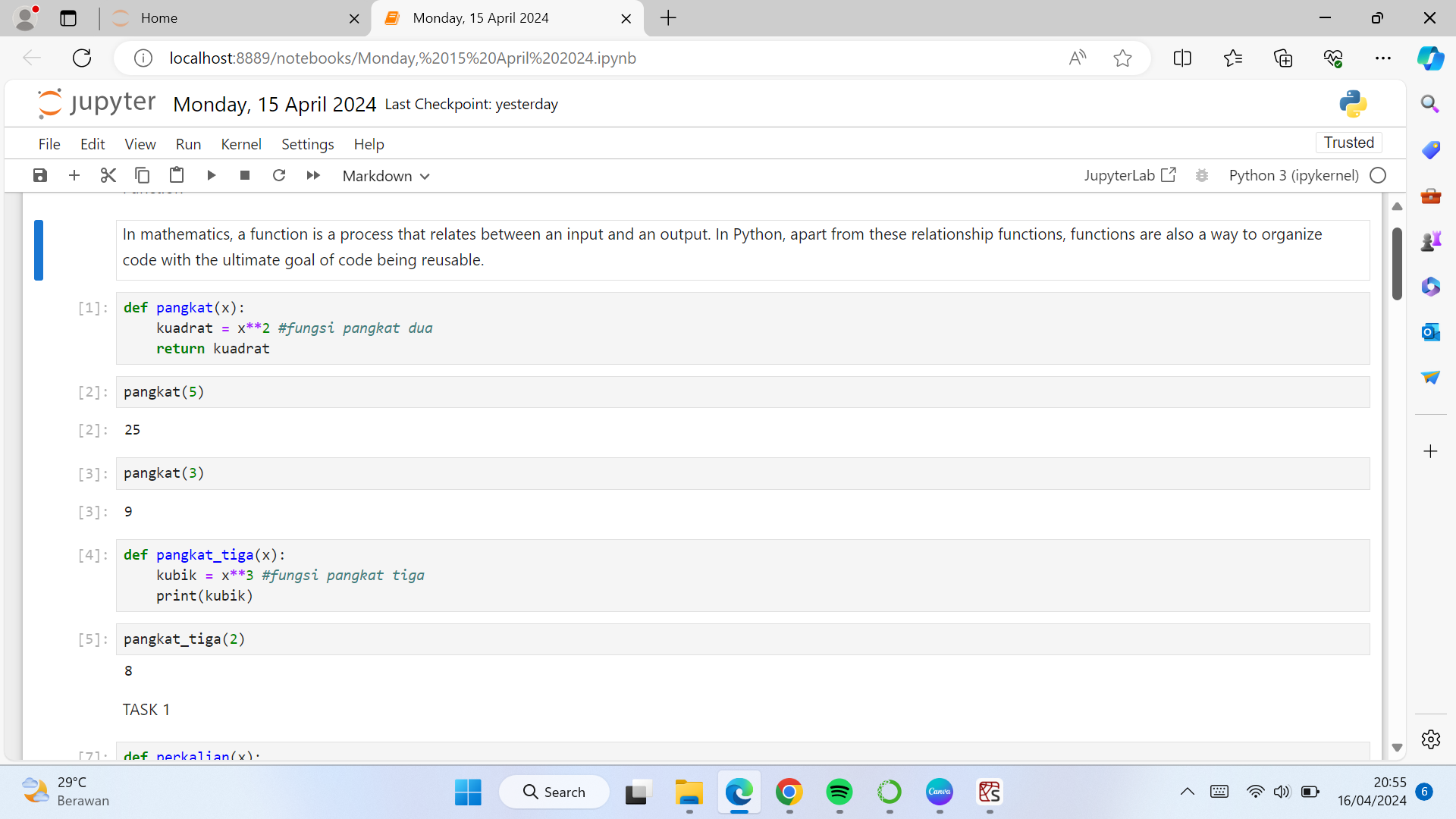
**Senin**

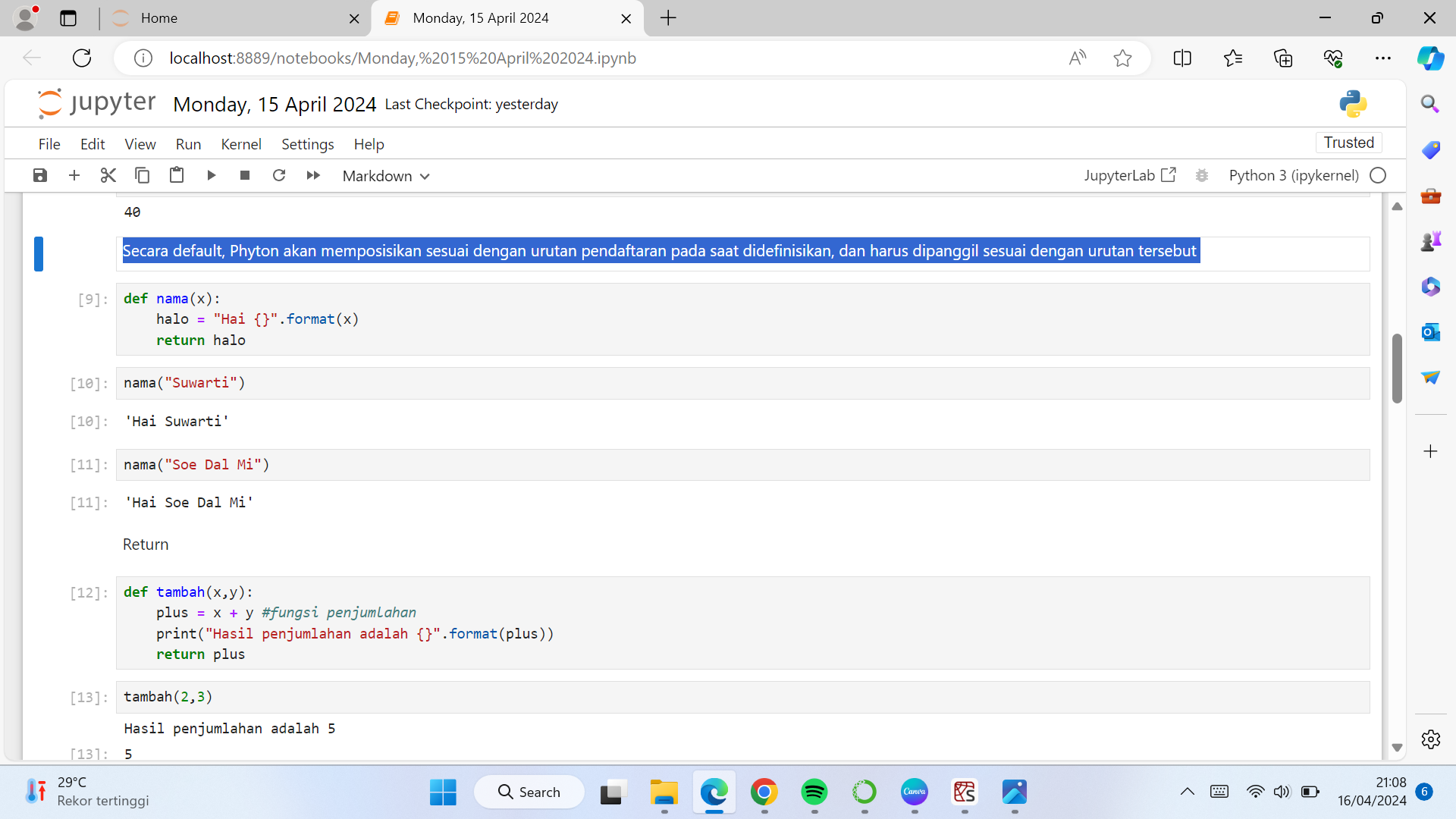
**Function**

**Python Functions** is 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 over and over again.

* In mathematics, a function is a process that relates between an input and an output.
* In Python, apart from these relationship functions, functions are also a way to organize code with the ultimate goal of code being reusable.
* Functions are defined with the def keyword followed by the function name and parameters in brackets ()
* Optionally, a docstring can be added - a documentation string that describes the context of the function

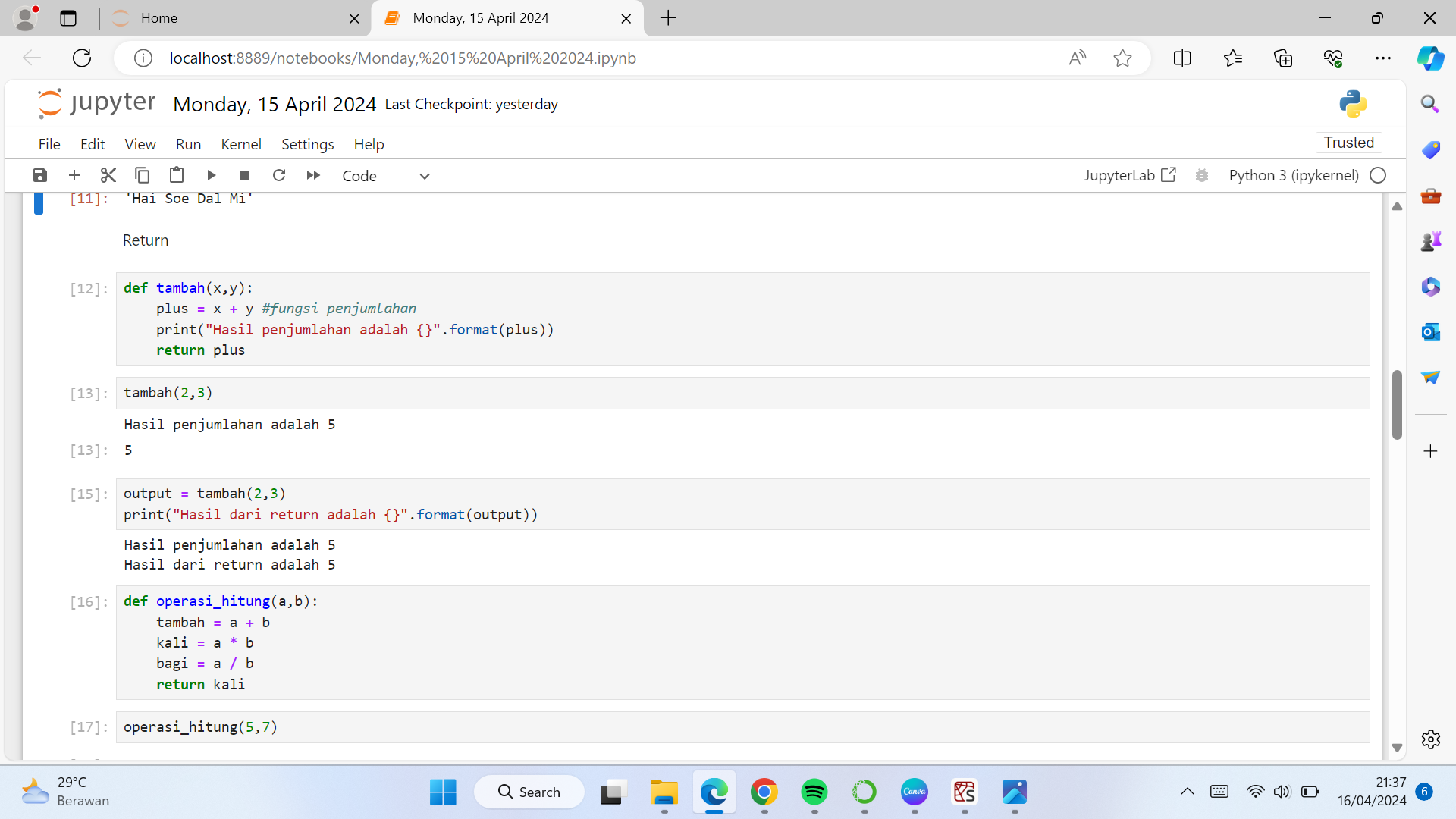


* By default, Python will position according to the registration order in which it was defined, and must be called in that order.

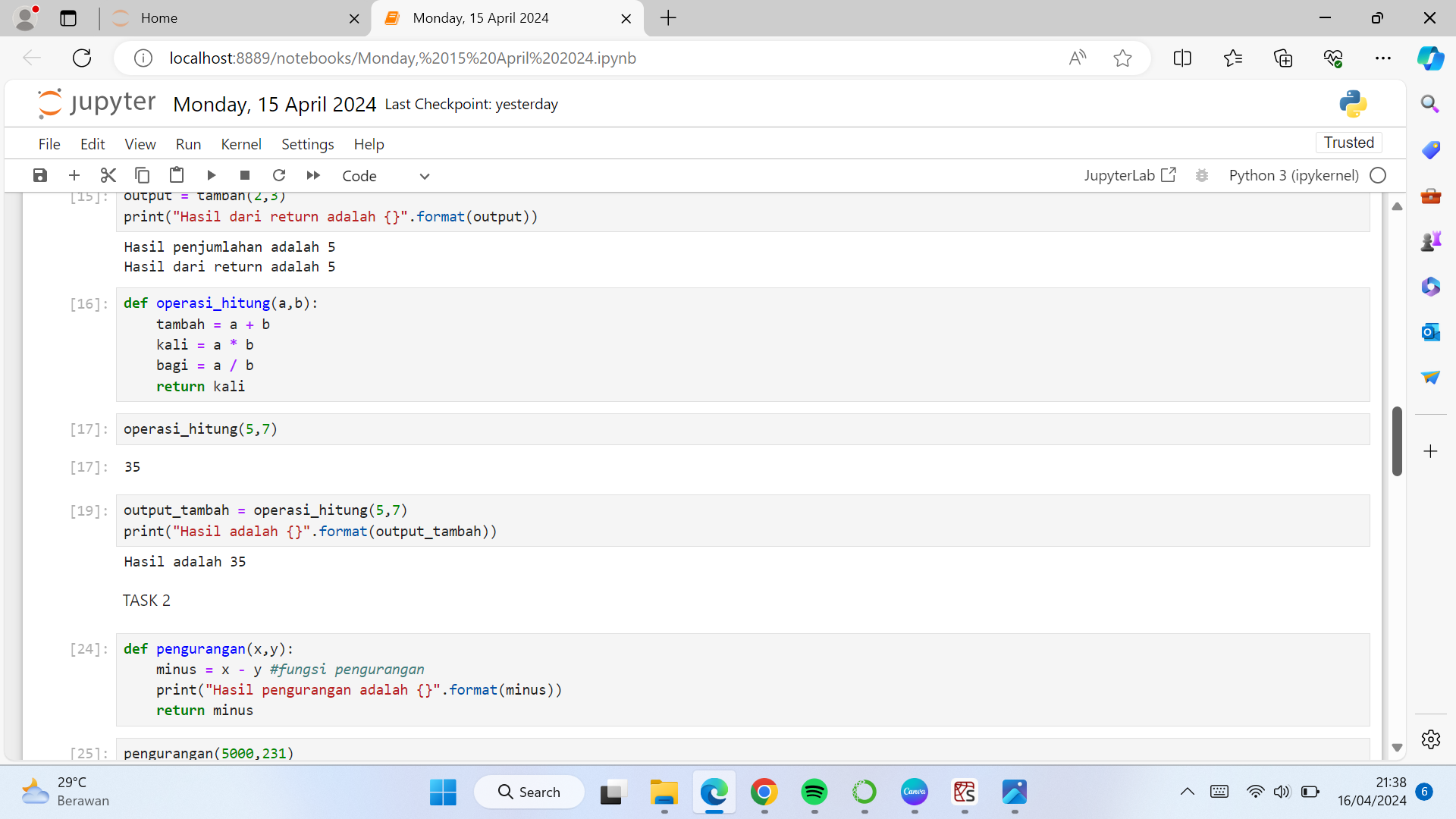


**Return**

The Phython Return statement is a special statement that can use inside in function or method to send the function’s result back to the caller. A Return statement consist of the return keyword followed by an optional return value. The return value of a Python function can be any Python object.

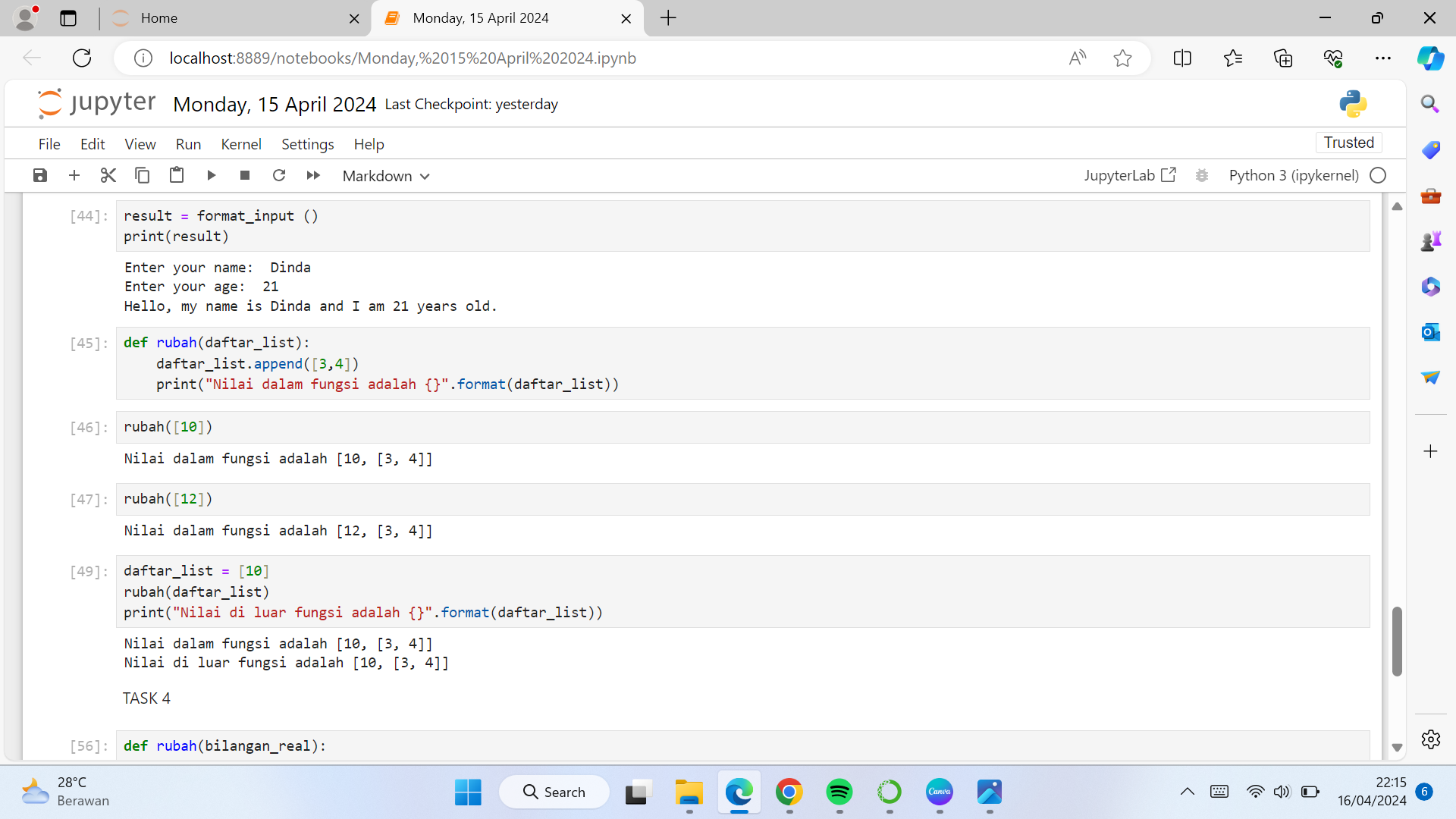


* The return value of a function can be stored in a variable.
* This will differentiate a function that returns a value from a function that does not return a value (often referred to as a procedure).



**Pass by reference vs value**

The difference between pass-by-reference and pass-by-value is that modifications made to arguments passed in by reference in the called function have effect in the calling function, whereas modifications made to arguments passed in by value in the called function can not affect the calling function.



**Selasa**

**NUMPY**

**What is NumPy?**

NumPy is a Python library used for working with arrays. It also has functions for working in the domain of linear algebra, fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely. NumPy stands for Numerical Python.

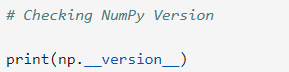
At the core of the NumPy package, is the *ndarray* object. This encapsulates *n*-dimensional arrays of homogeneous data types, with many operations being performed in compiled code for performance. There are several important differences between NumPy arrays and the standard Python sequences:

1. NumPy arrays have a fixed size at creation, unlike Python lists. Changing the size of an *ndarray* will create a new array and delete the original.
2. The elements in a NumPy array are all required to be of the same data type, and thus will be the same size in memory.

**The next step is how to apply NumPy in Python**

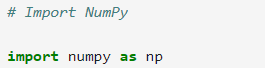
1. **Checking NumPy Version**

The “print(np.version)" syntax is used to determine the installed version of NumPy in Python.



1. **Import NumPy**

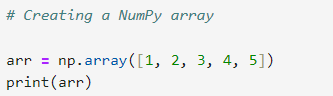
The “import numpy as np” statement is used to import the NumPy library into Python, allowing you to use its functions for numerical calculations and array manipulations.

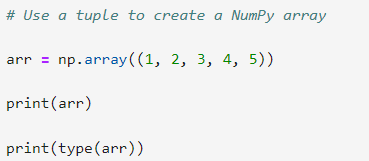


1. **Creating a NumPy Array**

To create an ndarray, we can pass a list, tuple or any array-like object into the array() method, and it will be converted into an ndarray.

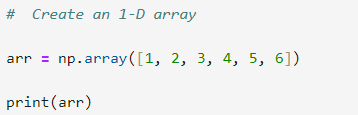
The syntax used to create a NumPy array from a list or tuple is the same, the only difference being the use of square brackets for a list [ ] and parentheses for a tuple ().



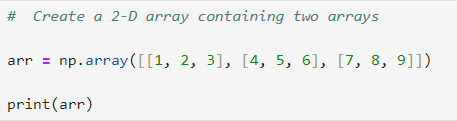


1. **Creating an array with specific dimensions**

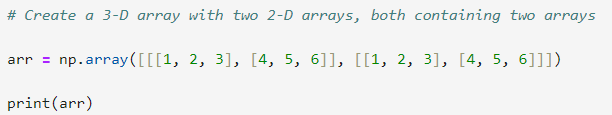
* 1 dimension: 1D arrays in NumPy are essentially like lists in Python. They are sequences of elements arranged in a single line.



* 2 dimension: 2D arrays in NumPy are like matrices. They have rows and columns, arranged in a grid format.

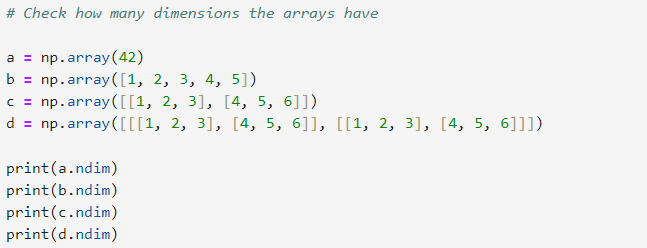


* Creating a 3-dimensional array with two 2-dimensional arrays, both containing two arrays, can be achieved using NumPy as follows:

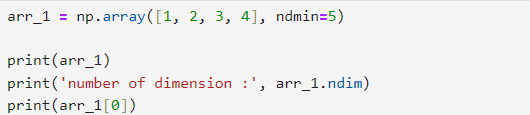


1. **Check how many dimensions the arrays have**

The "a = np.array" syntax is used to store an array variable into the NumPy library. And the "print(a.ndim)" syntax is used to provide feedback by displaying dimensional information to the user.



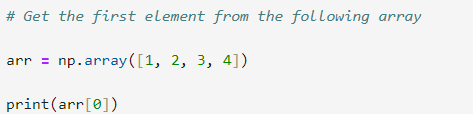
1. **Changing one dimension to another**

****

The syntax "ndmin=5" is used to reshape the dimension created into the fifth dimension.

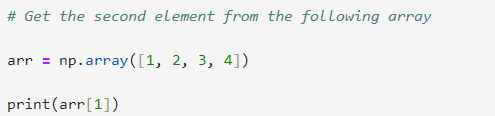
1. **Get the results of the desired elements of the following array**

* Get the first element from the following array

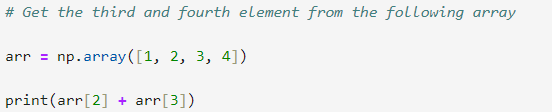


To retrieve the first element, you use [0] because the indexing of elements starts from 0, then 1, 2, 3, etc.

* Get the second element from the following array



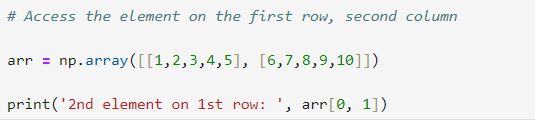
* Get the third and fourth element from the following array



The "(arr[2] + arr[3])" syntax is used to find two elements, namely the third element and the fourth element.

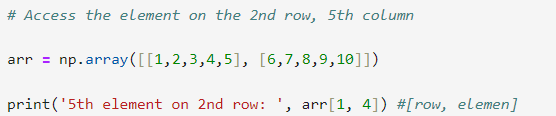
1. **Accessing elements from a specified array.**

* Access the element on the first row, second column

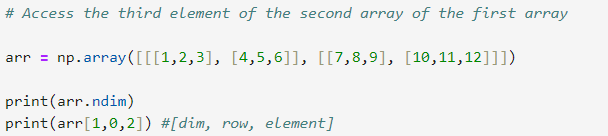


The "arr[0, 1]" syntax is used to retrieve the element at 1st row and 2nd column.

* Access the element on the 2nd row, 5th column



* Access the third element of the second array of the first array

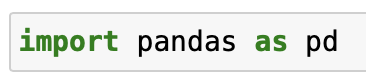


The "arr[0, 1]" syntax is used to access the 2nd dimension, the 1st row, and the 3rd element.

**Rabu**

**Pandas**

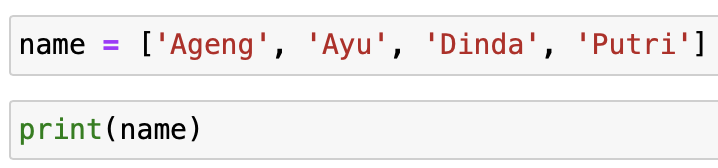
Pandas is a powerful and widely used open-source Python library primarily used for data manipulation and analysis. It provides high-performance data structures and tools for working with structured data, making it an essential tool for data scientists and analysts.



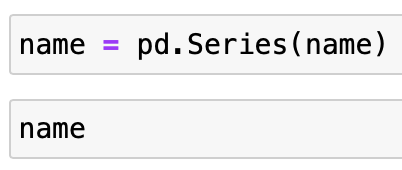
Pandas has two main objects: Series and DataFrame.

### **Series**

The Series object is a ONE-dimensional array that can store various types of data, such as integers, floats, strings, and others. It does not have column names as it consists of only one column. Each element in a Series has a label called an index.



Convert it into a Series



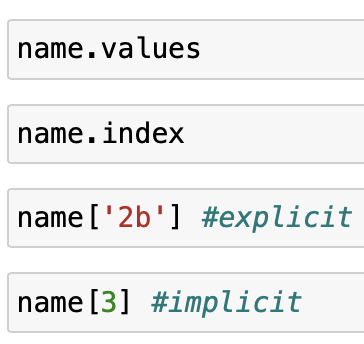
Convert the Series to an array



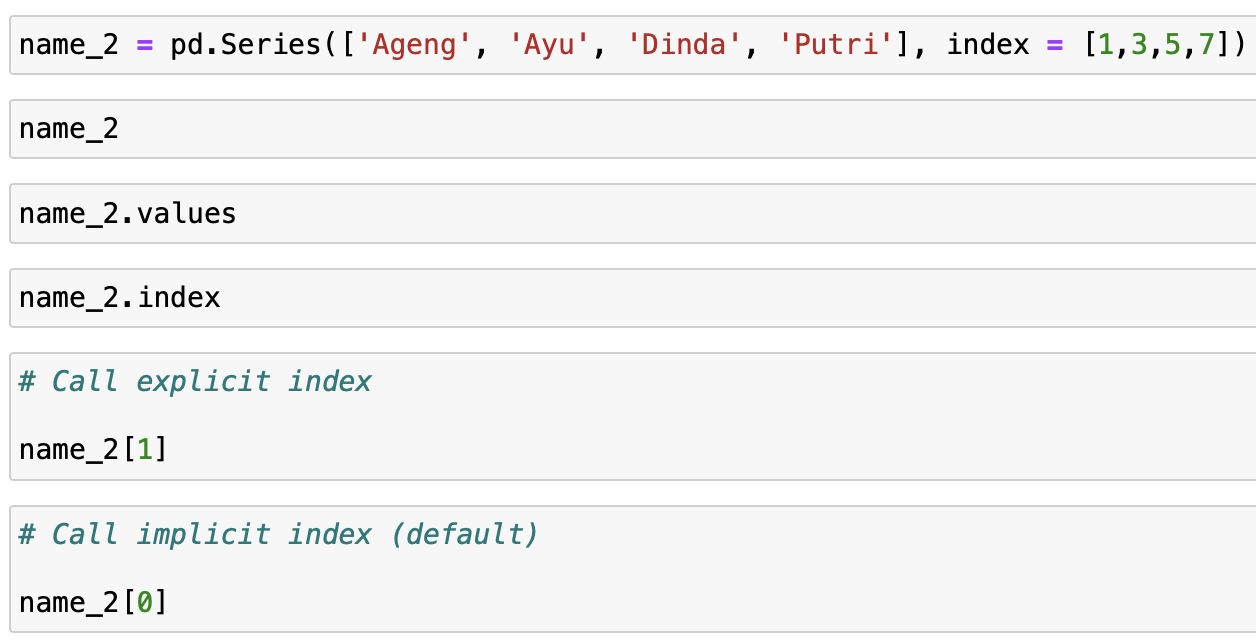
Display the indices.

The indices are represented as a range, where the start point is inclusive, and the stop point is exclusive in the range.

* **Implicit indexing**: Default numerical index assigned based on positional order.
* **Explicit indexing**: Custom labels or values are assigned to uniquely identify rows or columns.

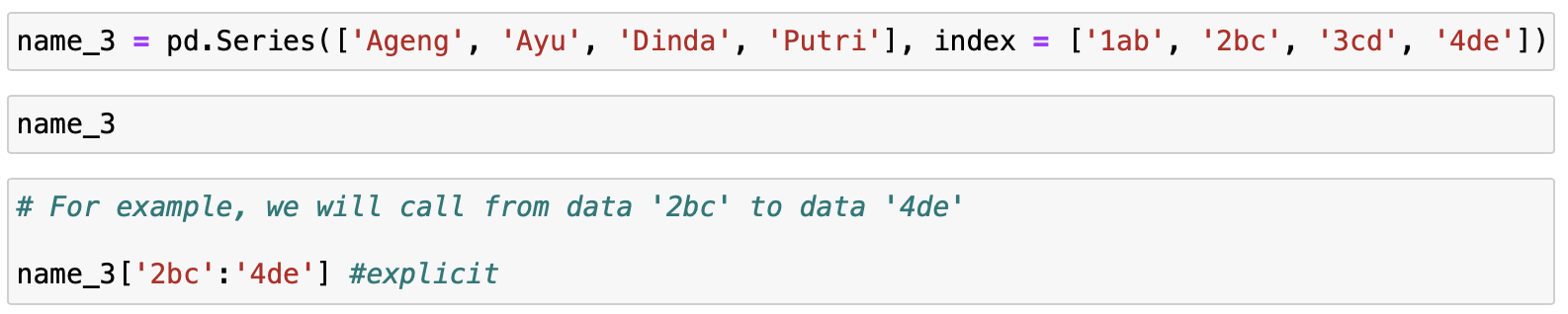


When the implicit and explicit indices are the same, when we call the data, it will rely only on its explicit indices.



The result of name\_2[0] is an error due to the similarity between the explicit indices and the implicit indices.

We'll now try to perform data-slicing

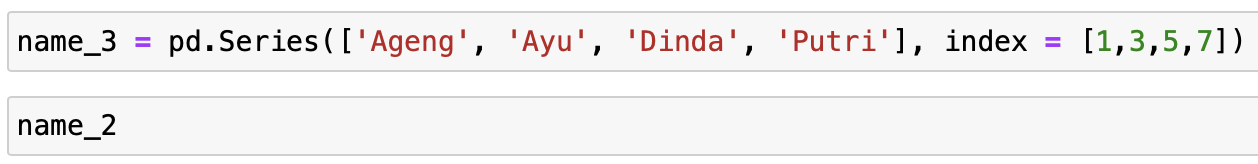


But if we slice its implicit indices, only the start point will appear, because implicit indices are in the form of a range



### **Ioc and iloc**

Example of data where some implicit and explicit indices are the same.



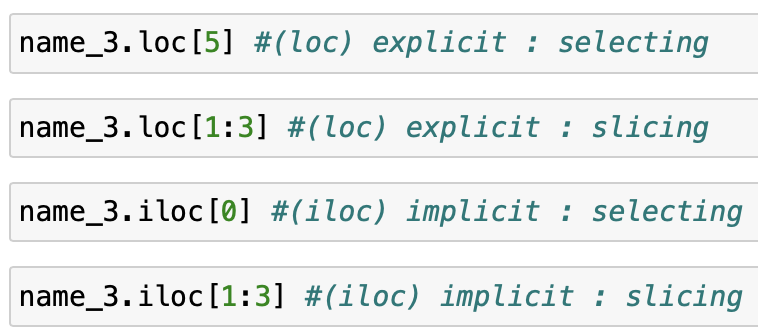
When we access a single index, it will display its explicit index.



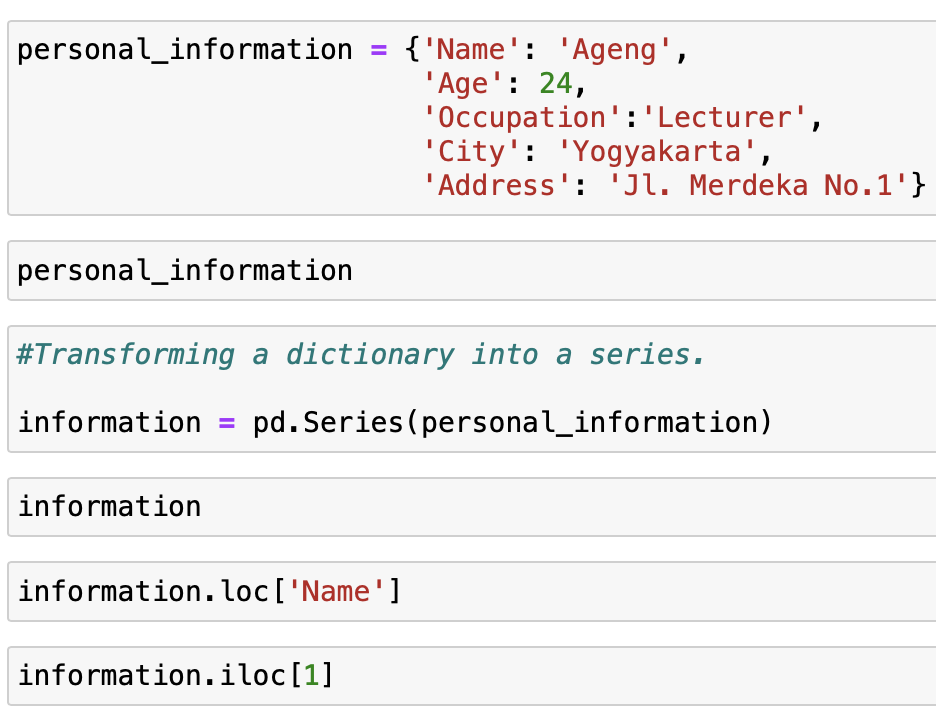
When explicit and implicit indices are the same, inconsistencies occur as in the case above.

To address this inconsistency, we will use the principles of loc and iloc.

loc is used to call its explicit indices, while iloc is used to call its implicit indices.



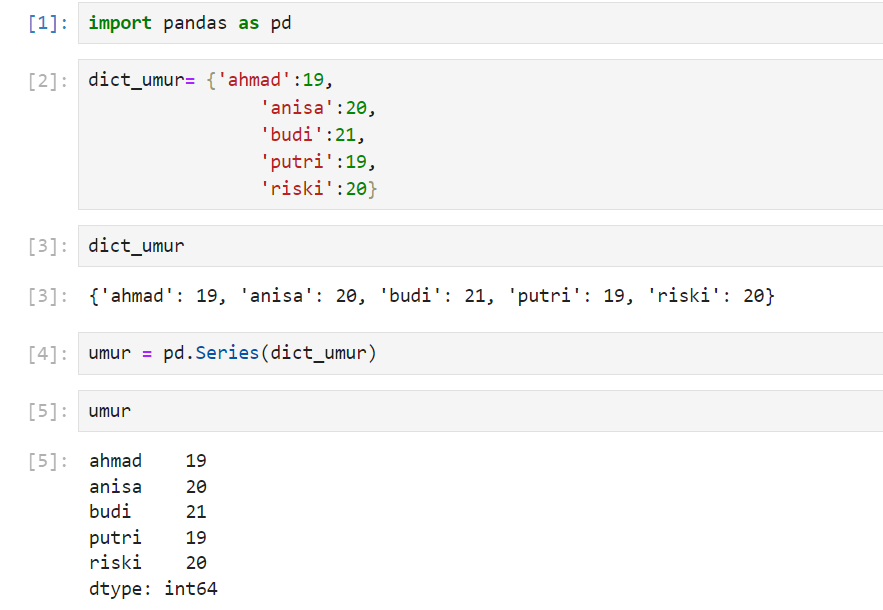
#### **Dictionary -- Series**



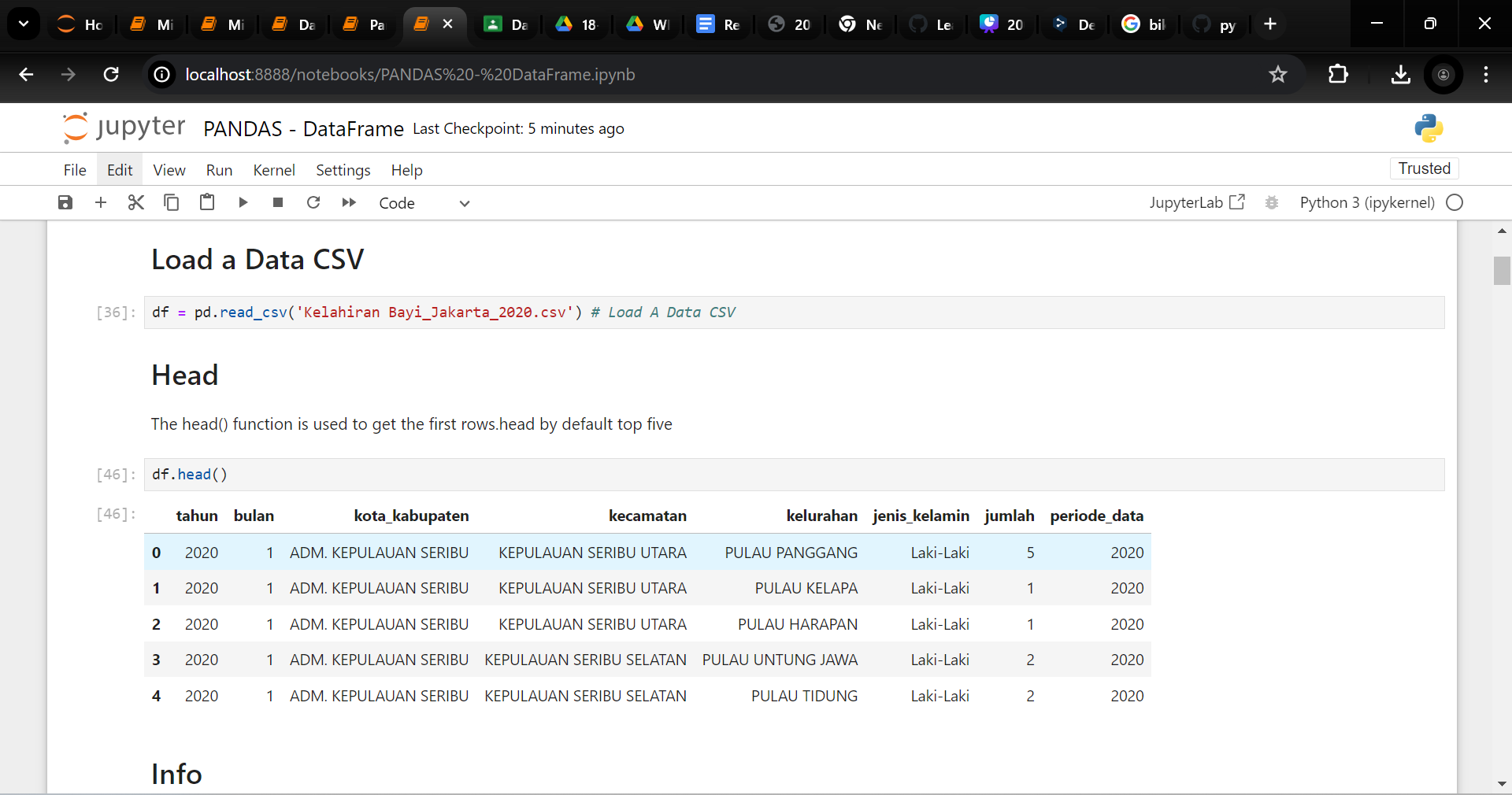
**Kamis**

**DataFrame**

DataFrame is a two-dimensional tabular data structure with labeled axes (rows and columns). This allows for easy manipulation and analysis of data. Essentially, a DataFrame is a collection of series, with at least one series.

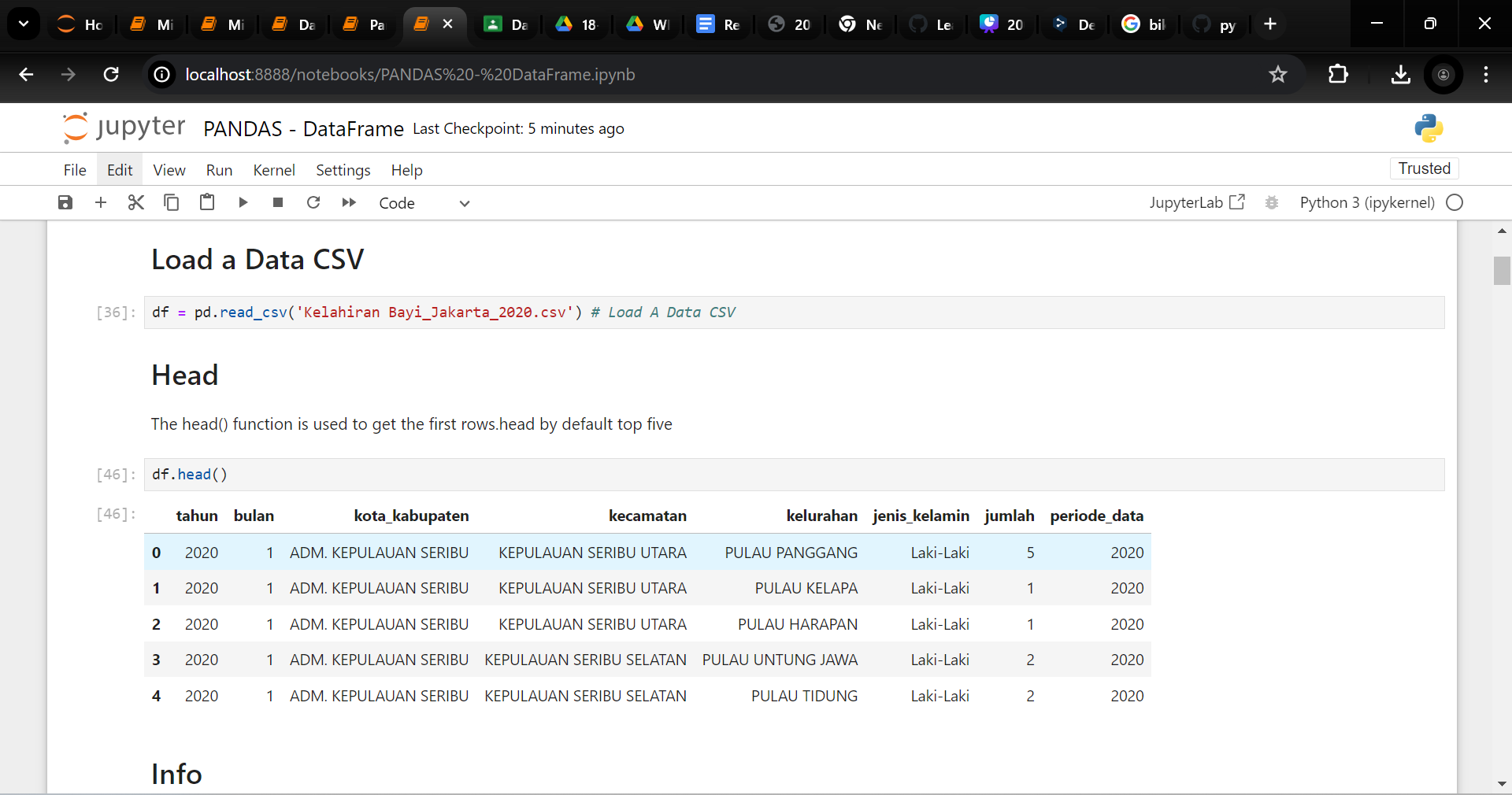
DataFrame With 2 Series : 

**Load Data CSV**

Load Data ‘Kelahiran\_Bayi\_Jakarta\_2020’.csv that has been aplouded in the same folder in Python.

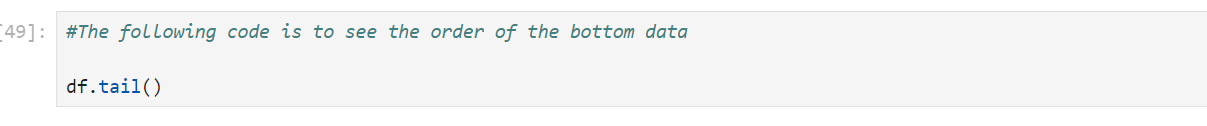
**Head()**

Head() function to a viewing top in data by default is top five can be customized as required.



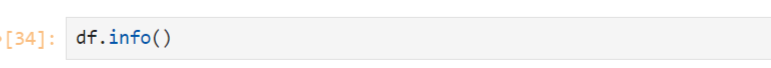
**Tail()**

Tail()returns the last 5 rows if a number is not specified,returns a specified number of last rows.



**Info()**

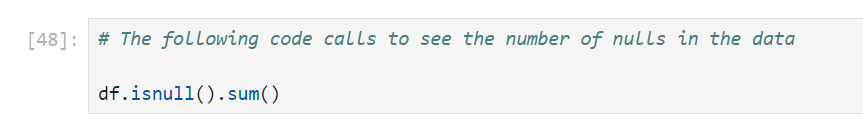
Info()Info method prints information about the DataFrame.



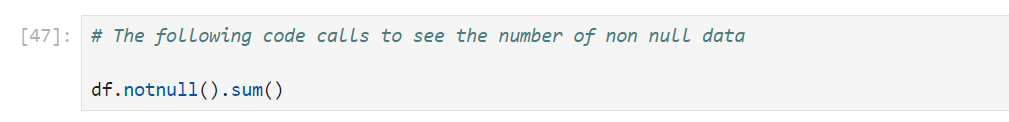
**Index**

Index returns the index information of the DataFrame.

**IsNull()**

isnull used to finds NULL values in DataFrame.

**NotNull()**

notnull used to finds values that are NOT NULL.

**Shape**

Shape is the number of rows and columns on the index of the DataFrame.

**Columns**

Column returns the label of each column in the DataFrame.

**Describe**

Returns a description summary for each column in the DataFrame,describe contains numeric data from managed datasets.



.

**Mean**

Returns a description summary for each column in the DataFrame.

**Median**

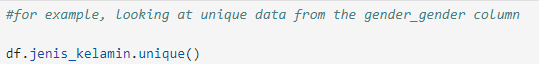
Returns a description summary for each column in the DataFrame.

**Mode**

Returns a description summary for each column in the DataFrame.

**Jum’at**

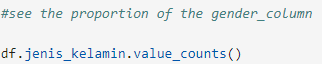
* The df.unique() syntax is used to get unique values in the columns of a pandas DataFrame.



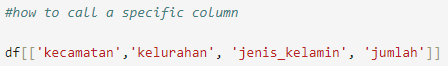
* The df.nunique() is a function in Python that is used to get the unique number of values in a column or axis in a DataFrame.



* The df.type\_value\_counts() is a method on a DataFrame in pandas that is used to count the number of occurrences of each unique value in a column, while ordering them based on the number of occurrences.



* Call a specific column is a notation for retrieving certain columns from a DataFrame df. This notation will return a new DataFrame containing only the specified columns.



* Calling a Python dataset with terms and conditions means retrieving or selecting certain data from a dataset based on certain terms or conditions. This can be done using boolean operators or logical statements, such as ==, !=, <, >, <=, >=, or and, or, not.



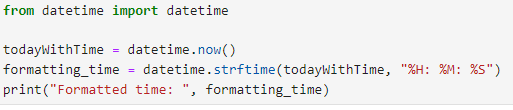
* Calls several columns accompanied by a filter terms and conditions



DateTime

DateTime is a module in Python that deals with real-time data and time. It provides classes and functions to manipulate and format dates and times. Here are some commonly used classes and functions in the DateTime module.

* date : This class represents a date (year, month, and day) without time.
* datetime : This class represents a date and time (year, month, day, hour, minute, second, and microsecond).
* time : This class represents time (hour, minute, second, and microsecond) without a date.



Random Library

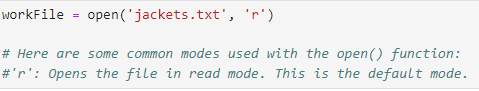
The random module in Python provides various functions to generate random numbers. Here are some commonly used functions in the random library:

* random.random(): This function returns a random float number between 0.0 to 1.0.

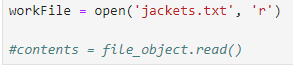


Txt File - Open - Read - Close

* Open : In Python, you can open and read a text file using the built-in open() function. The open() function in Python is used to open a file and return a file (.txt) object.



* Read : The read() function in Python is used to read the contents of a file object that has been opened using the open() function.



* Close : The close() function in Python is used to close a file that has been opened using the open() function. This is important because it frees up system resources and ensures that any changes made to the file are saved.

