Data Engineering and Data Analytics Journey

A table of data analysis

AI-generated content may be incorrect.

# Week 1: Introduction to Data Engineering, Data Analytics, and Python Basics

## What is Data Engineering?

Data Engineering concerns itself with the mechanics for the flow and access of data. Its goal is to make quality data available for fact-finding and data-driven decision making. Data engineers’ duty is to collect data which includes extracting, integrating, and organizing data from disparate (different) data sources.

## What is Data Analytics?

Data analytics is the process of collecting, cleaning, analysing, and interpreting large datasets to identify patterns, trends, and correlations, ultimately extracting meaningful insights that can be used to inform business decisions and solve problems.

**Types of Data Analytics**

* Descriptive analytics - tell us what happened.
* Diagnostic analytics - tell us why something happened.
* Predictive analytics - tell us what will likely happen in the future.
* Prescriptive analytics - tell us how to act.

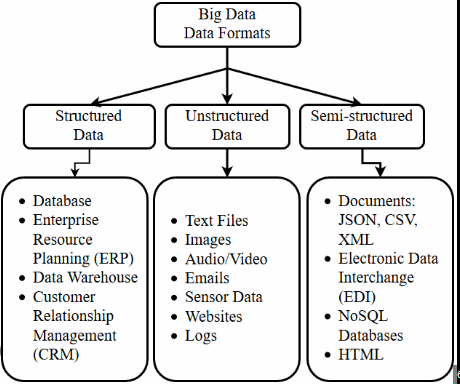
## Data Sources and Data Types

Data Source - In data engineering and analytics, "data sources" refer to the various locations where raw data originates, including

* Relational databases,
* NoSQL databases,
* Data warehouses,
* Data lakes,
* APIs,
* Log files,
* IoT devices,
* Social media platforms,
* Web analytics,
* Transactional systems

These data sources can be structured, semi-structured and unstructured.

Data Types



Structured Data

* Structured Data are data that have a well-defined and rigid structure and can be formatted in row and column format (Tabular Format). Usually, the row represents the individual data(records) and the column represents the attribute of data. Examples include:
  + Mobile Numbers
  + Data about a currency
  + Data stored in relational databases
  + Semi-Structured Data

Semi-structured data

* Data that may have a defined structure, but they may or may not follow the structure.
* These Data are usually stored in Non-Relational Databases Such as MongoDB, Redis, and Neo4j. Examples include:
  + JSON Files
  + XML Files
  + Name of the user (A user may or may not have a middle name)
  + Data from APIs.
  + Metadata about Any type of File
  + Emails (An Email may or may not have a body)

Unstructured Data

* Unstructured data are data that don’t have a well-defined structure and cannot be formatted in a row and column format.
* These data are stored in file systems and other storage.
* Usually, Images and Video Files are classified as Unstructured Data. Examples include
* Image and Video Files
* Voice Recordings
* Flat Files
* Social Media Posts
* Medical Record (Such as Scan Documents)
* Any legal document (not all legal documents will have the same format, so it is unstructured data)

## Introduction to Python for Data Science

### Python basics: variables, data types, loops, and functions

**Variables**

Variables are containers for storing data values. They act as placeholders for values. Thus, a variable is created the moment you first assign a value to it. A variable can either be global or local, global variables can be accessed by all functions defined and a local variable can only be accessed by a function in which it is initialized in.

* Variable Names
  + A variable can have a short name (like x and y) or a more descriptive name (age, carname, total\_volume). Rules for Python variables:
  + A variable name must start with a letter or the underscore character
  + A variable name cannot start with a number
  + A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and)
  + Variable names are case-sensitive (age, Age and AGE are three different variables)
  + A variable name cannot be any of the Python keywords.

**Data Types**

|  |  |
| --- | --- |
| Type | Name in Python |
| Numeric Types | int, float, complex |
| Sequence Types | list, tuple, range |
| Mapping Type | dict |
| Boolean Type | bool |
| Set Types | set, frozenset |
| Binary Types | bytes, bytearray, memoryview |
| Text Type | str |
| None Type | NoneType |

**Conditional Statements**

Conditional statements in Python are used to execute a specific block of code based on the truth value of a condition.

* IF
* IF-ELSE
* NESTED IF-ELSE

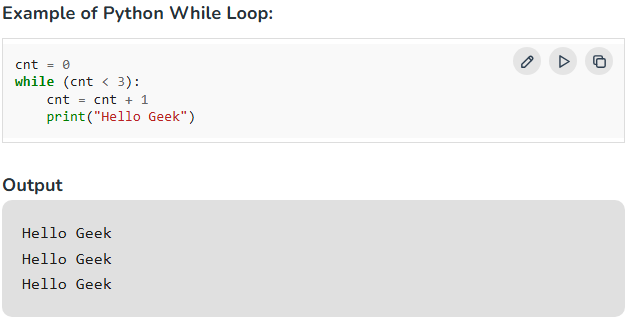
**Loops**

Loops in Python are used to repeat actions efficiently. The main types are For loops (counting through items) and While loops (based on conditions). Additionally, Nested Loops allow looping within loops for more complex tasks. While all the ways provide similar basic functionality, they differ in their syntax and condition-checking time.

* For loops
  + used for sequential traversal. For example: traversing a list or string or array etc.



* While - In Python, a while loop is used to execute a block of statements repeatedly until a given condition is satisfied. When the condition becomes false, the line immediately after the loop in the program is executed.



Nested Loops

* Python programming language allows to use one loop inside another loop which is called [nested loop](https://www.geeksforgeeks.org/python-nested-loops/).



There is more that we can learn about conditional statements and loops, [here](https://www.geeksforgeeks.org/loops-in-python/?ref=lbp).

**Functions**

Python Functions are a block of statements that return the specific task. The idea is to put some commonly or repeatedly done tasks together and make a function so that instead of writing the same code again and again for different inputs, we can do the function calls to reuse code contained in it repeatedly.

* A function is a block of code which only runs when it is called.
* You can pass data, known as parameters, into a function.
* A function can return data as a result.
* Functions increase Code Readability
* Functions increase Code Reusability

In Python a function is defined using the def keyword:

A white rectangular object with a black border

AI-generated content may be incorrect.

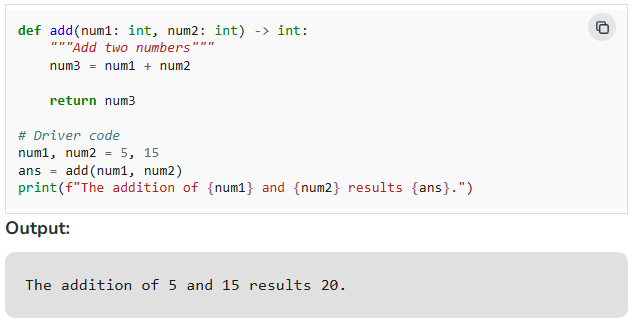
Calling a function in Python.

A white background with black text

AI-generated content may be incorrect.

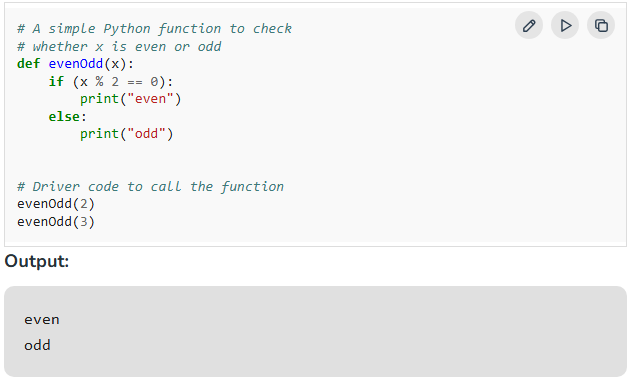
The output of the above code will be “Welcome to GFG”.

A function can accept parameters and return values as results.



**Python Functions Arguments**

* Arguments are the values passed inside the parenthesis of the function. A function can have any number of arguments separated by a comma.



We can learn more about Python arguments [here](https://www.geeksforgeeks.org/python-functions/?ref=header_outind).

## Working with Jupyter Notebooks

Jupyter Notebook is a free, open-source web application that allows users to create and share interactive documents that combine code, text, images, and other resources.

**Features**

Jupyter Notebook supports over 40 programming languages, including Python, R, and Scala. It allows users to:

* Run code and display results in real time
* Toggle between raw code and its rendered output
* Collaborate with others by viewing and interacting with the same notebook
* Add comments and feedback directly within a notebook
* Convert notebooks to standard output formats like HTML, PDF, Markdown, and Python

**Uses**

Jupyter Notebook is used for a variety of data science tasks, including:

* Exploratory data analysis (EDA)
* Data cleaning and transformation
* Data visualization
* Statistical modelling
* Machine learning
* Deep learning

**Components**

* A [Jupyter Notebook](https://jupyter.org/) has two components: a front-end web page and a back-end kernel. The kernel runs interactive code in a specific programming language and returns output to the user.

## Python Project

GitHub Link

# Week 2