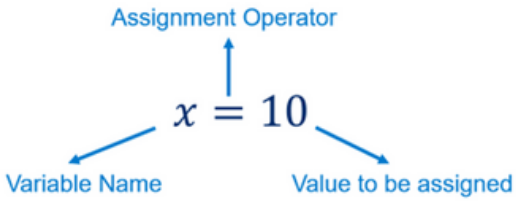
CFA Institute

PYTHON PROGRAMMING FUNDAMENTALS: CHEAT SHEET

Dr. Ryan Ahmed

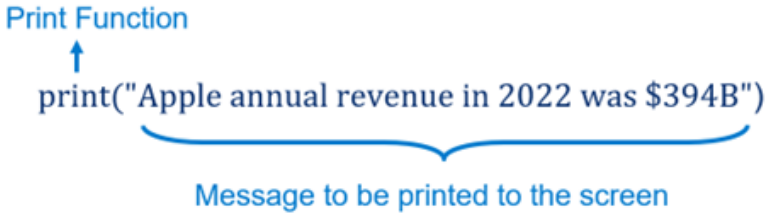
1. *Variable Assignment*

- Variables are used to hold values which could be integers, strings, floating points, or Booleans.
- The assignment operator "=" is used to assign a value to a variable.



2. *Print Operation*

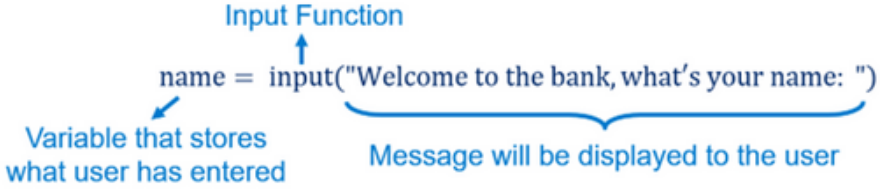
- **The print() function is used to print a message to the screen. The message is in a "String" data type.**



- The format() method prints a generalized statement to the screen by formatting a specified value and inserting it in the placeholder indicated by curly braces "{ }".

3. *Get User Input*

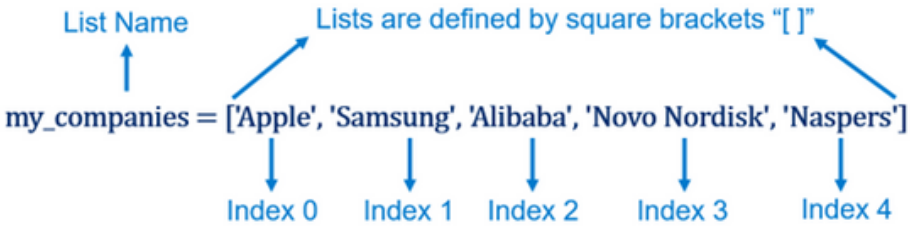
- User inputs are obtained using the input() built-in function.
- Users can enter data using the keyboard, data will be stored in the "name" variable in a "String" datatype.



4. *Python Lists*

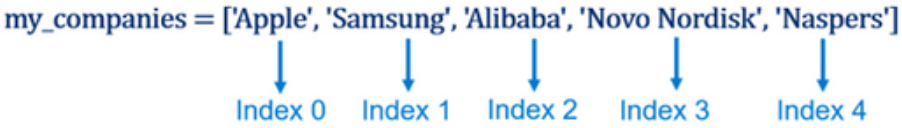
a. Lists Definition

- In Python, a list is a collection of items that are ordered and have a specific index.
- Lists are defined using square brackets "[ ]" and elements are separated by commas.



b. Lists Indexing

- Python list elements are accessed using indexing in which the first element in the list has an index of 0.




```
my_companies[0] ⟶ 'Apple'
my_companies[1] ⟶ 'Samsung'
my_companies[4] ⟶ 'Naspers'
```

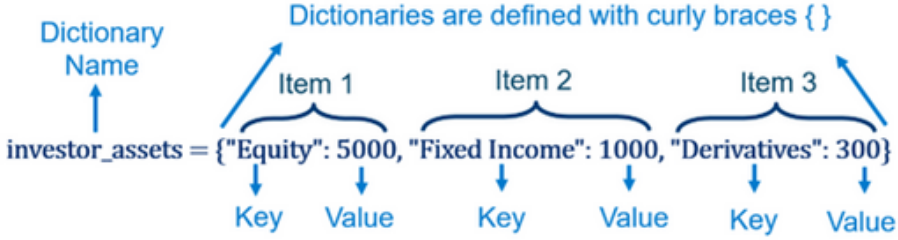
c. Lists Slicing

- List slicing is used to obtain more than one element from a list.
- The slicing operator [n:m] is used to obtain elements starting with index n up to but not including the element in index m.


```
my_companies[1:4] ⟶ ['Samsung', 'Alibaba', 'Novo Nordisk']
```

5. *Python Dictionary*

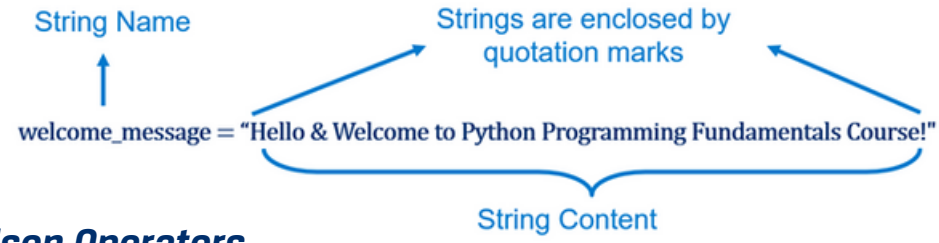
- Python dictionaries consist of a collection of key-value pairs that are unordered.
- Dictionary elements are accessed using their keys. Keys are unique while values are not.




```
investor_assets["Fixed Income"] ⟶ 1000
```

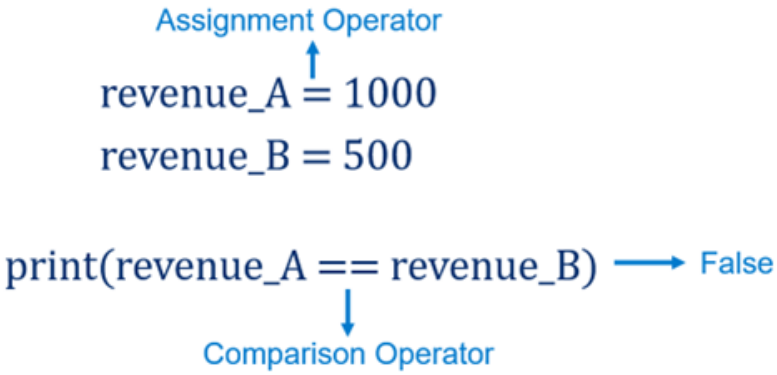
6. Python Strings

- In Python, a string is a sequence of characters that are enclosed by single or double quotation marks.



7. Comparison Operators

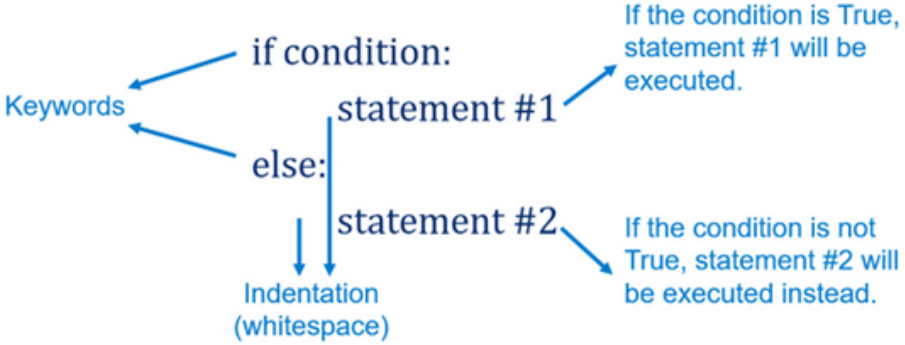
- Comparison Operators compare the values of two operands and return "True" or "False".
- ·Remember: assignment operators "=" assigns a value to a variable while comparison operators "==" compares two values.



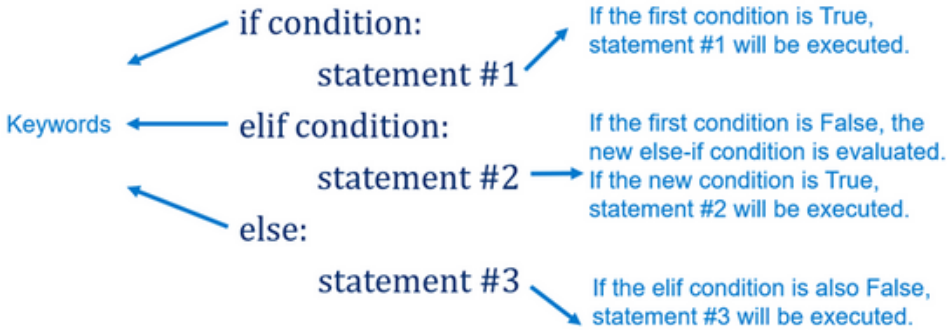
Operator	Description
a == b	Is "a" equal to "b"?
a > b	Is "a" greater than "b"?
a < b	Is "a" less than "b"?
a >= b	Is "a" greater than or equal to "b"?
a <= b	Is "a" less than or equal to "b"?
a != b	Is "a" not equal to "b"?

8. Conditional Statements

- In Python, If-else statements are used for decision-making.

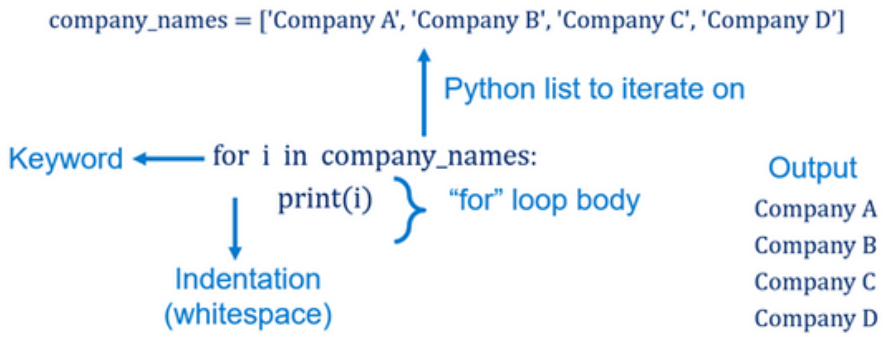


- The "elif" statement adds another decision branch to the If-else condition. This is used to evaluate multiple expressions with more than two branches.



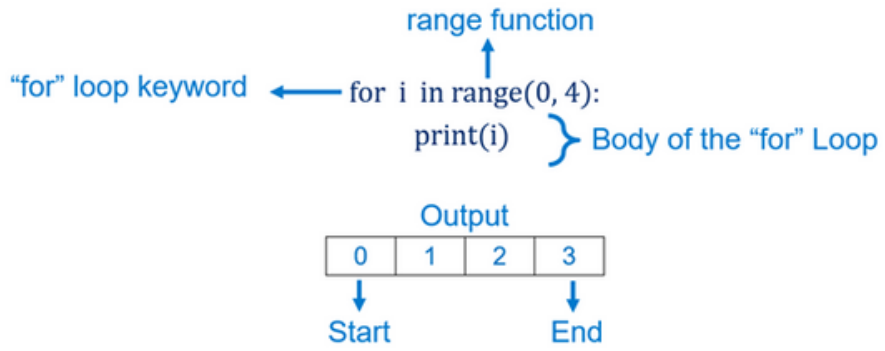
9. *For Loops*

- For loops are used to repeat a block of code for a fixed number of times.
- They are used to iterate over a sequence such as a list or dictionary.



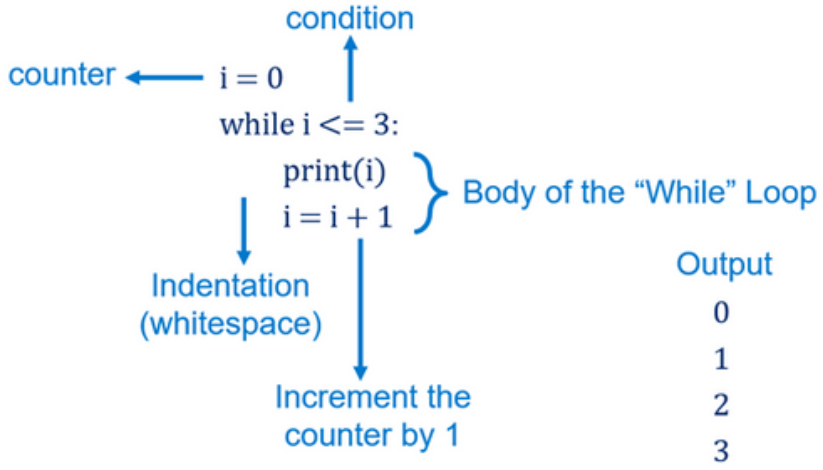
10. *Python Range( ) Function*

- The range() function generates integers starting from 0 up to but not including the last number.
- The range() function generates a list of numbers that are used to iterate over "For" loops.



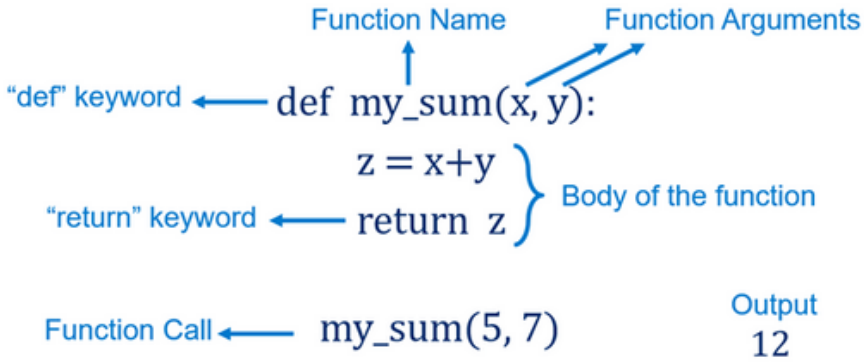
11. "While" Loops

- "While" loops continuously repeat a block of code as long as a given condition is held "True".



12. Functions

- In Python, functions represent a block of code that performs a specific task.
- A function is executed when called, you can send it data and it returns results.



13. Python Built-in Functions

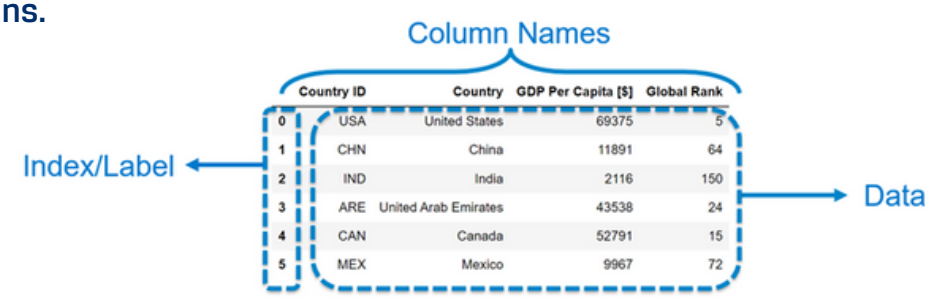
- Python has several built-in functions that can be directly called. There is no need to define these functions beforehand.

abs()	delattr()	hash()	memoryview()	set()
all()	dict()	help()	min()	setattr()
any()	dir()	hex()	next()	slice()
ascii()	divmod()	id()	object()	sorted()
bin()	enumerate()	input()	oct()	staticmethod()
bool()	eval()	int()	open()	str()
breakpoint()	exec()	isinstance()	ord()	sum()
bytearray()	filter()	issubclass()	pow()	super()
bytes()	float()	iter()	print()	tuple()
callable()	format()	len()	property()	type()
chr()	frozenset()	list()	range()	vars()
classmethod()	getattr()	locals()	repr()	zip()
compile()	globals()	map()	reversed()	__import__()
complex()	hasattr()	max()	round()	

14. *Pandas for Data Analysis*

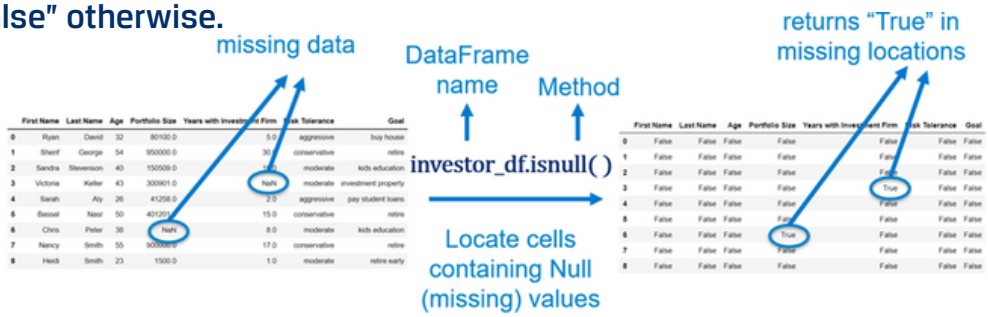
a. Pandas DataFrames

- Pandas is a data manipulation and analysis library in Python that uses a data structure known as DataFrame.
- A DataFrame is a two-dimensional tabular data structure with labeled rows and columns.

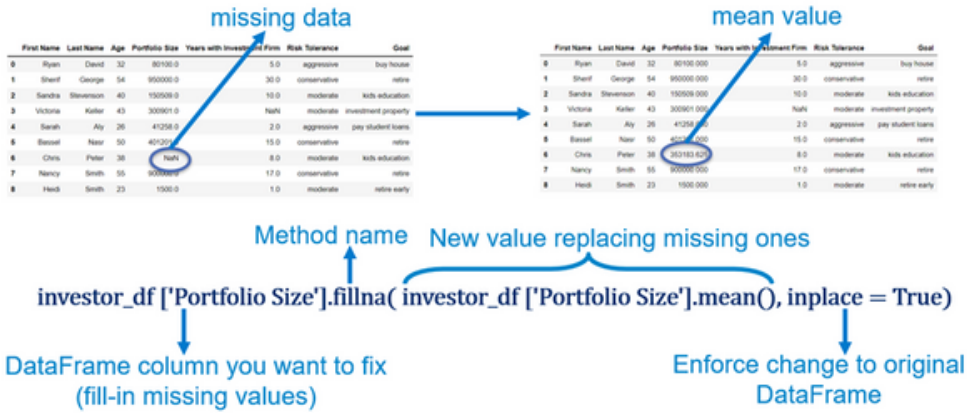


b. Handle Missing Data

- The ".isnull()" method is used to locate missing values in Pandas DataFrames.
- It works by returning another DataFrame containing "True" in missing values location and "False" otherwise.

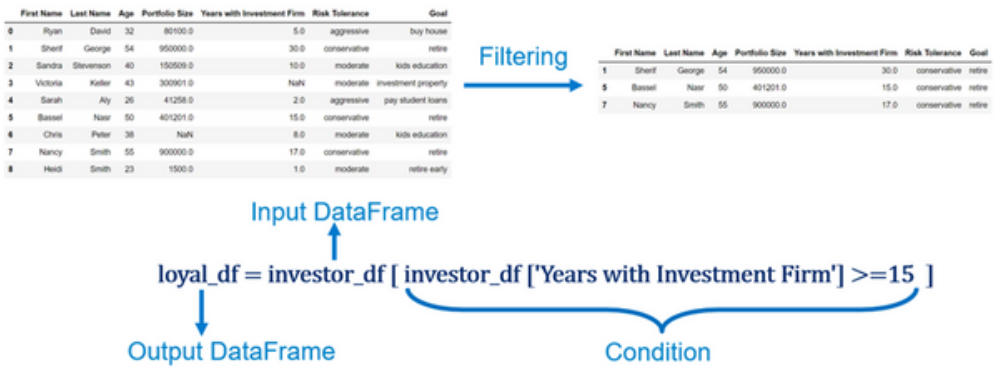


- ".fillna()" method is used to fill missing values with a specific value (mean, median,.. etc.).

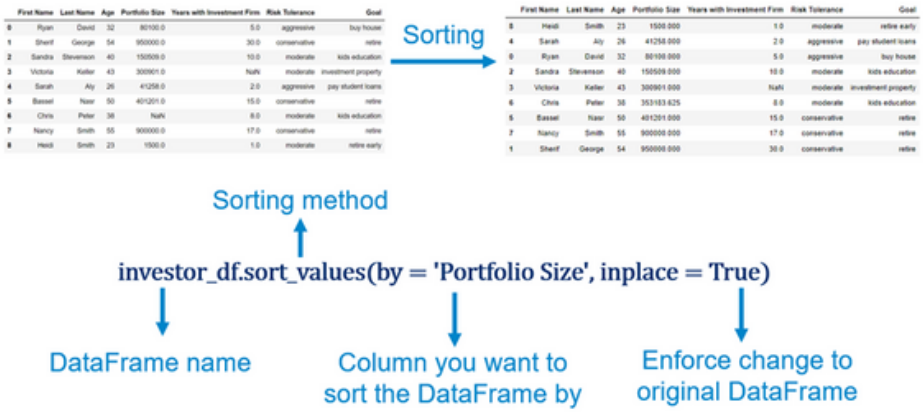


c. DataFrame Sorting and Ordering

- Data filtering is the process of filtering Pandas DataFrames based on a specific condition.

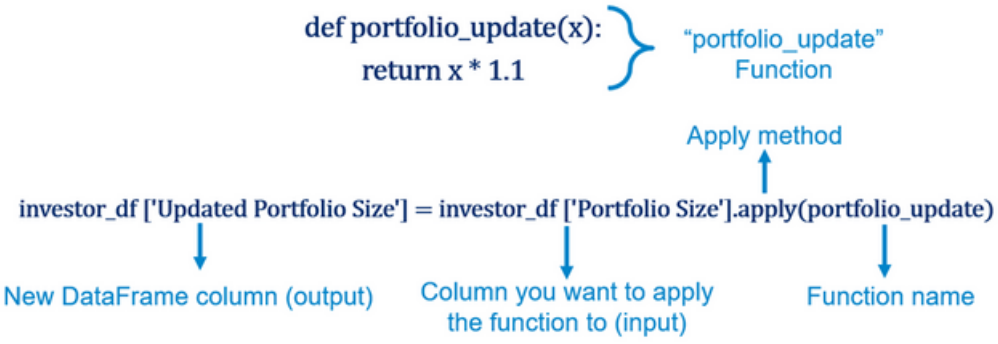


- **Data sorting is the process of organizing data into some meaningful order to make it easier to understand and examine.**



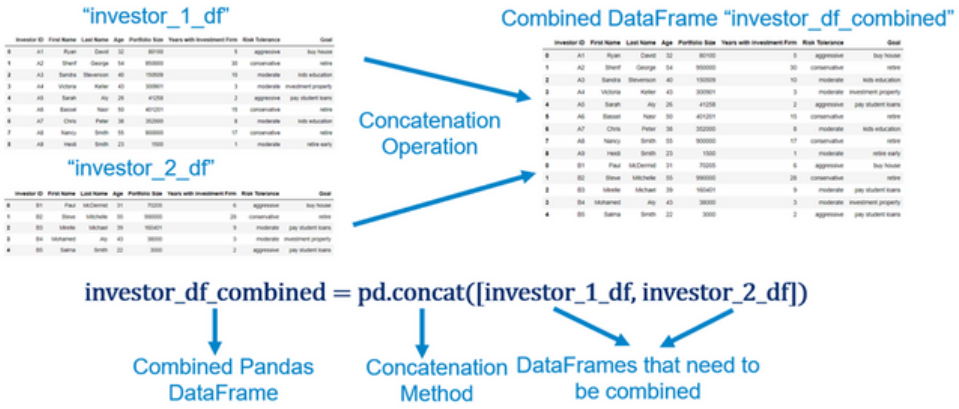
d. DataFrame with Functions

- Pandas DataFrame apply() function is used to apply a function along an axis in the DataFrame.
- Both python built-in functions and custom functions can be used.

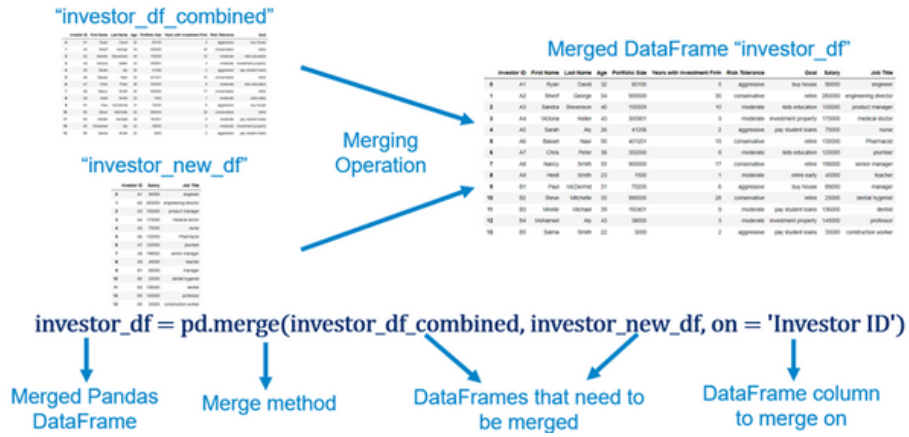


e. DataFrame Concatenation and Merging

- Pandas DataFrame concatenation works by combining two or more DataFrames along an axis to produce a new one.

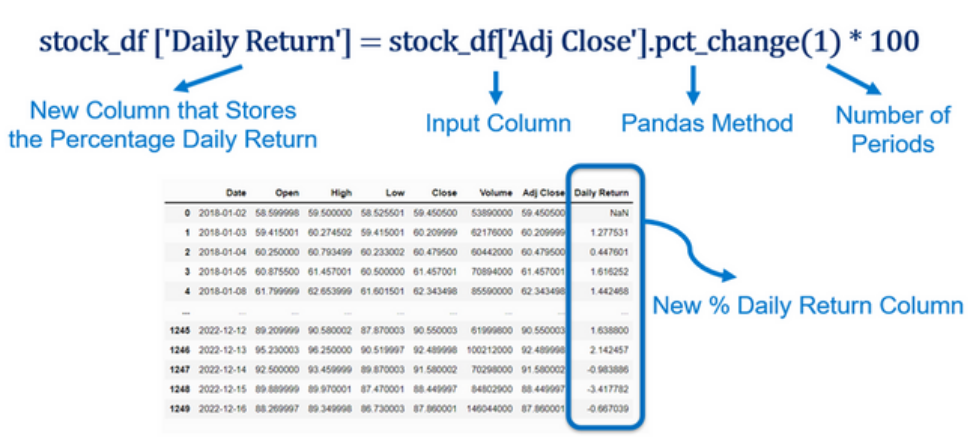


- Pandas merge() function is used to merge two DataFrames on an axis.



15. Calculate Percentage Daily Returns in Python

- Pandas *.pct_change( )* method calculates the percentage change between current and prior elements in a Pandas DataFrame.



	Date	Open	High	Low	Close	Volume	Adj Close	Daily Return
0	2018-01-02	58.599998	59.500000	58.525501	59.450500	53890000	59.450500	NaN
1	2018-01-03	59.415001	60.274502	59.415001	60.209999	62176000	60.209999	1.277531
2	2018-01-04	60.250000	60.793499	60.233002	60.479500	60442000	60.479500	0.447601
3	2018-01-05	60.875500	61.457001	60.500000	61.457001	70894000	61.457001	1.616252
4	2018-01-08	61.799999	62.653999	61.601501	62.343498	85590000	62.343498	1.442468
...	...	...	...	...	...	...	...	...
1245	2022-12-12	89.209999	90.580002	87.870003	90.550003	61999800	90.550003	1.638800
1246	2022-12-13	95.230003	96.250000	90.519997	92.489998	100212000	92.489998	2.142457
1247	2022-12-14	92.500000	93.459999	89.870003	91.580002	70298000	91.580002	-0.983886
1248	2022-12-15	89.889999	89.970001	87.470001	88.449997	84802900	88.449997	-3.417782
1249	2022-12-16	88.269997	89.349998	86.730003	87.860001	146044000	87.860001	-0.667039

16. Generate Line Plots Using Plotly Express

- Plotly Express is a user-friendly high-level interface to the Plotly library.