Introducing Python

Niti / TDMDAL

Workshop Website : https://tdmdal.github.io/XXXX

Python’s Popularity

- In 2018, Python became the biggest gainer in the Tiobe index of language popularity
- In June 2017, Python became the most visited tag on Stack Overflow
- In 2016, Python became the most frequently taught language in colleges and universities

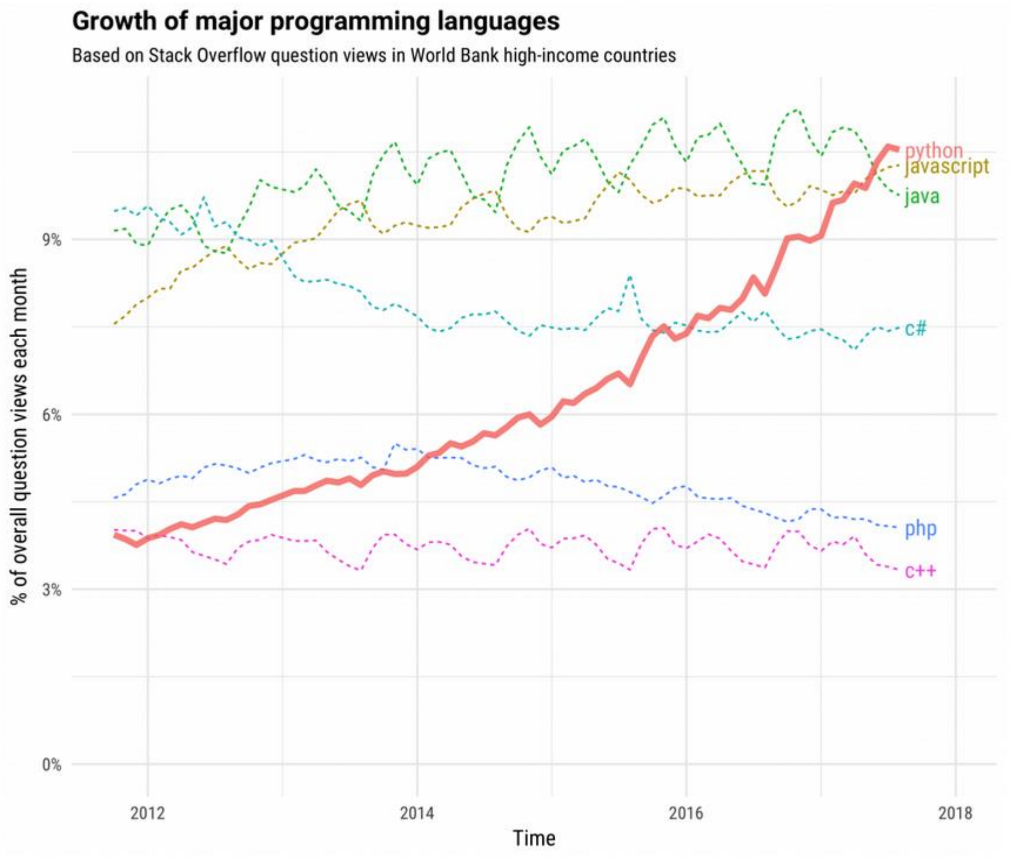


Fig. 1 Six of the ten most-visited Stack Overflow tags in high-income countries from 2012 to 2018 (Source: https://stackoverflow.blog/2017/09/06/incredible-growth-python/)

Python leads in many domains

- Statistical analysis
- Scientific computing
- Machine learning
- Data visualization
- Artificial intelligence

- Others:
  - Scripting & automation
  - Web development
  - Systems testing & prototyping
  - Desktop & mobile applications
  - Education!

Python’s Key Advantages

1. General-purpose language
2. Interpreted language
3. Dynamically-typed language

4. Easy to learn
5. Broad support

6. Data science libraries
    - Machine learning
    - Text analysis
    - Web scrapping

8. Others:
    i. Object-oriented programming
    ii. Functional programming

    iii. Big data application
    iv. Scripting and automation
    v. Rapid prototyping
    vi. Web development

1. General-Purpose Language

- Greatness for many over excellence for one application
- Solve a lot of different problems across a lot of different domains
- No need to switch to another language when writing scientific code
- Becoming better at Python leads to genuine programming skill!

2. Interpreted language

- Ease of Use
- Time
- Portable and Extensible

- Modes of use:
  - Command-line mode
  - Script mode
  - Jupyter notebook

3. Dynamically-typed language

- Different type-systems differ based on when types are checked.
- In Python, types are checked on the fly i.e. during the execution
- Allows type of the variable to change over its lifetime
- This makes testing/debugging slightly faster and easier

4. Easy to Learn

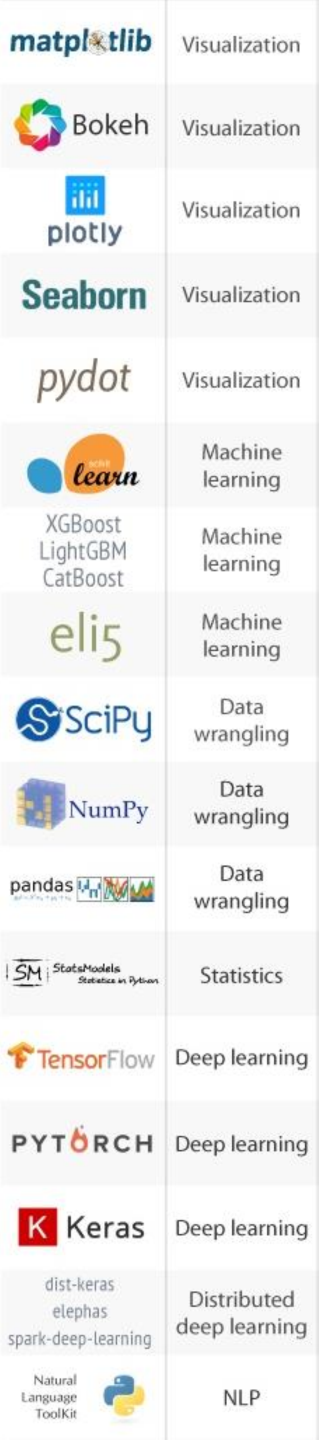
- Simple and straightforward syntax
- Easy to read and closely resembles the English language
- More time solving problem and less time on language complexities
- Can still learn about programming paradigms

5. Broad Support

- Free and open source
- Runs on every major operating systems and platforms
- Contributions from a large community of users
- Well-furnished standard library - https://docs.python.org/3/library/
- Free and seamless interfaces for major libraries and API-powered servers

6. Data Science - Machine Learning

- Leading language of many data scientists
- One of the best collection of machine learning and deep learning libraries
- Numerical engines have attracted academic scholars and private researchers
- Popular data visualization libraries



matplotlib	Visualization
Bokeh	Visualization
plotly	Visualization
Seaborn	Visualization
pydot	Visualization
scikit learn	Machine learning
XGBoost LightGBM CatBoost	Machine learning
eli5	Machine learning
SciPy	Data wrangling
NumPy	Data wrangling
pandas	Data wrangling
SM StatsModels Statistics in Python	Statistics
TensorFlow	Deep learning
PYTORCH	Deep learning
Keras	Deep learning
dist-keras elephas spark-deep-learning	Distributed deep learning
Natural Language ToolKit	NLP

6. Data Science - Text Analysis

- Popular for text mining and Natural Language Processing (NLP)

- **NLTK**
  - been around a long time as an education and research tool
  - provides common natural language functions
- **spaCy**
  - modern library that invested in performance and accuracy benchmarking and tuning
- **Gensim**
  - leading state-of-the-art package for processing text and working with word vector models, a notable advancement in NLP

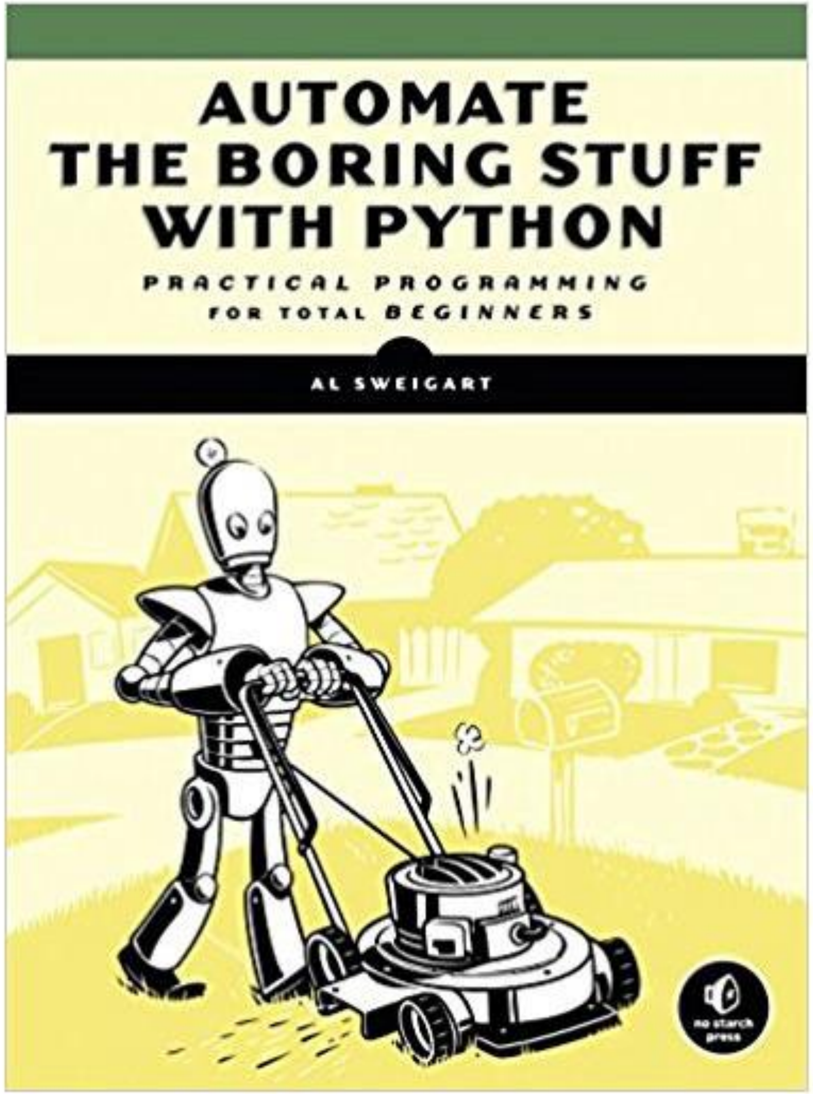
6. Data Science - Web Scraping

- Web scraping is about extracting data from websites.
- Python is one of the most popular language for web scraping
- No need to start from scratch
- Python libraries for web-scrapping to get started:
  - To make web requests - requests
  - To process HTML content - Scrapy and Beautiful Soup
  - Other libraries - lxml, Selenium, etc

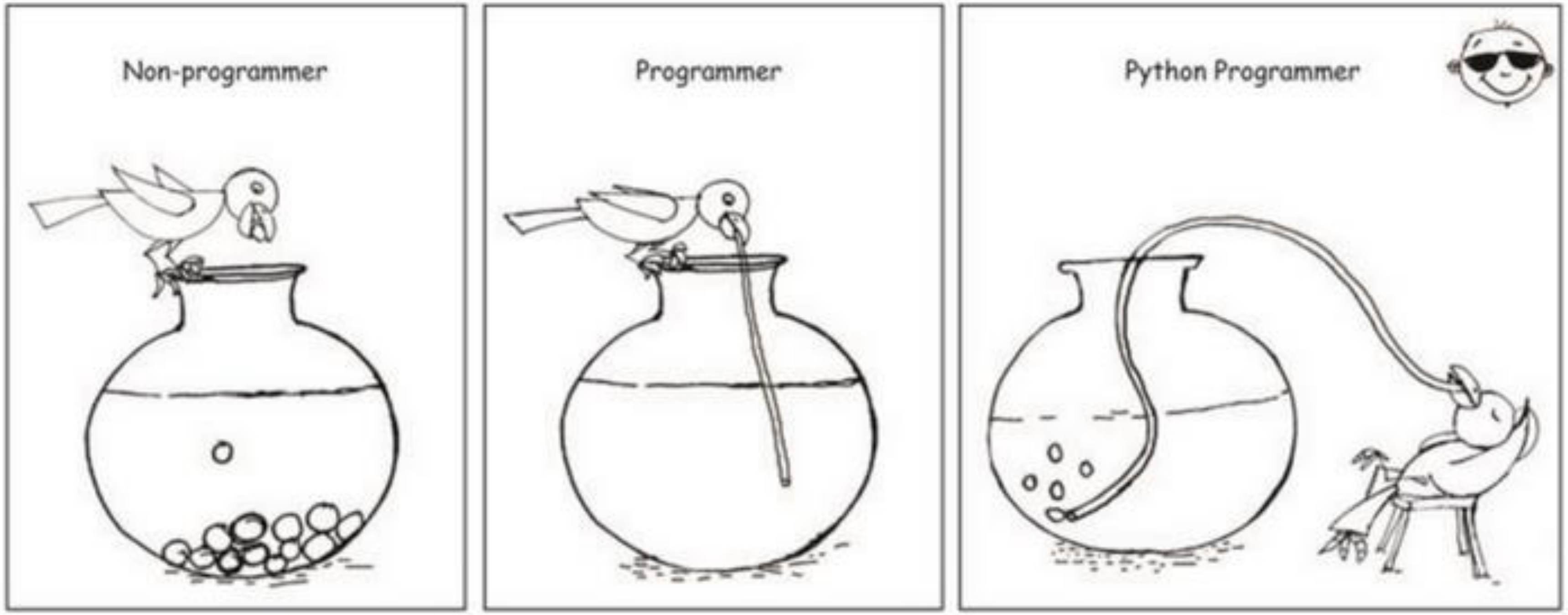
8. Others

- Object-oriented programming
- Functional programming

- Scripting & Automation*
- Big data application
- Rapid Prototyping
- Web-Development
- Desktop & Mobile Applications
- Education!


AUTOMATE THE BORING STUFF WITH PYTHON
PRACTICAL PROGRAMMING FOR TOTAL BEGINNERS
AL SWEIGART
no starch press

Automate Boring Stuff with Python—best book for Python lovers



Who wants to become a Python Programmer?

Session 2 : Agenda

1. Data Structures

- Basic
  - Values and Types
  - Type - String
  - Variables

- Native to Python
  - List
  - Dictionary

2. Operations

- Indexing & Slicing
- Membership
- Arithmetic

- Functions
- Iterations
- Conditionals

Data Structure - Values and Types

- A program works with values
- Values can be numbers or texts, special characters, etc.
- Values belong to different types
  - Numbers are usually of type integer (e.g. 42) or float (e.g. 35.5)
  - Texts are of type string (e.g. ‘Hello World!’)
  - True or False are of type boolean

Data Structure - Strings

- Strings are sequence of character(s)
- Strings are contained by either single or double quotes
- Two strings can be combined by using + operator
- Strings can be repeated by * operator

Data Structure - Variables

- A powerful feature of programming languages is the ability to manipulate variables

- Variable is a name that it refers to a value

- Variable names must follow naming rules

- NameError is a common error

Operation - Functions

- Variables are manipulated using functions (and operators)
- The name of the function is bound to compound operation
- Functions that are attached to specific class of objects are called methods and are accessed using the dot expression
- Python has
  - Built-in functions
  - Third party functions and
  - You can create your own functions

Data Structure - List

- A list is a mutable, ordered sequence of items
- It can be indexed, sliced, and changed
- Each element can be accessed using its position in the list
- Positions are indicated by an integer value called index

Operation – Indexing and Slicing

Indexing

- [ ] operator
- Selects one element from a sequence
- Integer inside [ ] indicates the position of the element

Slicing

- [:] operator
- Selects a subsequence from a sequence
- Integer before and after : indicates start and stop indices
- If no integers are defined then it returns everything

- *Python index starts at 0*
- *A minus sign before index indicates backward indexing*

Operation – Arithmetic

- Elements of lists and lists themselves can be manipulated using
  - +
  - -
  - *

  and other such operators known as arithmetic operators

- Arithmetic operators behave differently on
  - lists and strings vs
  - integers and float

Operation - Iterations

- Iterations are useful for manipulating each item in a list

- “for”statement combined with “in” is most commonly used for iterating over sequence of item

- “in” is a membership operator

Operation - Conditionals

- In order to write useful programs
  - we almost always need the ability to check conditions and
  - change the behavior of program accordingly

- In Python, conditionals can be created with ‘if’ directive

- Conditionals are often combined with comparison operators
  - ==
  - !=
  - >=
  - < , etc

Data Structure - Dictionary

- A mutable, unordered set of key-value pairs separated by :
- Each key must be unique
- To access a given element, refer to it by its key
- Are written using curly brackets
- Each key-value pair is separated from the next with a comma

List vs Dictionary

List

- ordered collection of items
- contains only values

Dictionaries

- don’t have any order
- associates each key with a value

- Use dictionary when there is an unordered set of unique keys that map to values

- Because keys and values linked to each other, the performance will be better than lists when checking membership of an element

Session 3 : Agenda

- NumPy
  - List vs ndarray
- Pandas
- Matplotlib

NumPy

- On its own Python isn't ideal for working with large data
- This is where Numpy comes in:
  - Stands for “Numerical Python”
  - Core of Python’s scientific computing

  - Provides “ndarray” objects i.e. multidimensional arrays
  - Ndarray takes up less space and are faster in performance
  - Supports optimized implementations of arrays such as vectorised operations
  - Bears similarities with R and MATLAB
  - Pandas is built on top of Numpy

List vs ndarray

List

- made for heterogeneous types
- support adding and removing elements
- takes more memory space
- performance can lag

ndarray

- works on homogeneous types
- does not allow removing or adding items
- packed into memory more efficiently
- Performance is usually faster

Pandas

- Most widely used python library in data science
- Provides high-performance tools to extract, clean, transform and analyze your data

- Data structures:
  - 2d table object called “Dataframe”
  - 1d array object called “Series

- Additional functionalities over Numpy:
  - Pivot table
  - Column based computations
  - Plotting graphs

Pandas

- Importing
- Indexing
    - By column names
    - By index numbers
    - By index labels (if applicable)
- Changing data types
- Math Operations
- Resetting Index
- Dropping Columns
- Creating new columns
- Aggregating and Grouping in Pandas
- Visualization
    - Built-in functions
    - Matplotlib
    - Seaborn

End of presentation