

Machine Learning in Production

*Master the art of delivering robust
Machine Learning solutions with MLOps*

Suhas Pote



www.bpbonline.com

Copyright © 2023 BPB Online

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor BPB Online or its dealers and distributors, will be held liable for any damages caused or alleged to have been caused directly or indirectly by this book.

BPB Online has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, BPB Online cannot guarantee the accuracy of this information.

First published: 2023

Published by BPB Online

WeWork

119 Marylebone Road

London NW1 5PU

UK | UAE | INDIA | SINGAPORE

ISBN 978-93-55518-101

www.bpbonline.com

Dedicated to

My parents:

*I am so grateful for your belief in me, and
for the many sacrifices you have made throughout
my life. Your love and support mean the world to me.*

About the Author

Suhas Pote has over eight years of multidisciplinary experience in data science, playing central roles in numerous projects as a technical leader and data scientist, delivering projects using open-source technologies for big companies, including successful projects in South America, Europe, and the United States. He is experienced in client engagement and working collaboratively with different teams. Currently, he is a process manager at Eclerx and is an accomplished postgraduate, having completed a degree in Data Science, Business Analytics, and Big Data. He holds a Bachelor's degree focused on Electronics and Telecommunication Engineering. In the meantime, he successfully got many certifications in data science and related tools. Furthermore, the author participates as a speaker in Data Science conferences and writes technical articles on machine learning and related topics. He also contributes to technical communities worldwide, such as Stack Overflow.

About the Reviewer

Karan Vijay Singh is a passionate Big Data / ML Engineer with extensive professional experience in the domain of Cloud Cost Optimization. With his optimisation skills, he has saved his current organization 0.5M dollar a year in cost. He also has co-authored a research paper in 2019 in the field of cloud computing which has over 270 citations worldwide.

Karan did his Master's in Mathematics (Computer Science) from University of Waterloo in Canada. He is currently working as a Data Engineer at Lightspeed Commerce in Canada.

Acknowledgement

I want to express my deepest gratitude to my family and friends for their unwavering support and encouragement throughout this book's writing, especially my wife, Dr. Archana and my son, Eshansh.

I am also grateful to BPB Publications for their guidance and expertise in bringing this book to fruition. It was a long journey of revising this book, with valuable participation and collaboration of reviewers, technical experts, and editors.

I would also like to acknowledge the valuable contributions of my colleagues and co-worker during many years working in the tech industry, who have taught me so much and provided valuable feedback on my work.

Finally, I would like to thank all the readers who have taken an interest in my book and for their support in making it a reality. Your encouragement has been invaluable.

Preface

Productionizing the machine learning model is a complex task that requires a comprehensive understanding of the latest technologies and CI/CD pipeline. MLOps become increasingly popular in the field of Data Science.

This book is designed to provide a comprehensive guide to building and deploying ML applications with MLOps. It covers a wide range of topics, including the basics of Python programming, Git, Machine Learning life cycle, Docker, and advanced concepts such as packaging Python code for ML models, monitoring, model security, Kubernetes, testing using pytest and the use of CI/CD pipeline for building and deploying robust and scalable ML applications on cloud platforms, including Azure, GCP, and AWS.

Throughout the book, you will learn about the MLOps, various tools, and techniques to deploy ML models. You will also learn how to use them to productionize ML models and applications that are efficient, scalable, and easy to maintain. Additionally, you will learn about best practices and design patterns for MLOps.

This book is intended for data scientists, software developers, data engineers, data analysts, and managers who are new to MLOps and want to learn how to productionize ML models. It is also helpful for experienced data scientists and ML engineers who want to expand their knowledge of these technologies and improve their skills in deploying ML models in production.

With this book, you will gain the knowledge and skills to become proficient in the field of MLOps. I hope you will find this book informative and helpful.

Chapter 1: Python 101 – explains the Python fundamental concepts needed for the reader to equip with the prerequisites required for this book, including Python installation, data structures, control statements, loops, functions, and data manipulation using pandas. This is a quick refresher for those who have worked with Python. If you are a beginner, this chapter will cover the most common Python commands needed to build and deploy ML models in production. It allows the reader to learn fundamental concepts related to the Object-Oriented Programming paradigm using Python.

Chapter 2: Git and GitHub Fundamentals – presents a detailed overview of Git workflow, including common Git commands with practical examples. This is

essential content for the entire book, as this chapter covers fundamental aspects of Git and GitHub that influence technical decisions to build the CI/CD pipelines to deploy ML models in the production environment.

Chapter 3: Challenges in ML Model Deployment – covers the various stages of the ML life cycle, including details on the challenges of each stage. It also covers the common challenges in deploying models in the production environment and how MLOps can help to overcome them. Additionally, the chapter discusses different approaches to deploying ML models in production.

Chapter 4: Packaging ML Models – allows the reader to learn fundamental concepts related to modular programming using the Python language, including Python packaging, dependency management, and good practices of software development to develop stable, readable, and extensible code for robust enterprise applications. Furthermore, the chapter explains the virtual environment, testing code using pytest, serializing, and deserializing ML models. It covers packaging ML models, code, and dependencies so that the package can be installed and consumed on another machine or server.

Chapter 5: MLflow-Platform to Manage the ML Life Cycle – gives special attention to streamlining machine learning development, including tracking experiments, packaging code into reproducible runs, and sharing and deploying models. It demonstrates how to train, deploy, and reuse ML models using MLflow through practical examples based on the use case. This chapter explains the role of MLflow in an ML life cycle.

Chapter 6: Docker for ML – shows the basic concepts of Docker and provides practical examples of common Docker commands with a use case for the reader. Learning these commands allows the reader to package ML code with its dependencies and run the application inside Docker containers. This chapter includes practical examples of Docker objects such as Docker images and containers.

Chapter 7: Build ML Web Apps Using API – explains in detail the most commonly used frameworks for building web-based ML apps using the Python language, including FastAPI, Streamlite, and Flask. It also allows the reader to learn the basics of Gunicorn, NGINX, and APIs, including an explanation of REST APIs, and much more.

Chapter 8: Build Native ML Apps – is dedicated to building native ML applications in Python to give the reader more familiarity with converting Python apps into Windows and Android apps and ways to consume them. This chapter covers practical examples of working with Tkinter, kivy, kivyMD, pyinstaller, and buildozer.

Chapter 9: CI/CD for ML – allows the reader to learn and implement the different stages of the CI/CD pipeline using GitHub and Jenkins, including committing, building, testing, and deploying. This chapter explains the GitHub and Jenkins integrations to build an automated CI/CD pipeline for deploying ML apps using Python and Docker.

Chapter 10: Deploying ML Models on Heroku – will guide the reader in configuring and building a CI/CD pipeline using GitHub Actions to deploy a web-based ML app on the Heroku platform. This chapter also explains three methods for deploying the web app on Heroku: Heroku Git, GitHub integration, and Container registry. It covers practical examples for creating workflow (YAML) files for GitHub Actions, including using tox and pytest for testing the code, and an automated Heroku pipeline for faster deployments.

Chapter 11: Deploying ML Models on Microsoft Azure – explains the step-by-step approach to building CI/CD pipeline using GitHub Actions to deploy web-based ML apps to Azure web services. In the other approach, the reader will learn to build a CI/CD pipeline and deploy web-based scalable ML apps to Azure Container Instances (ACI) and Azure Kubernetes Service (AKS) using Azure DevOps and Azure Machine Learning (AML) Service.

Chapter 12: Deploying ML Models on Google Cloud Platform – Shows how to build an automated CI/CD pipeline to deploy ML models on Google Kubernetes Engine (GKE), without the need to integrate any external tool, service, or platform. This chapter allows the reader to learn and implement Kubernetes functionality to run scalable ML apps using Google Kubernetes Engine (GKE).

Chapter 13: Deploying ML Models on Amazon Web Services – presents an overview of various cloud compute services offered by Amazon Web Services (AWS). This chapter allows the reader to learn and implement an automated CI/CD pipeline with Continuous Training (CT) to deploy scalable enterprise ML apps on Amazon Elastic Container Service (ECS). Additionally, it covers the integration of Application Load Balancer (ALB), Amazon Virtual Private Cloud

(Amazon VPC), and security groups into Amazon Elastic Container Service (ECS) for building robust and secure enterprise ML apps.

Chapter 14: Monitoring and Debugging – is dedicated to monitoring ML and operational metrics, including servers, cost of services, drifts in ML, input data, ML models, and much more. This chapter shows the importance and fundamental concepts of monitoring in the ML life cycle. It also covers practical examples using whylogs and WhyLabs for ML model monitoring, and Prometheus and Grafana for operational monitoring.

Chapter 15: Post-Productionizing ML Models – presents a detailed overview of adversarial machine learning, including different types of adversarial attacks and how to mitigate them. It also covers fundamental concepts of A/B testing and the future scope of MLOps in the industry.

Code Bundle and Coloured Images

Please follow the link to download the *Code Bundle* and the *Coloured Images* of the book:

<https://rebrand.ly/vlv0nzp>

The code bundle for the book is also hosted on GitHub at **<https://github.com/bpbpublications/Machine-Learning-in-Production>**. In case there's an update to the code, it will be updated on the existing GitHub repository.

We have code bundles from our rich catalogue of books and videos available at **<https://github.com/bpbpublications>**. Check them out!

Errata

We take immense pride in our work at BPB Publications and follow best practices to ensure the accuracy of our content to provide with an indulging reading experience to our subscribers. Our readers are our mirrors, and we use their inputs to reflect and improve upon human errors, if any, that may have occurred during the publishing processes involved. To let us maintain the quality and help us reach out to any readers who might be having difficulties due to any unforeseen errors, please write to us at :

errata@bpbonline.com

Your support, suggestions and feedbacks are highly appreciated by the BPB Publications' Family.

Did you know that BPB offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.bpbonline.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at :

business@bpbonline.com for more details.

At www.bpbonline.com, you can also read a collection of free technical articles, sign up for a range of free newsletters, and receive exclusive discounts and offers on BPB books and eBooks.

Piracy

If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at business@bpbonline.com with a link to the material.

If you are interested in becoming an author

If there is a topic that you have expertise in, and you are interested in either writing or contributing to a book, please visit www.bpbonline.com. We have worked with thousands of developers and tech professionals, just like you, to help them share their insights with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions. We at BPB can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about BPB, please visit www.bpbonline.com.

Join our book's Discord space

Join the book's Discord Workspace for Latest updates, Offers, Tech happenings around the world, New Release and Sessions with the Authors:

<https://discord.bpbonline.com>



Table of Contents

1. Python 101.....	1
Introduction.....	1
Structure.....	1
Objectives.....	2
Install Python	2
<i>On Windows and Mac OS</i>	<i>2</i>
<i>On Linux</i>	<i>2</i>
Install Anaconda.....	2
Install code editor	3
Hello World!	3
<i>Execution of Python file.....</i>	<i>3</i>
Data structures	4
<i>Common data structures</i>	<i>4</i>
Control statements and loops	6
Functions.....	8
Object Oriented Programming (OOP).....	9
Numerical Python (NumPy).....	10
Pandas	12
Conclusion.....	15
Points to remember	15
Multiple choice questions.....	16
<i>Answers</i>	<i>16</i>
Questions	16
Key terms.....	16
2. Git and GitHub Fundamentals.....	17
Introduction.....	17
Structure.....	17
Objectives.....	18
Git concepts.....	18

<i>Git + Hub = GitHub</i>	19
Common Git workflow.....	19
Install Git and create a GitHub account.....	20
<i>Linux (Debian/Ubuntu)</i>	20
<i>Git for all platforms</i>	20
<i>GUI clients</i>	20
<i>Create a GitHub account</i>	20
Common Git commands.....	21
<i>Setup</i>	21
<i>New repository</i>	21
<i>Update</i>	21
<i>Changes</i>	21
<i>Revert</i>	22
Let's Git.....	22
<i>Configuration</i>	22
<i>Initialize the Git repository</i>	22
<i>Check Git status</i>	22
<i>Add a new file</i>	23
Conclusion.....	26
Points to remember.....	26
Multiple choice questions.....	26
<i>Answers</i>	27
3. Challenges in ML Model Deployment	29
Introduction.....	29
Structure.....	29
Objectives.....	30
ML life cycle.....	30
<i>Business impact</i>	31
<i>Data collection</i>	31
<i>Data preparation</i>	32
<i>Feature engineering</i>	32
<i>Build and train the model</i>	33

<i>Test and evaluate</i>	34
<i>Model deployment</i>	34
<i>Monitoring and optimization</i>	35
Types of model deployment.....	35
<i>Batch predictions</i>	35
<i>Web service/REST API</i>	36
<i>Mobile and edge devices</i>	36
<i>Real-time</i>	37
Challenges in deploying models in the production environment.....	37
<i>Team coordination</i>	37
<i>Data-related challenges</i>	38
<i>Portability</i>	38
<i>Scalability</i>	38
<i>Robustness</i>	38
<i>Security</i>	39
MLOps.....	39
Benefits of MLOps.....	40
<i>Efficient management of ML life cycle</i>	41
<i>Reproducibility</i>	41
<i>Automation</i>	41
<i>Tracking and feedback loop</i>	42
Conclusion.....	42
Points to remember.....	42
Multiple choice questions.....	43
<i>Answers</i>	43
Questions.....	43
Key terms.....	43
4. Packaging ML Models.....	45
Introduction.....	45
Structure.....	45
Objectives.....	46
Virtual environments.....	46

Requirements file	47
Serializing and de-serializing ML models	48
Testing Python code with pytest	48
<i>pytest fixtures</i>	49
Python packaging and dependency management	49
<i>Modular programming</i>	50
<i>Module</i>	50
<i>Package</i>	50
Developing, building, and deploying ML packages	51
<i>Business problem</i>	52
<i>Data</i>	52
<i>Building the ML model</i>	53
<i>Developing the package</i>	53
Set up environment variables and paths	70
<i>Build the package</i>	71
<i>Install the package</i>	71
<i>Package usage with example</i>	73
Conclusion	74
Points to remember	74
Multiple choice questions	74
<i>Answers</i>	75
Questions	75
Key terms	75
5. MLflow-Platform to Manage the ML Life Cycle	77
Introduction	77
Structure	77
Objectives	78
Introduction to MLflow	78
<i>Set up your environment and install MLflow</i>	79
<i>Miniconda installation</i>	79
<i>MLflow components</i>	82
MLflow tracking	83

<i>Log data into the run</i>	84
MLflow projects	94
MLflow models	97
MLflow registry	100
<i>Set up the MySQL server for MLflow</i>	101
<i>Start the MLflow server</i>	103
Conclusion	109
Points to remember	109
Multiple choice questions	109
<i>Answers</i>	110
Questions	110
6. Docker for ML	111
Introduction	111
Structure	111
Objectives	112
Introduction to Docker	112
<i>It works on my machine!</i>	112
<i>Long setup</i>	112
<i>Setting up your environment and installing Docker</i>	113
<i>Docker installation</i>	113
<i>Uninstall old versions</i>	113
<i>Install Docker Engine</i>	114
<i>Docker compose</i>	115
Hello World with Docker	116
Docker objects	117
<i>Dockerfile</i>	117
<i>Docker image</i>	117
<i>Docker containers</i>	118
<i>Docker container networking</i>	118
Create a Dockerfile	119
Build a Docker image	120
Run a Docker container	120

Dockerize and deploy the ML model	122
Common Docker commands	127
Conclusion.....	127
Points to remember	127
Multiple choice questions.....	128
<i>Answers</i>	128
Questions	128
7. Build ML Web Apps Using API	129
Introduction.....	129
Structure.....	129
Objectives.....	130
Rest APIs	130
FastAPI.....	131
Streamlit.....	139
Flask.....	143
<i>Gunicorn</i>	150
<i>NGINX</i>	150
Conclusion.....	154
Points to remember	154
Multiple choice questions.....	155
<i>Answers</i>	155
8. Build Native ML Apps	157
Introduction.....	157
Structure.....	157
Objectives.....	158
Introduction to Tkinter	158
<i>Hello World app using Tkinter</i>	159
Build an ML-based app using Tkinter.....	160
<i>Tkinter app</i>	163
Convert Python app into Windows EXE file	167
Build an ML-based app using kivy and kivyMD	170
<i>KivyMD app</i>	173

Convert the Python app into an Android app	179
Conclusion.....	180
Points to remember	181
Multiple choice questions.....	181
<i>Answers</i>	181
Questions	181
9. CI/CD for ML	183
Introduction.....	183
Structure.....	183
Objectives.....	184
CI/CD pipeline for ML.....	184
Continuous Integration (CI).....	185
Continuous Delivery/Deployment (CD).....	186
Continuous Training (CT)	186
Introduction to Jenkins	187
<i>Installation</i>	187
Build CI/CD pipeline using GitHub, Docker, and Jenkins.....	189
<i>Develop codebase</i>	189
<i>Create a Personal Access Token (PAT) on GitHub</i>	196
<i>Create a webhook on the GitHub repository</i>	197
Configure Jenkins	199
Create CI/CD pipeline using Jenkins.....	202
<i>Stage 1: 1-GitHub-to-container</i>	203
<i>Stage 2: 2-training</i>	205
<i>Stage 3: 3-testing</i>	207
<i>Stage 4: 4-deployment-status-email</i>	210
Conclusion.....	217
Points to remember	217
Multiple choice questions.....	218
<i>Answers</i>	218
Questions	218

10. Deploying ML Models on Heroku	219
Introduction.....	219
Structure.....	219
Objectives.....	220
Heroku.....	220
Setting up Heroku	221
Deployment with Heroku Git.....	222
Deployment with GitHub repository integration.....	222
<i>REVIEW APPS</i>	223
<i>STAGING</i>	223
<i>PRODUCTION</i>	224
<i>Heroku Pipeline flow</i>	224
Deployment with Container Registry.....	225
GitHub Actions	225
<i>Configuration</i>	226
CI/CD pipeline using GitHub Actions and Heroku.....	227
Conclusion.....	246
Points to remember	246
Multiple choice questions.....	247
<i>Answers</i>	247
Questions	247
11. Deploying ML Models on Microsoft Azure.....	249
Introduction.....	249
Structure.....	249
Objectives.....	250
Azure	250
Set up an Azure account.....	251
Deployment using GitHub Actions	251
<i>Infrastructure setup</i>	263
<i>Azure Container Registry</i>	263
<i>Azure App Service</i>	267
<i>GitHub Actions</i>	270

<i>Service principal</i>	273
<i>Configure Azure App Service to use GitHub Actions for CD</i>	274
Deployment using Azure DevOps and Azure ML	278
Azure Machine Learning (AML) service.....	279
<i>Workspace</i>	280
<i>Experiments</i>	280
<i>Runs</i>	280
Configure CI pipeline.....	290
Configure CD pipeline.....	292
Conclusion.....	299
Points to remember	299
Multiple choice questions.....	299
<i>Answers</i>	300
Questions	300
12. Deploying ML Models on Google Cloud Platform.....	301
Introduction.....	301
Structure.....	301
Objectives.....	302
Google Cloud Platform (GCP).....	302
<i>Set up the GCP account</i>	303
Cloud Source Repositories	304
Cloud Build	309
Container Registry.....	311
Kubernetes.....	312
Google Kubernetes Engine (GKE).....	313
Deployment using Cloud Shell – Manual Trigger	316
CI/CD pipeline using Cloud Build.....	317
<i>Create a trigger in Cloud Build</i>	318
Conclusion.....	323
Points to remember	323
Multiple choice questions.....	323
<i>Answers</i>	324

Questions	324
13. Deploying ML Models on Amazon Web Services.....	325
Introduction.....	325
Structure.....	325
Objectives.....	326
Introduction to Amazon Web Services (AWS).....	326
<i>AWS compute services</i>	326
<i>Amazon Elastic Compute Cloud (EC2)</i>	327
Amazon Elastic Container Service (ECS).....	327
Amazon Elastic Kubernetes Service (EKS)	327
Amazon Elastic Container Registry (ECR)	327
<i>AWS Fargate</i>	328
<i>AWS Lambda</i>	328
Amazon SageMaker	328
Set up an AWS account	329
AWS CodeCommit	331
<i>Continuous Training</i>	336
Amazon Elastic Container Registry (ECR)	336
<i>Docker Hub rate limit</i>	337
AWS CodeBuild	339
<i>Attach container registry access to CodeBuild's service role</i>	345
Amazon Elastic Container Service (ECS).....	346
<i>AWS ECS deployment models</i>	347
<i>EC2 instance</i>	347
<i>Fargate</i>	348
<i>Task definition</i>	349
<i>Running task with the task definition</i>	350
<i>Amazon VPC and subnets</i>	351
<i>Load balancing</i>	352
<i>Target group</i>	353
<i>Security Groups</i>	356
<i>Application Load Balancers (ALB)</i>	358

<i>Service</i>	363
CI/CD pipeline using CodePipeline	369
<i>AWS CodePipeline</i>	371
Monitoring.....	378
Conclusion.....	379
Points to remember	379
Multiple choice questions.....	380
<i>Answers</i>	380
Questions	380
14. Monitoring and Debugging.....	381
Introduction.....	381
Structure.....	381
Objectives.....	382
Importance of monitoring	382
Fundamentals of ML monitoring	383
Metrics for monitoring your ML system	385
Drift in ML.....	386
<i>Types of drift in ML</i>	386
<i>Techniques to detect the drift in ML</i>	388
<i>Addressing the drift in ML</i>	389
Operational monitoring with Prometheus and Grafana.....	390
ML model monitoring with whylogs and WhyLabs.....	402
<i>whylogs</i>	403
<i>Constraints for data quality validation</i>	404
<i>WhyLabs</i>	407
Conclusion.....	416
Points to remember	416
Multiple choice questions.....	417
<i>Answers</i>	417
Questions	417

15. Post-Productionizing ML Models	419
Introduction.....	419
Structure.....	419
Objectives.....	420
Bridging the gap between the ML model and the creation of business value	420
Model security.....	420
<i>Adversarial attack</i>	421
<i>Data poisoning attack</i>	422
<i>Distributed Denial of Service attack (DDoS)</i>	422
<i>Data privacy attack</i>	422
<i>Mitigate the risk of model attacks</i>	423
A/B testing	423
MLOps is the future	424
Conclusion.....	425
Points to remember	425
Multiple choice questions.....	425
<i>Answers</i>	426
Questions	426
Index	427-434

CHAPTER 1

Python

101

Introduction

Python 101 guides you with everything from the installation of Python to data manipulation in Pandas DataFrame. In a nutshell, this chapter is designed to equip you with the prerequisites required for this book. It is a quick refresher for those who have worked with Python. And if you are a beginner, this chapter is going to cover the most common Python commands required to build and deploy machine models.

Structure

- This chapter covers the following topics:
- Installation of Python
- Hello World!
- Data structures
- Control statements and loops
- Functions
- OOPs
- NumPy
- Pandas

Objectives

After studying this chapter, you should be able to install Python on your machine, and store and manage data using common data structures in Python. You should be able to use Python's data processing packages for data manipulation, and you should also know how to create classes, methods, and objects.

Install Python

Make sure the Python version you are installing is compatible with the libraries and applications you will work on. Also, maintain the same Python version throughout the project to avoid any errors or exceptions. Here, you will install Python version 3.6.10.

On Windows and Mac OS

Here's the link to install Python 3.6.10:

<https://www.python.org/downloads/release/python-3610/>

On Linux

Ubuntu 16.10 and 17.04:

```
sudo apt update
sudo apt install python3.6
```

Ubuntu 17.10, 18.04 (Bionic) and onward:

Ubuntu 17.10 and 18.04 already come with Python 3.6 as default. Just run `python3` in the terminal to invoke it.

Install Anaconda

Anaconda Individual Edition contains conda and Anaconda Navigator, as well as Python and hundreds of scientific packages. When you install Anaconda, all of these will also get installed.

<https://docs.anaconda.com/anaconda/install/>

Note: Review the system requirements listed on the site before installing Anaconda Individual Edition.

Install code editor

Any of the following code editors can be chosen; however, the preferred one is Visual Studio Code:

Visual Studio Code (<https://code.visualstudio.com>)

Sublime Text (<https://www.sublimetext.com>)

Notepad++ (<https://notepad-plus-plus.org/downloads/>)

Hello World!

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its simple, easy-to-learn syntax emphasizes readability and therefore, reduces the cost of program maintenance. Python supports modules and packages, which encourages modular programming and code reusability. It is developed under an OSI-approved open-source license, making it freely usable and distributable, even for commercial use. The **Python Package Index (PyPI)** hosts thousands of third-party modules for Python.

Open the terminal and type a **python** to open the Python console.

Now you are in the Python console.

Example: 1.1 Hello world in Python

```
1. print("Hello World")
2. Hello World
```

To come out of the console you can use:

```
1. exit()
```

Execution of Python file

Let's create a file named `hello_world.py` and add the `print("Hello World")` to it.

Now, run the file in the terminal:

```
python hello_world.py
```

You should get the following output:

```
Hello World
```

Data structures

Data structures are nothing but particular ways of storing and managing data in memory so that it can be easily accessed and modified later. Python comes with a comprehensive set of data structures, which play an important role in programming because they are reusable, easily accessible, and manageable.

Common data structures

You will study some of the common data structures in this section.

Array

Arrays are collections of homogeneous items. One can use the same data type in a single array.

Example: 1.2 Array data type

```
1. import array as arr
2.
3. my_array = arr.array("i", (1, 2, 3, 4, 5))
4.
5. print(my_array)
6. array('i', [1, 2, 3, 4, 5])
7.
8. print(my_array[1])
9. 2
```

Dictionary

Dictionaries are defined as comma separated **key:value** pairs enclosed in curly braces.

Example: 1.3 Dictionary data type

```
1. my_dict = {'name': 'Adam', 'emp_id': 3521, 'dept': 'Marketing'}
2. print(my_dict['name'])
3. Adam
```

List

It is a collection of heterogeneous items enclosed in square brackets. One can use the same or different data types in a list.