Python 101: From N00b to N3rd Glossary

!= (Not Equals Operator): The exclamation mark followed by an equals sign is used to check if two values or variables are not equal. It returns True if the values on both sides are not equal, and False if they are equal. For example, x != 5 checks if the value of x is not equal to 5.

" (Double Quotes): In Python, double quotes are used to represent string literals. Strings are a sequence of characters enclosed in double quotes (e.g., "Hello, World!"). They are commonly used for storing and manipulating text data in Python.

= (Equals Operator): The equals operator is used for assignment in Python. It assigns a value on the right side of the operator to the variable on the left side. For example, x = 10 assigns the value 10 to the variable x.

= (Greater Than or Equal To Operator): The greater than or equal to operator compares if the value on the left side is greater than or equal to the value on the right side. It returns True if the condition is true, and False otherwise. For example, x >= 5 checks if the value of x is greater than or equal to 5.

= Operator: The "=" operator in Python is used for assignment. It assigns a value to a variable. For example, x = 10 assigns the value 10 to the variable x. It should not be confused with the "==" operator, which is used for equality comparison.

== (Equality Operator): The double equals operator is used to compare two values or variables for equality. It returns True if the values on both sides are equal, and False otherwise. For example, x == 5 checks if the value of x is equal to 5.

> (Greater Than Operator): The greater than operator is used to compare if the value on the left side is greater than the value on the right side. It returns True if the condition is true, and False otherwise. For example, x > 5 checks if the value of x is greater than 5.

&: A logical operator in Python used for performing logical AND operations. It returns True if both operands are True, and False otherwise.

( ): Denotes a tuple in Python, which is an ordered collection of elements enclosed in parentheses. Tuples are similar to lists, but they are immutable, meaning their elements cannot be changed after creation.

.iat: An indexing attribute in pandas used for accessing a specific value in a dataframe or series by its row and column position. It provides a more efficient way to retrieve single values compared to other indexing methods.

.index: An attribute in pandas used to access the index labels of a dataframe or series. It returns the index object associated with the data.

[ ]: Denotes a list in Python, which is an ordered collection of elements enclosed in square brackets. lists can contain multiple values or elements.

[function]?: The use of a question mark after a function name in Python is a convention commonly used to access the documentation or help for that function. It allows you to retrieve information about the function, including its parameters, usage, and examples.

{ }: Denotes a dictionary in Python, which is an unordered collection of key-value pairs enclosed in curly braces. Dictionaries allow for efficient lookup and retrieval of values based on their associated keys.

< (Less Than Operator): The less than operator is used to compare if the value on the left side is less than the value on the right side. It returns True if the condition is true, and False otherwise. For example, x < 10 checks if the value of x is less than 10.

<= (Less Than or Equal To Operator): The less than or equal to operator compares if the value on the left side is less than or equal to the value on the right side. It returns True if the condition is true, and False otherwise. For example, x <= 10 checks if the value of x is less than or equal to 10.

<name of environment>/scripts/activate: This command is used to activate a virtual environment in Windows. It activates the specified virtual environment named "<name of environment>". Similar to the Unix-based systems, activating a virtual environment modifies the environment variables in Windows to use the Python interpreter and packages within that environment.

acct_df.iloc[:,1:2]: An iloc indexing operation that selects a range of columns from the second to the third column (inclusive) of a dataframe and returns all rows within that range.

aggregate: A method in pandas used to compute multiple aggregate functions on one or more columns of a dataframe. It applies the specified functions to the grouped data and returns the results.

and/or: "and" and "or" are logical operators in Python used to combine multiple conditions.

and: operator returns true if both conditions on its left and right sides are true.

append() function: Adds an element to the end of a list.

argument: A value that is passed to a function when the function is called. It provides the necessary inputs or data for the function to perform its task.

as: A keyword used in Python to create an alias or alternative name for a module, class, or function. It allows you to refer to the alias instead of the full name, providing a shorter or more convenient way to access the entity.

assignment statement: Assigns a value to a variable using the "=" operator.

bang (!): In Python, the bang (!) character is not a specific operator or syntax. It is commonly used in other programming languages or command-line interfaces to represent the "not" or "negation" operation. However, in Python, the "not" operator is used for negation.

bins: A parameter used in data analysis and visualization to specify the number or range of intervals into which the data will be divided or grouped.

booleans: Booleans are a data type in Python that represent truth values. They can have one of two possible values: True or False. Booleans are often used in conditional statements or logical operations to make decisions or evaluate expressions.

break statement: The "break" statement is used in loops to immediately exit the loop and resume execution at the next statement after the loop. It is often used to terminate a loop prematurely based on a certain condition.

case: A keyword or concept referring to different conditional cases or scenarios in programming, where different actions or operations may be performed based on specific conditions.

cast: Casting, in the context of Python, refers to the process of converting one data type into another. It allows you to change the data type of a variable to match the requirements of a specific operation or to store the value in a different format. Python provides built-in functions like int(), float(), str(), and bool() for casting variables to different types.

code block: A code block in Python is a section of code that is grouped together and executed as a unit. It is typically marked by the indentation level. Code blocks are used to define the body of control structures like loops or conditional statements, as well as function definitions and class definitions.

col_index: Refers to the index or label of a specific column in a dataframe. It is used to locate and access data within a specific column.

colon: The colon (:) is a punctuation mark used in Python to indicate the start of a new code block. It is commonly used in control structures, such as if statements, for loops, and function definitions. The colon is followed by an indented block of code that is executed when the condition or block is satisfied. The colon serves as a visual indicator of the beginning of a code block and helps maintain the code's structure and readability.

continue statement: The "continue" statement is used in loops to skip the current iteration and move to the next iteration. It is typically used to bypass specific code execution for certain conditions within a loop.

cut: A function in pandas used for binning or categorizing numerical data into discrete intervals or bins. It assigns labels or categories to the data based on specified bin edges.

data analytics: Data analytics refers to the process of examining, transforming, and interpreting data to uncover meaningful insights and make informed decisions. In Python, various libraries and tools, such as pandas, NumPy, and scikit-learn, are commonly used for data analytics tasks.

data collection: Data collection involves gathering or acquiring data from different sources, such as databases, files, or APIs, for further analysis. In Python, you can use various techniques and libraries, such as web scraping, database connections, or data ingestion functions, to collect data and store it in a suitable format for analysis.

data Interpretation: Data interpretation involves drawing meaningful insights and conclusions from the analyzed data. It requires understanding the context, patterns, and relationships discovered during data

analysis. Python provides a rich ecosystem of data analysis libraries and tools that facilitate data interpretation, allowing you to derive actionable insights and make informed decisions.

data preparation: Data preparation involves cleaning, transforming, and organizing data to ensure its quality and suitability for analysis. This step often includes tasks such as removing duplicates, handling missing values, normalizing data, or feature engineering. Python provides powerful libraries like pandas for data manipulation and preprocessing tasks.

data processing: Data processing refers to the manipulation, transformation, and analysis of data to extract meaningful insights or derive useful information. In Python, various libraries and techniques are available for data processing tasks, such as cleaning, filtering, aggregating, and transforming data.

data structures: Containers that allow you to organize and store data efficiently.

dataframes: A two-dimensional tabular data structure provided by the pandas library. It organizes data into rows and columns, similar to a spreadsheet or SQL table, and offers powerful data manipulation and analysis capabilities.

deactivate: This command is used to deactivate the currently active virtual environment. It restores the system's default Python environment, disabling the isolation provided by the virtual environment. Once deactivated, the system will use the globally installed Python interpreter and packages.

def: A keyword in Python used to define a function. It is followed by the function name, a set of parentheses, and a colon. The code block that follows defines the body of the function.

del: The del keyword in Python is used to delete variables, objects, or elements from a data structure. In the context of dictionaries, it can be used to remove a specific key-value pair from the dictionary.

dict(): A built-in function in Python that creates a new dictionary. It can be used to convert other iterable objects, such as tuples or lists of key-value pairs, into a dictionary. The keys and values in the iterable are mapped to corresponding key-value pairs in the dictionary.

dictionaries of lists: A data structure where a dictionary contains lists as its values. Each key in the dictionary maps to a list of elements.

dictionary comprehension: Similar to list comprehension, dictionary comprehension is a concise way to create dictionaries in Python. It allows you to generate a new dictionary by iterating over an iterable and specifying key-value pairs using an expression within curly braces.

dictionary: A data structure in Python that stores data as key-value pairs. Each key in the dictionary is unique and associated with a value.

dtypes: An attribute in pandas used to obtain the data types of the columns in a dataframe. It returns a series containing the data type of each column.

elif: The "elif" statement is short for "else if" and is used in conjunction with the "if" statement. It allows you to check additional conditions after the initial "if" condition. If the preceding "if" condition is false, the "elif" condition is evaluated. If the "elif" condition is true, the corresponding block of code is executed.

else: The "else" statement is used in conjunction with the "if" statement. It provides an alternative block of code to be executed when the preceding "if" and "elif" conditions are false. The "else" block is optional, and it executes if none of the preceding conditions are true.

endswith: A method in Python used to check if a string ends with a specific substring. It returns a boolean value indicating whether the string ends with the specified substring.

existence checks: Checks performed to determine if a key or value exists in a dictionary. These checks can be useful to avoid errors when accessing or manipulating dictionary elements.

expression: A combination of values, variables, operators, and function calls that evaluates to a single value. Expressions can be used in various contexts, such as assignments, function arguments, or within other expressions.

False: False is a boolean value in Python that represents the concept of falsehood. It is used to indicate a condition or statement that is false or not valid.

fillna('value specified'): A method in pandas used to replace missing or NaN values in a dataframe or series with a specified value.

filtering: The process of extracting specific subsets of data from a larger dataset based on specified conditions or criteria. It allows you to select and retain only the rows or columns that meet the given conditions.

float: Float is a numeric data type in Python that represents decimal numbers. Floats can be used to store both whole numbers and numbers with decimal points. For example, 3.14 is a float value in Python.

for loop: A "for" loop is used to iterate over a sequence (such as a list or a string) or other iterable objects. It executes a block of code for each item in the sequence.

f-string: A string formatting technique in Python that allows you to embed expressions inside string literals. It is denoted by placing an 'f' before the opening quotation mark and using curly braces {} to enclose the expressions.

function: A reusable block of code that performs a specific task. Functions in Python are defined using the "def" keyword, followed by the function name, parentheses, and an optional list of parameters. They can be called or invoked to execute the code inside the function.

get_loc: A method in pandas used to retrieve the integer position of a specific label within an index. It returns the index position corresponding to the label.

groupby: A method in pandas used for grouping data based on one or more columns. It allows you to group the data and perform aggregate operations on each group.

grouping: The process of dividing data into groups based on one or more variables or columns. It enables aggregation and analysis of data within each group separately.

head: A function in the pandas library that returns the first few rows of a dataframe. It is useful for quickly inspecting the structure and contents of the dataframe.

if: The "if" statement is a control structure in Python that allows you to execute a block of code conditionally. It evaluates a given condition and, if the condition is true, executes the code within the if block.

ignore_index=True: A parameter used when concatenating or appending dataframes in pandas. It specifies whether to ignore the original row index labels and create a new index from 0 onwards.

iloc: A method in pandas used for integer-based indexing and slicing of dataframes. It allows you to access and manipulate data by specifying integer positions of rows and columns.

iloc[:,1]: An iloc indexing operation that selects the second column of a dataframe and returns all rows of that column.

iloc[1,:]: An iloc indexing operation that selects the second row of a dataframe and returns all columns of that row.

in operator: The "in" operator is used in Python to check if a value or element exists in a sequence or collection. It returns True if the value is found and False otherwise.

Indentation: Indentation must be consistent across your program: In Python, it is essential to maintain consistent indentation throughout your program. This means that all lines within the same code block should have the same indentation level. Inconsistencies in indentation can lead to syntax errors and code misinterpretation by the Python interpreter.

index: Numeric representation that uniquely identifies an element in a data structure.

insert() function: Inserts an element at a specific position within a list.

int: Int, short for integer, is a numeric data type in Python that represents whole numbers. Integers can be positive or negative and do not have decimal points. For example, 5 and -10 are integer values in Python.

Integrated Development Environment (IDE): An Integrated Development Environment, or IDE, is a software application that provides a comprehensive set of tools for writing, testing, and debugging code. It typically includes a code editor, a debugger, a compiler or interpreter, and various other features to facilitate the development process. Python has several popular IDEs, such as PyCharm, Visual Studio Code (with Python extensions), and IDLE (included with Python installation).

Internet of Things (IoT): The Internet of Things (IoT) refers to a network of interconnected physical devices (things) that can communicate, collect, and exchange data with each other through the internet. In the context of Python, IoT applications often involve using Python to interface with sensors, control actuators, process data, and build applications that interact with IoT devices or systems. Python libraries like Adafruit CircuitPython and MQTT can be used for IoT development with Python.

items function: The items() function is used with dictionaries in Python to return a view object that contains key-value pairs as tuples. This view object can be used to iterate over the key-value pairs or perform operations on them.

iterative variable: The iterative variable refers to the variable used to control the iteration or loop. It is often used to store each value from the sequence during each iteration of the loop.

k: A variable or parameter often used to represent the number of clusters in clustering algorithms or the number of folds in cross-validation techniques.

key: In a dictionary, the key is the identifier that describes the associated value. It is used to access and retrieve the corresponding value.

keys function: The keys() function is used with dictionaries in Python to return a view object that contains all the keys of the dictionary. This view object can be used to iterate over the keys or perform operations on them.

key-value pair: A pair of elements in a dictionary, where the key describes the data (similar to a column header in a database), and the value represents the actual value of the data.

lambda function: An anonymous, single-line function defined using the "lambda" keyword. Lambda functions are used for simple and concise operations where a full function definition is not necessary.

len() function: Returns the number of elements in a data structure or the length of a sequence.

list comprehension: list comprehension is a concise and elegant way to create lists in Python. It allows you to create a new list by iterating over an existing sequence or iterable and applying an expression or condition to each element.

lists of dictionaries: A data structure where a list contains multiple dictionaries as its elements. Each dictionary within the list can have different keys and values.

lists: Mutable, ordered collections of elements enclosed in square brackets ([]).

loc: A method in pandas used for label-based indexing and slicing of dataframes. It allows you to access and manipulate data by specifying row and column labels.

logical operators: Logical operators in Python are used to combine or manipulate conditions. The three logical operators are "and," "or," and "not".

loops in Python: "for" loops and "while" loops.

mutability: Refers to the ability to modify an object after its creation.

nested if statements: In Python, nested if statements refer to placing one if statement inside another if, elif, or else block. This allows you to create multiple levels of conditions based on various scenarios. The inner if statements are executed only if the outer condition(s) are true.

not: "not" is a logical operator in Python used to negate a condition. It returns the opposite of the truth value of a condition. For example, not x == 5 checks if the value of x is not equal to 5.

or: operator returns true if at least one of the conditions on its left or right side is true.

pandas: A popular open-source library in Python for data manipulation, analysis, and visualization. It provides data structures and functions to efficiently handle structured data, such as series (one-dimensional labeled arrays) and dataframes (two-dimensional tabular data).

pip freeze: This command lists all the installed packages and their versions in the current Python environment. It provides a snapshot of the installed packages along with their version numbers. This is useful for documenting the dependencies of a project or sharing the requirements with others.

pip install pandas: This command installs the "pandas" package using pip. "pandas" is a popular Python library for data manipulation, analysis, and cleaning. It provides powerful data structures and data analysis tools, making it useful for tasks such as working with structured data, time series analysis, and data preprocessing.

pip install --upgrade python: This command is used in the command line interface (CLI) to upgrade the Python interpreter itself. "pip" is a package management system in Python that allows you to install, upgrade, and manage Python packages and libraries. The "install" command installs a package, and the "--upgrade" flag ensures that the installed package is upgraded to the latest version.

pip install virtualenv: This command installs the "virtualenv" package using pip. "virtualenv" is a Python package that allows you to create and manage virtual environments. It provides a convenient way to isolate project-specific dependencies and keep them separate from the system-wide Python installation.

pprint: The pprint module in Python provides the pprint function, which is used to pretty print data structures, including dictionaries, by formatting them in a visually appealing manner.

pretty print: A feature in Python that formats and prints complex data structures, such as dictionaries, in a more readable and organized way. It can be enabled by importing the pprint module and using the pprint function.

range function (takes up to three arguments: start, stop, step): The range() function in Python is used to generate a sequence of numbers. It can take up to three arguments: start, stop, and step.

read_csv: A function in the pandas library that reads data from a CSV (Comma-Separated Values) file and returns it as a dataframe. It is commonly used to import structured data from files into Python.

remove() function: Removes the first occurrence of a specified element from a list.

return: A keyword used in a function to specify the value(s) that the function should send back as a result. It allows the function to output a value or set of values.

reverse parameter: An optional argument for the sort() function that sorts the list in descending order when set to True.

row_index: Refers to the index or label of a specific row in a dataframe. It is used to locate and access data within a specific row.

scraping: Scraping refers to the process of extracting data from websites or web pages. In Python, you can use libraries like BeautifulSoup or Scrapy to automate the retrieval and extraction of data from HTML or XML documents.

series: A one-dimensional labeled array-like data structure provided by the pandas library. It can hold data of any type and supports various operations and functions for data manipulation.

Shift + Enter: Shortcut for running a code block.

size: A method or attribute in pandas used to determine the size or total number of elements in a dataframe, series, or array.

sort() function: Sorts the elements of a list in ascending order.

sorting: Arranging data in a particular order, typically based on one or more columns or variables. Sorting can be performed in ascending or descending order to better analyze and understand the data.

source <name of environment>/bin/activate: This command is used to activate a virtual environment in Unix-based systems (e.g., Linux or macOS). It activates the specified virtual environment named "<name of environment>". Activating a virtual environment modifies the environment variables so that the Python interpreter and installed packages within that environment are used when executing commands.

start: The start argument in the range() function represents the starting value of the sequence generated by the range function.

startswith: A method in Python used to check if a string starts with a specific substring. It returns a boolean value indicating whether the string starts with the specified substring.

statistical modeling: Statistical modeling refers to the process of building mathematical models that describe relationships between variables in a dataset. These models help analyze and interpret data, make predictions, or uncover patterns. Python offers libraries like statsmodels and scikit-learn that provide a wide range of statistical modeling techniques, such as linear regression, logistic regression, or time series analysis.

statistical summaries: Numerical summaries or descriptive statistics that provide insights into the distribution, central tendency, variability, and other characteristics of a dataset. Examples include mean, standard deviation, minimum, maximum, and quartiles.

step: The step argument in the range() function represents the increment between consecutive values in the sequence generated by the range function.

stop: The stop argument in the range() function represents the end value (exclusive) of the sequence generated by the range function.

str.contains: A method in pandas used to check if a string contains a specific substring. It returns a boolean series indicating whether each element in the column contains the specified substring.

t function: A method in pandas used to transpose a dataframe or series. It swaps the rows and columns, transforming a row-based structure into a column-based structure or vice versa.

tail: The tail() function in Pandas is used to display the last few rows of a DataFrame. By default, it shows the last five rows, but you can specify the number of rows to display.

transformation clause: In list comprehension, the transformation clause specifies the expression or operation to be applied to each element of the original sequence. It typically follows the iteration clause and is enclosed in square brackets.

True: True is a boolean value in Python that represents the concept of truth. It is used to indicate a condition or statement that is true or valid.

tuples: Immutable, ordered collections of elements enclosed in parentheses (()).

type function: The type() function in Python is used to determine the data type of a variable or value. It returns the type of the object as a result. For example, if you have a variable x and want to know its type, you can use type(x) to get the data type of x, such as int, float, str, or bool. The type() function is useful for debugging, type checking, and understanding the data you're working with.

upper() function: A built-in function in Python that converts a string to uppercase letters. It returns a new string with all the characters converted to uppercase, while leaving non-alphabetic characters unchanged.

value: In a dictionary, the value is the actual data or information associated with a specific key. It represents the content or value of the data.

values function: The values() function is used with dictionaries in Python to return a view object that contains all the values of the dictionary. This view object can be used to iterate over the values or perform operations on them.

variables: Variables in Python are used to store and represent data. They are containers that hold a value, which can be of different types, such as numbers, text, or booleans. Variables allow you to store and manipulate data dynamically during program execution.

virtual environments: Virtual environments are isolated Python environments that allow you to install and manage packages separately for different projects. They enable you to have different versions of packages and avoid conflicts between project dependencies. Virtual environments help maintain project-specific dependencies and configurations. They are especially useful when working on multiple projects with different requirements or when collaborating with others.

virtualenv mars: This command creates a new virtual environment named "mars". It uses the "virtualenv" package to set up an isolated Python environment specifically for the project or task at hand. "mars" is the chosen name for the virtual environment, but you can use any name you prefer.

visualization: Visualization involves creating graphical representations of data to visually explore and communicate patterns, trends, and relationships. Python has popular libraries like Matplotlib, Seaborn, and Plotly, which provide a wide range of functions and tools to create static or interactive visualizations, including bar charts, scatter plots, heatmaps, or interactive dashboards.

while loop: A "while" loop is used to repeatedly execute a block of code as long as a given condition is true. It continues execution until the condition becomes false.

zip(): A built-in function in Python that takes multiple iterables as arguments and returns an iterator that generates tuples containing elements from each iterable. It pairs corresponding elements together, creating an iterator of tuples that can be used, for example, to iterate over multiple lists simultaneously.