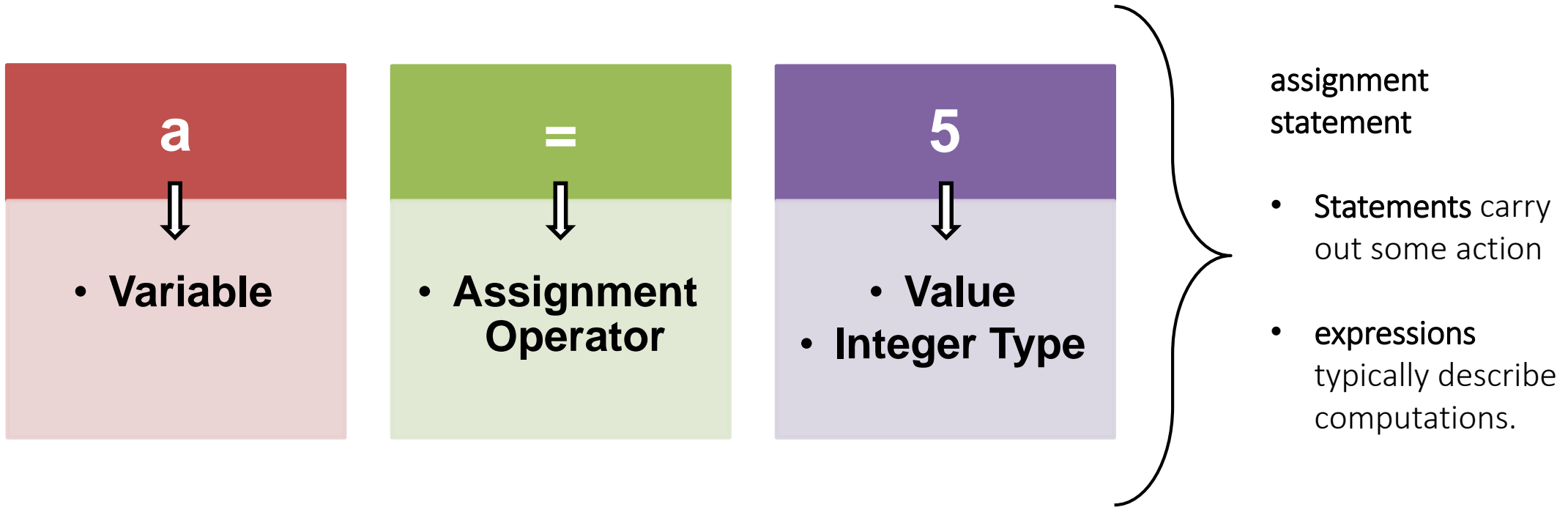
## 1. Basic

- i. Values
- ii. Types
- iii. Variables

## 2. Native to Python

- i. List
- ii. Dictionary

# Data Structures: Basic

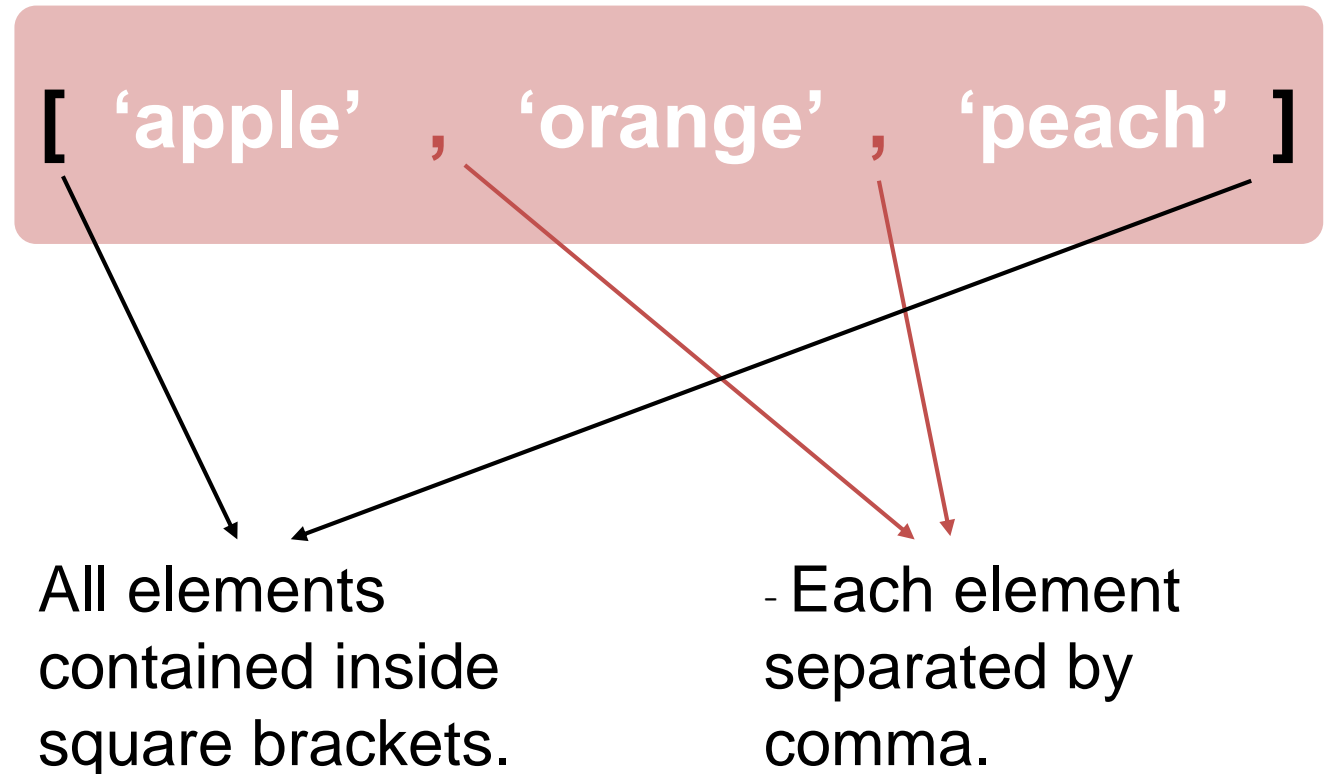


- A program works with values
- Values can be numbers, texts and/or special characters
- Values belong to different [data types](#)

# Data Structures: Native to Python

## 1. LIST

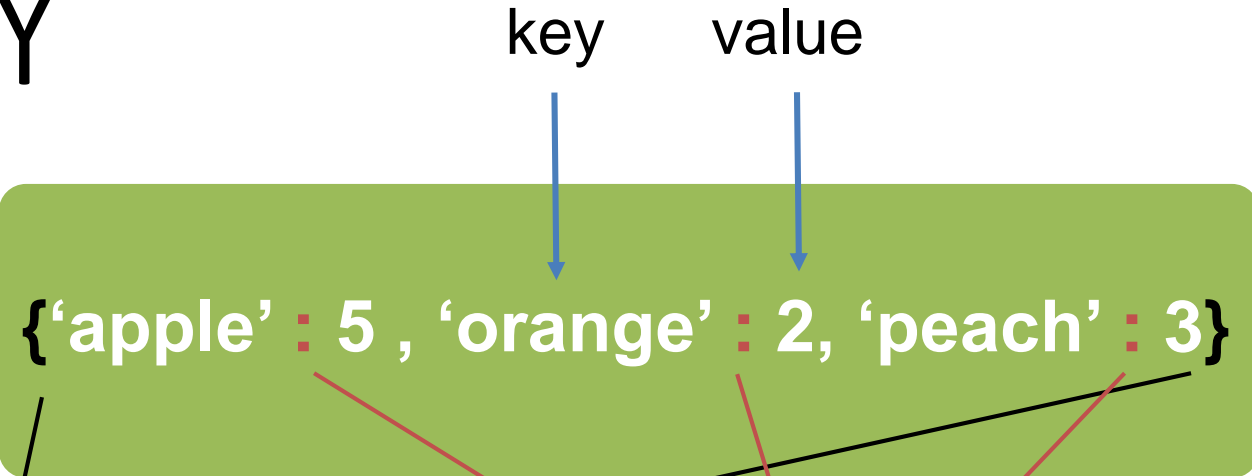
- Mutable
- Ordered
- Sequence of items





# 2. DICTIONARY

- Mutable
- Unordered
- Key-value pairs



All key-value pairs are contained inside curly brackets.

Key and its value are separated by colon.

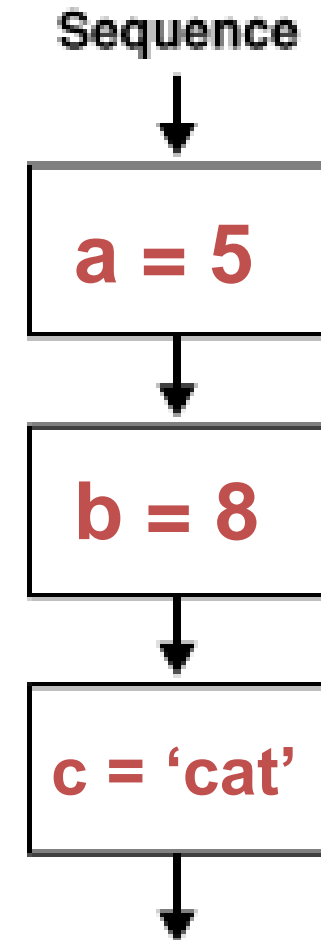
# Programming Structures

# Programming Structures

1. Sequential
2. Iteration
3. Conditional

# 1. SEQUENTIAL

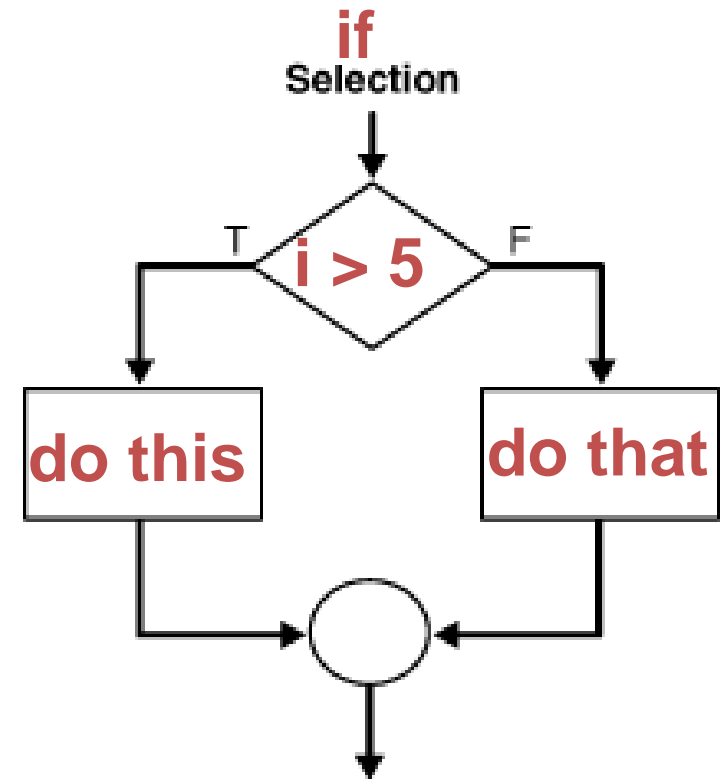
Programs are mostly written sequentially, meaning the first line of program runs first followed by the program in the second line, then the third line and so on.



# 2. Conditional

Programs become more useful when we can change its behavior given a condition is satisfied.

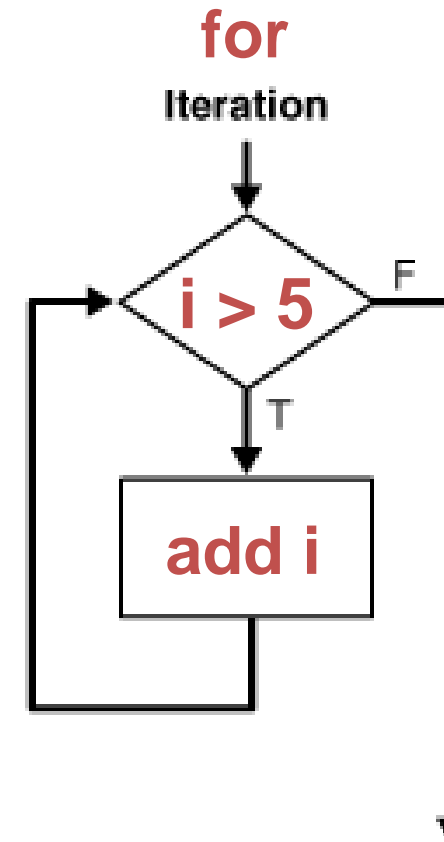
numlist = [4,6,10, 15]



### 3. ITERATION

Programs becomes powerful when the same block of code can be repeatedly executed on either identical tasks or similar tasks

numlist = [4,6,10,15]  
newlist = [6,10,15]



Go to next element  
or end

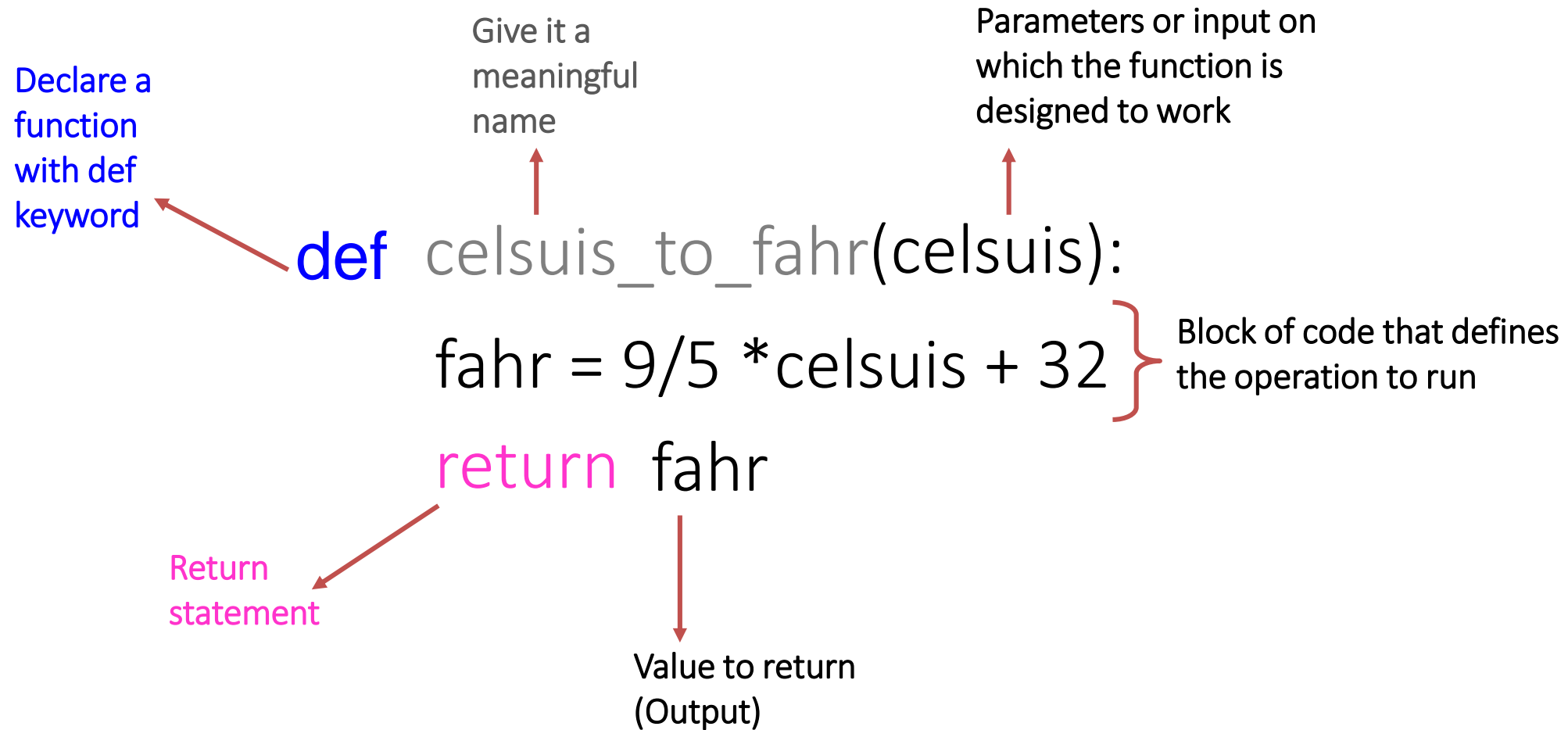
# Functions

# Functions

1. Custom functions
2. Built-in functions
3. Methods
4. Third-party packages



# Functions: Custom Functions



## Functions: Custom Functions

```
def celsuis_to_fahr(celsuis):  
    fahr = 9/5 *celsuis + 32  
    return fahr
```

## Functions: Built-in Functions

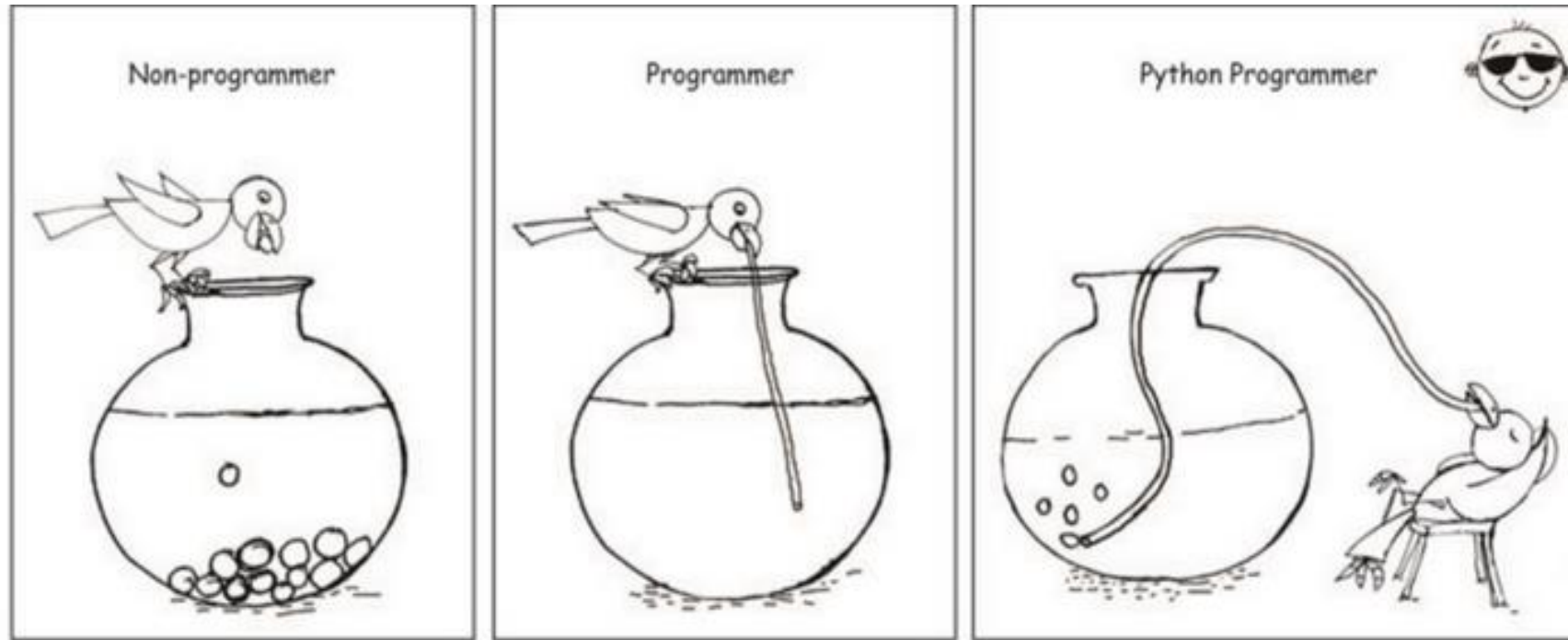
- Python interpreter has a number of functions and types built into it that are always available.
- `print()` is an example of built-in function. It prints the given object to the standard output device (screen) or to the text stream file.
- [Here](#) is the list of Python's built-in functions.

## Functions: Methods

- Functions that are attached to specific class of objects.
- Methods are accessed using the dot expression.
- Methods available to an object can be viewed using "dir" function.

## Functions: Third Party Packages

- Python has an active supporting community of contributors and users who also make their software available for other Python developers to use under its open source license terms.
- The **SciPy** stack is a Python based eco-system of open-source software for mathematics, science, and engineering. In particular, these are some of its core packages that we will use:
  - NumPy
  - Pandas
  - Matplotlib
- **Scikit-Learn** is another Python's go-to package for Machine Learning.



Who wants to become a Python Programmer?

# Questions?

Thank you