1. **diff between python 2 and python 3**

| **Comparison Parameter** | **Python 2** | **Python 3** |
| --- | --- | --- |
| Year of Release | Python 2 was released in the year 2000. | Python 3 was released in the year 2008. |
| “Print” Keyword | In Python 2, print is considered to be a statement and not a function. | In Python 3, print is considered to be a function and not a statement. |
| Storage of Strings | In Python 2, strings are stored as ASCII by default. | In Python 3, strings are stored as UNICODE by default. |
| Division of Integers | On the division of two integers, we get an integral value in Python 2. For instance, 7/2 yields 3 in Python 2. | On the division of two integers, we get a floating-point value in Python 3. For instance, 7/2 yields 3.5 in Python 3. |
| Exceptions | In Python 2, exceptions are enclosed in notations. | In Python 3, exceptions are enclosed in parentheses. |
| Variable leakage | The values of global variables do change in Python 2 if they are used inside a for-loop. | The value of variables never changes in Python 3. |
| Iteration | In Python 2, the xrange() function has been defined for iterations. | In Python 3, the new Range() function was introduced to perform iterations. |
| Ease of Syntax | Python 2 has more complicated syntax than Python 3. | Python 3 has an easier syntax compared to Python 2. |
| Libraries | A lot of libraries of Python 2 are not forward compatible. | A lot of libraries are created in Python 3 to be strictly used with Python 3. |
| Usage in today’s times | Python 2 is no longer in use since 2020. | Python 3 is more popular than Python 2 and is still in use in today’s times. |
| Backward compatibility | Python 2 codes can be ported to Python 3 with a lot of effort. | Python 3 is not backward compatible with Python 2. |
| Application | Python 2 was mostly used to become a DevOps Engineer. It is no longer in use after 2020. | Python 3 is used in a lot of fields like Software Engineering, Data Science, etc. |

1. **Shallo copy vs deep copy**

| **Shallow Copy** | **Deep Copy** |
| --- | --- |
| Shallow Copy stores the references of objects to the original memory address. | Deep copy stores copies of the object’s value. |
| Shallow Copy reflects changes made to the new/copied object in the original object. | Deep copy doesn’t reflect changes made to the new/copied object in the original object. |
| Shallow Copy stores the copy of the original object and points the references to the objects. | Deep copy stores the copy of the original object and recursively copies the objects as well. |
| Shallow copy is faster. | Deep copy is comparatively slower. |

1. **job /task scheduling**

The Task Scheduler **enables you to automatically perform routine tasks on a chosen computer**. Task Scheduler does this by monitoring whatever criteria you choose (referred to as triggers) and then executing the tasks when those criteria are met.

1. **Email sending in python**
2. **What is python module package?**

**Module:** The **module** is a simple Python file that contains collections of functions and global variables and with having a .py extension file. It is an executable file

def myModule(name):

    print("This is My Module : "+ name)

Save the code in file called demo\_module.py

Import module named demo\_module and call myModule function inside it.

import demo\_module

demo\_module.myModule("Math")

or

from demo\_module import myModule

**Package:** The **package** is a simple directory having collections of modules. This directory contains Python modules and also having [\_\_init\_\_.py](https://www.geeksforgeeks.org/__init__-in-python/) file by which the interpreter interprets it as a Package. The package is simply a namespace. The package also contains sub-packages inside it.

Student(Package)

| \_\_init\_\_.py (Constructor)

| details.py (Module)

| marks.py (Module)

| collegeDetails.py (Module)

**Library:** The **library** is having a collection of related functionality of codes that allows you to perform many tasks without writing your code. It is a reusable chunk of code that we can use by importing it in our program, we can just use it by importing that library and calling the method of that library with period(.).

Importing pandas library and call read\_csv method using alias of pandas i.e. pd.

**import** pandas as pd

df **=** pd.read\_csv("file\_name.csv")

1. **overview of pythons some imp libraries**
2. TensorFlow : This library was developed by Google in collaboration with Brain Team. TensorFlow is a part of almost every Google application for machine learning. library for writing new algorithms .
3. Scikit-lear: It is a Python library is associated with NumPy and SciPy. It is considered as one of the best libraries for working with complex data.
4. Numpy: Numpy is considered as one of the most popular machine learning library in Python.TensorFlow and other libraries uses Numpy internally for performing multiple operations on Tensors. Array interface is the best and the most important feature of Numpy. This interface can be utilized for expressing images, sound waves, and other binary raw streams as an array of real numbers in N-dimensional.
5. Keras: is considered as one of the coolest machine learning libraries in Python. It provides an easier mechanism to express neural networks. Keras also provides some of the best utilities for compiling models, processing data-sets, visualization of graphs, and much more. In the backend, Keras uses either Theano or TensorFlow internally.
6. pyTorch: PyTorch is the largest machine learning library that allow developers to perform tensor computations wan ith acceleration of GPU, creates dynamic computational graphs, and calculate gradients automatically. Other than this, PyTorch offers rich APIs for solving application issues related to neural networks. This machine learning library is based on Torch
7. LightGBM: Gradient Boosting is one of the best and most popular machine learning library, which helps developers in building new algorithms by using redefined elementary models and namely decision trees. These libraries are LightGBM, XGBoost, and CatBoost. All these libraries are competitors that helps in solving a common problem and can be utilized in almost the similar manner
8. Eli5: Most often the results of machine learning model predictions are not accurate, and Eli5 machine learning library built in Python helps in overcoming this challenge. It is a combination of visualization and debug all the machine learning models and track all working steps of an algorithm.
9. SciPy: is a machine learning library for application developers and engineers. However, you still need to know the difference between SciPy library and SciPy stack. SciPy library contains modules for optimization, linear algebra, integration, and statistics.
10. Theano: is a computational framework machine learning library in Python for computing multidimensional arrays. Theano works similar to TensorFlow, but it not as efficient as TensorFlow. Because of its inability to fit into production environments.
11. Pandas: is a machine learning library in Python that provides data structures of high-level and a wide variety of tools for analysis. One of the great feature of this library is the ability to translate complex operations with data using one or two commands. Pandas have so many inbuilt methods for grouping, combining data, and filtering, as well as time-series functionality.
12. **important modules in python**
13. **Range vs xrange**

| **range()** | **xrange()** |
| --- | --- |
| Returns a list of integers. | Returns a generator object. |
| Execution speed is slower | Execution speed is faster. |
| Takes more memory as it keeps the entire list of elements in memory. | Takes less memory as it keeps only one element at a time in memory. |
| All arithmetic operations can be performed as it returns a list. | Such operations cannot be performed on xrange(). |
| In python 3, xrange() is not supported. | In python 2, xrange() is used to iterate in for loops. |

1. **Difference between division and floor division**

The division operator is performed on two operands. This means that we get an exact division result.

For example: print(9/4.0) produces 2.25.

The floor division performs an integer division. This means that we get an integer as a result.

For example: print(9//4.0) produces 2. The fractional part is truncated.

1. **Can we use list as a key in dict?why?**

**A dictionary or a list cannot be a key.**

Lists are [mutable objects](https://www.geeksforgeeks.org/mutable-vs-immutable-objects-in-python/) which means that we can change values inside a list append or delete values of the list . So if a hash function is generated from a list and then the items of the lists changed the dictionary will generate a new hash value for this list and could not find it.

a dictionary key must be of a type that is immutable. For example, you can use an integer, float, string, or [Boolean](https://realpython.com/python-boolean/) as a dictionary key. However, neither a list nor another dictionary can serve as a dictionary key, because lists and dictionaries are mutable. Values, on the other hand, can be any type and can be used more than once.

1. **How to convert csvfile to xls**
2. **What is environment in python**

A virtual environment is a tool that helps to keep dependencies required by different projects separate by creating isolated python virtual environments for them.

Why do we need a virtual environment?

Imagine a scenario where you are working on two web-based python projects and one of them uses a Django 1.9 and the other uses Django 1.10 and so on. In such situations virtual environment can be really useful to maintain dependencies of both the projects.

Virtual Environment should be used whenever you work on any Python based project. It is generally good to have one new virtual environment for every Python based project you work on. So the dependencies of every project are isolated from the system and each other.

1. **web scrapping**

Web scraping is an automatic method to obtain large amounts of data from websites. Most of this data is unstructured data in an HTML format which is then converted into structured data in a spreadsheet or a database so that it can be used in various applications. There are many different ways to perform web scraping to obtain data from websites. These include using online services, particular API’s or even creating your code for web scraping from scratch. Many large websites, like Google, Twitter, Facebook, StackOverflow, etc. have API’s that allow you to access their data in a structured format. This is the best option, but there are other sites that don’t allow users to access large amounts of data in a structured form or they are simply not that technologically advanced. In that situation, it’s best to use Web Scraping to scrape the website for data.

Web scraping requires two parts, namely the **crawler** and the **scraper**. The crawler is an artificial intelligence algorithm that browses the web to search for the particular data required by following the links across the internet. The scraper, on the other hand, is a specific tool created to extract data from the website. The design of the scraper can vary greatly according to the complexity and scope of the project so that it can quickly and accurately extract the data.

### How  Web Scrapers Work?

Web Scrapers can extract all the data on particular sites or the specific data that a user wants. Ideally, it’s best if you specify the data you want so that the web scraper only extracts that data quickly. For example, you might want to scrape an Amazon page for the types of juicers available, but you might only want the data about the models of different juicers and not the customer reviews.

So, when a web scraper needs to scrape a site, first the URLs are provided. Then it loads all the HTML code for those sites and a more advanced scraper might even extract all the CSS and Javascript elements as well. Then the scraper obtains the required data from this HTML code and outputs this data in the format specified by the user. Mostly, this is in the form of an Excel spreadsheet or a CSV file, but the data can also be saved in other formats, such as a JSON file.

Web Scrapers can be divided on the basis of many different criteria, including Self-built or Pre-built Web Scrapers, Browser extension or Software Web Scrapers, and Cloud or Local Web Scrapers.

[**Scrapy**](https://scrapy.org/) is a very popular open-source web crawling framework that is written in Python. It is ideal for web scraping as well as extracting data using APIs. [**Beautiful soup**](https://pypi.org/project/beautifulsoup4/) is another Python library that is highly suitable for Web Scraping.

Web Scraping has multiple applications across various industries. Let’s check out some of these now!

#### 1. Price Monitoring

Web Scraping can be used by companies to scrap the product data for their products and competing products as well to see how it impacts their pricing strategies. Companies can use this data to fix the optimal pricing for their products so that they can obtain maximum revenue.

#### 2. Market Research

Web scraping can be used for market research by companies. High-quality web scraped data obtained in large volumes can be very helpful for companies in analyzing consumer trends and understanding which direction the company should move in the future.

#### 3. News Monitoring

Web scraping news sites can provide detailed reports on the current news to a company. This is even more essential for companies that are frequently in the news or that depend on daily news for their day-to-day functioning. After all, news reports can make or break a company in a single day!

#### 4. Sentiment Analysis

If companies want to understand the general sentiment for their products among their consumers, then Sentiment Analysis is a must. Companies can use web scraping to collect data from social media websites such as Facebook and Twitter as to what the general sentiment about their products is. This will help them in creating products that people desire and moving ahead of their competition.

#### 5. Email Marketing

Companies can also use Web scraping for email marketing. They can collect Email ID’s from various sites using web scraping and then send bulk promotional and marketing Emails to all the people owning these Email ID’s