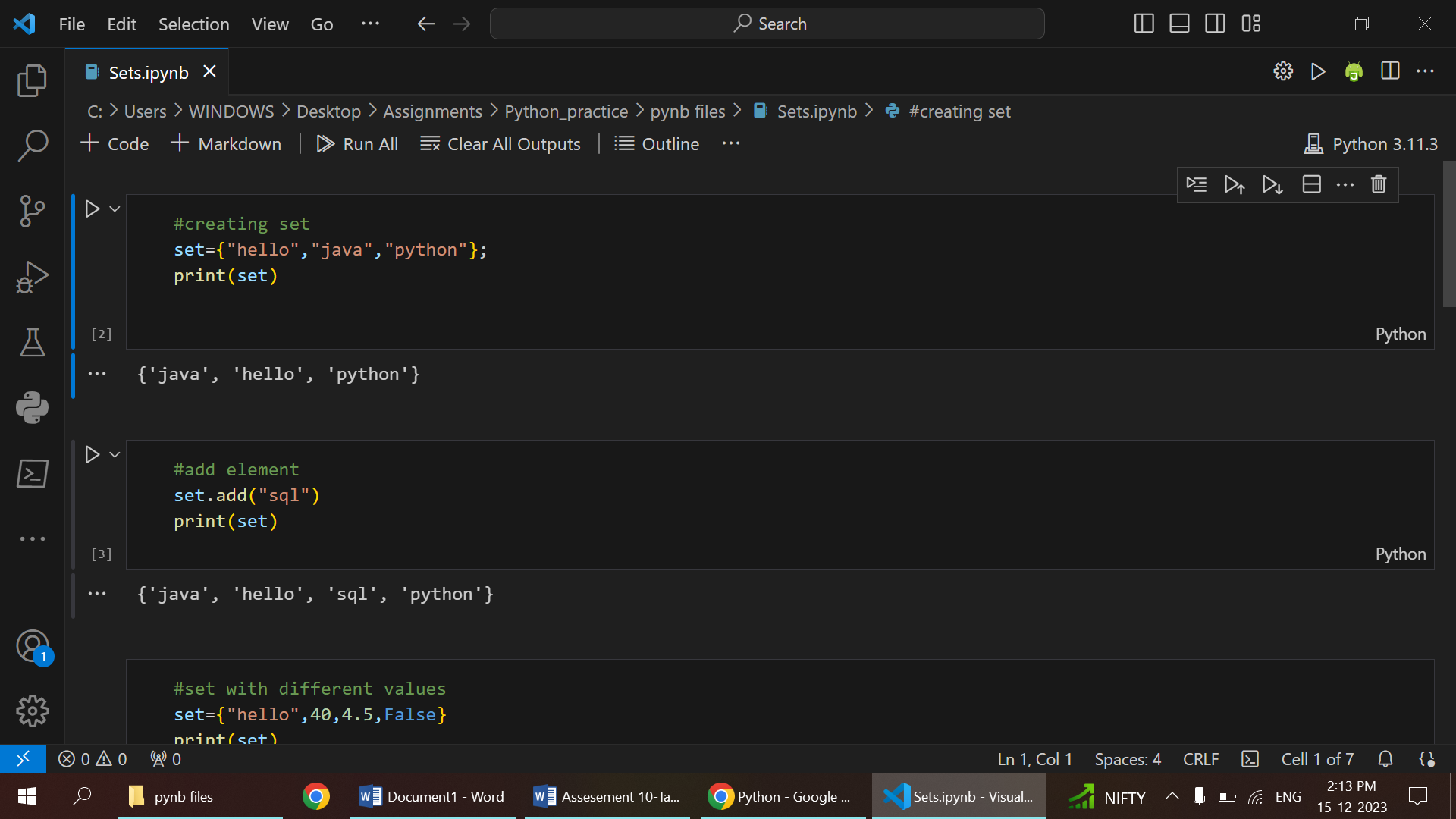
**PYTHON**

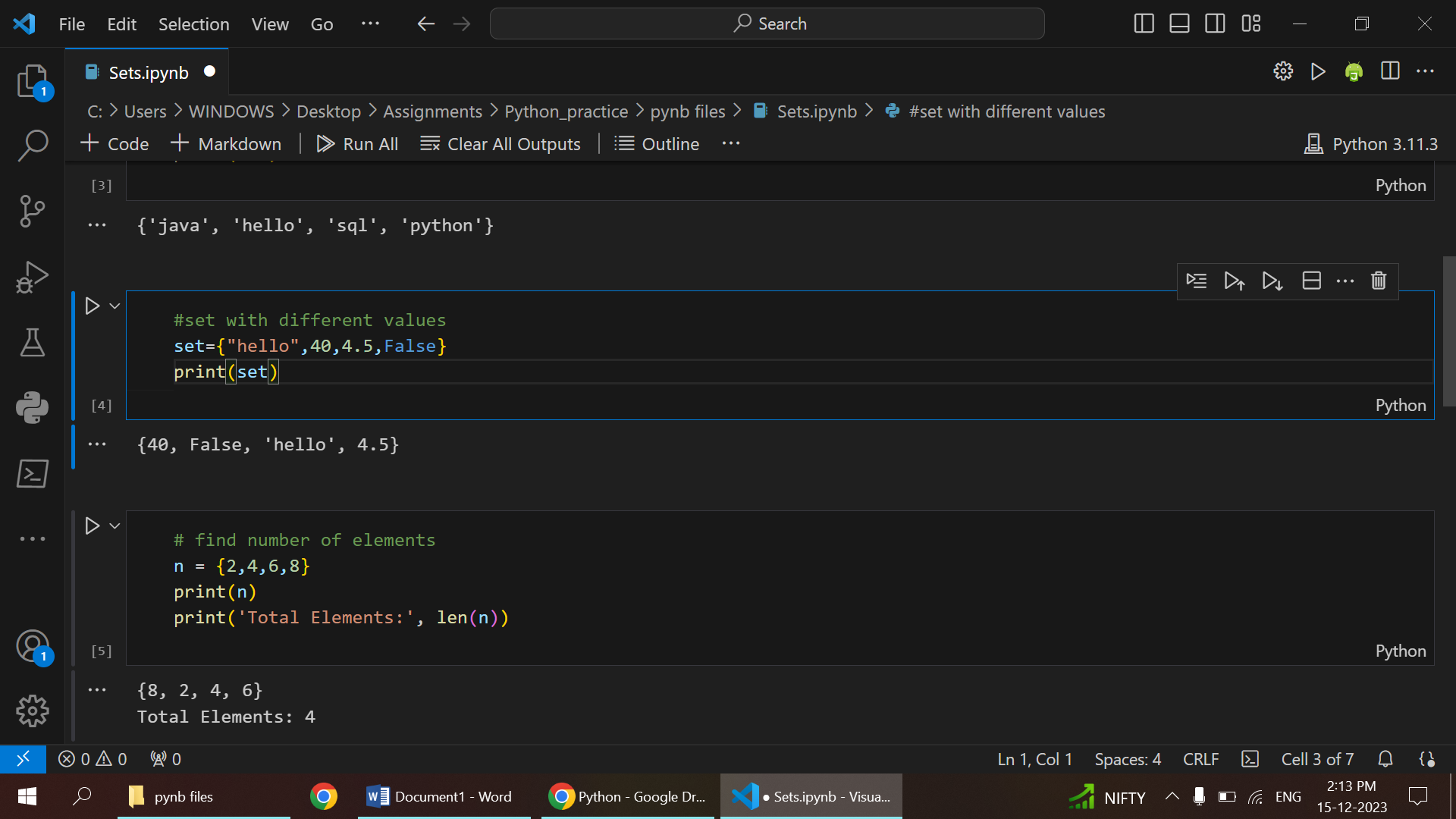
S.R.TAANUSRI

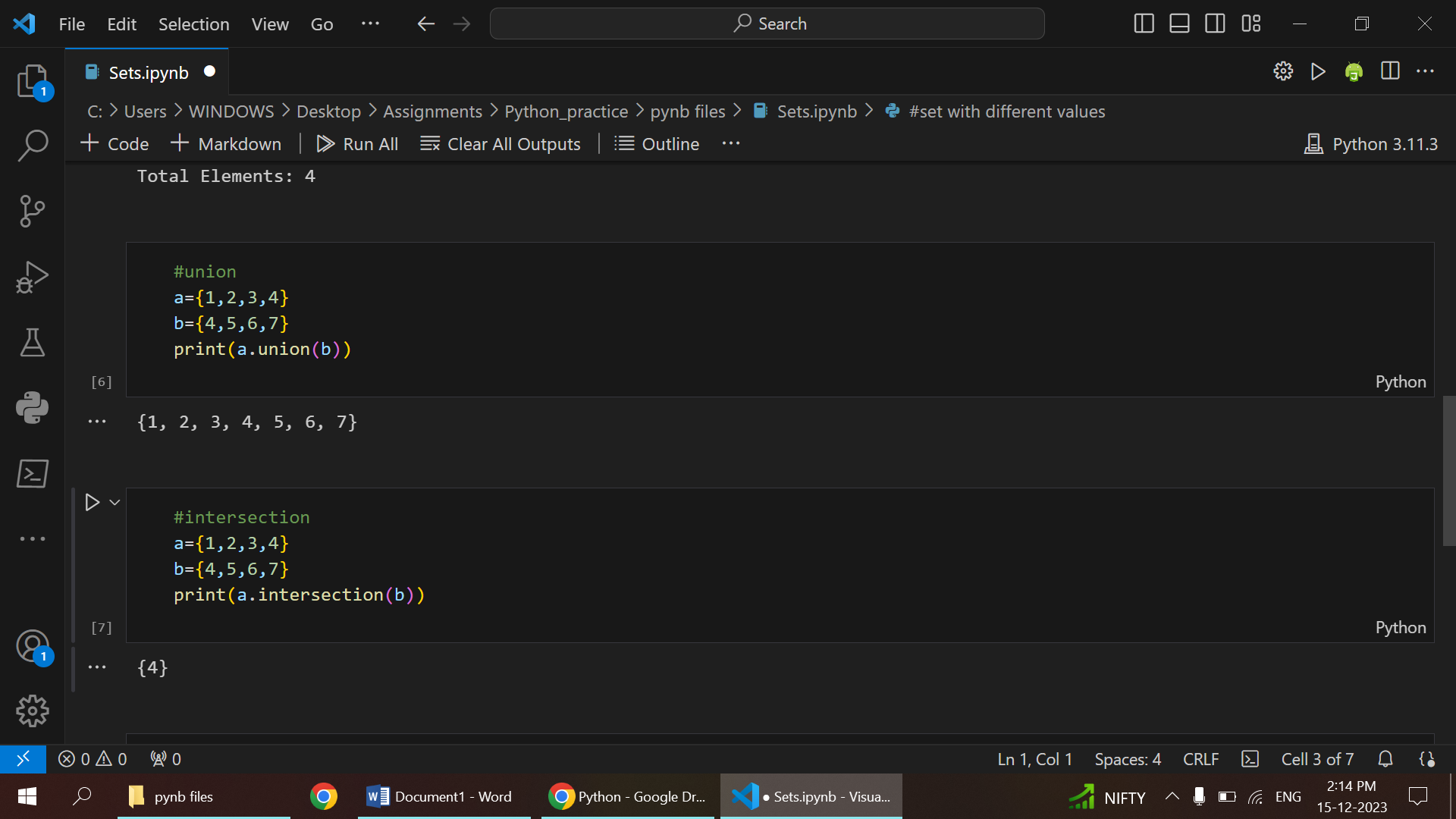
15.12.2023

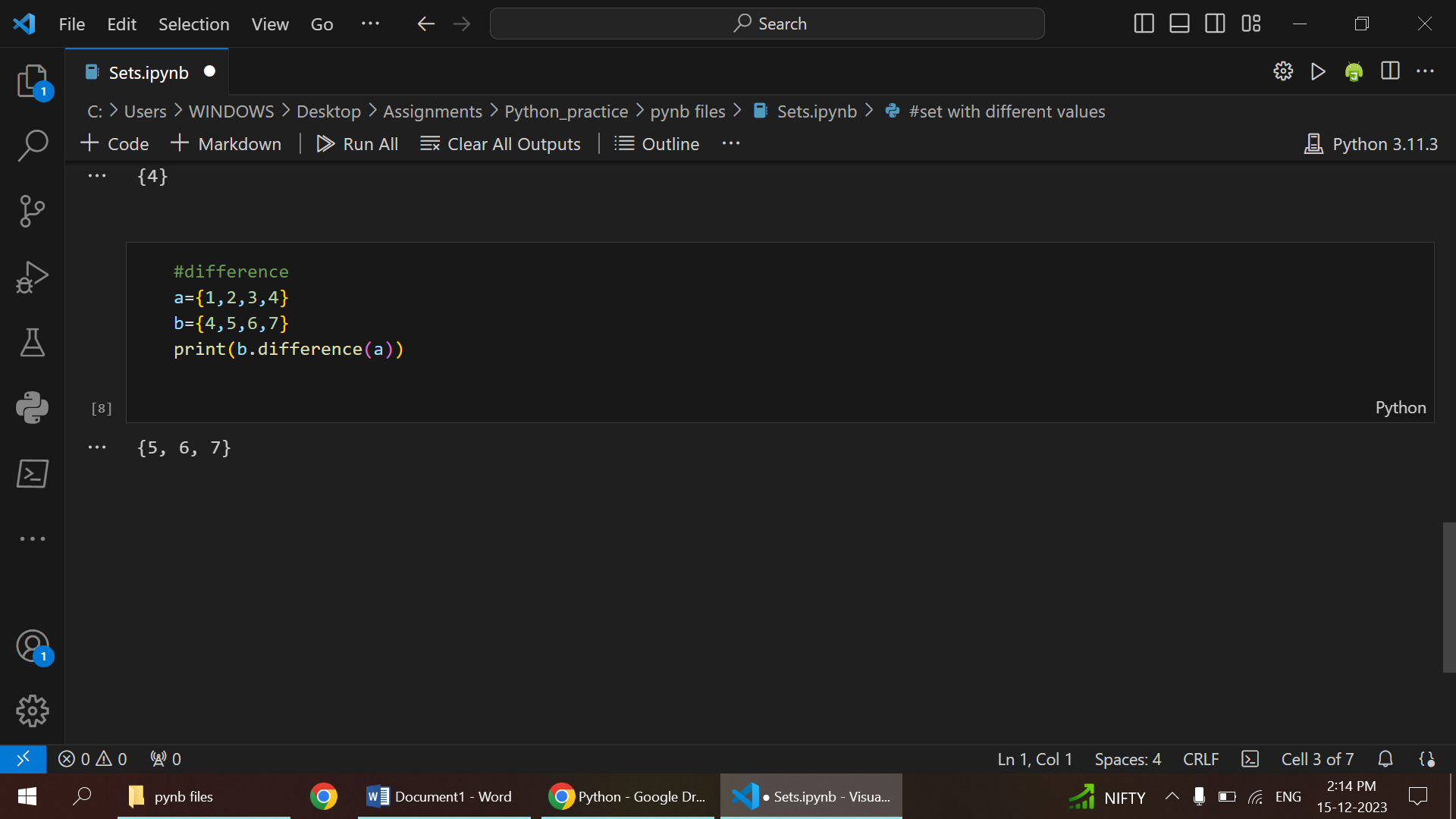
**SETS**

1. A set is a collection of distinct elements or objects.
2. Sets are denoted by curly braces {} and can contain various types of elements.
3. Elements in a set have no specific order, and each element is unique.
4. Membership in a set is indicated by the symbol "∈", meaning an element belongs to the set.
5. Sets can undergo operations such as union (∪∪) and intersection (∩∩), and they follow principles like the absence of order and uniqueness of elements.









**SET METHODS**

* Creation:

Sets can be created using curly braces {} or the set() constructor.

* Addition:

Elements are added to a set using the add() method.

* Removal:

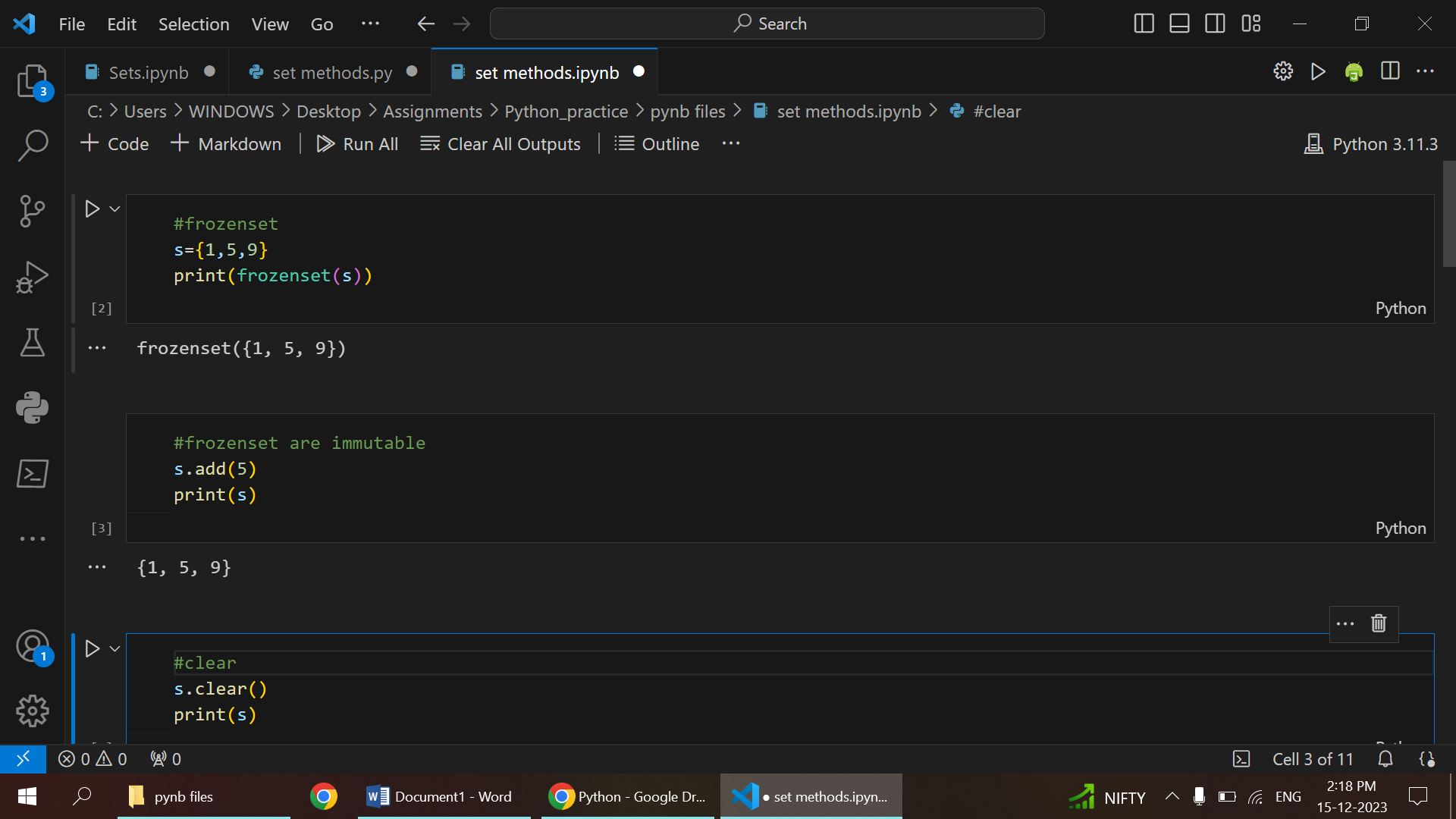
Elements are removed from a set using the remove() or discard() method.

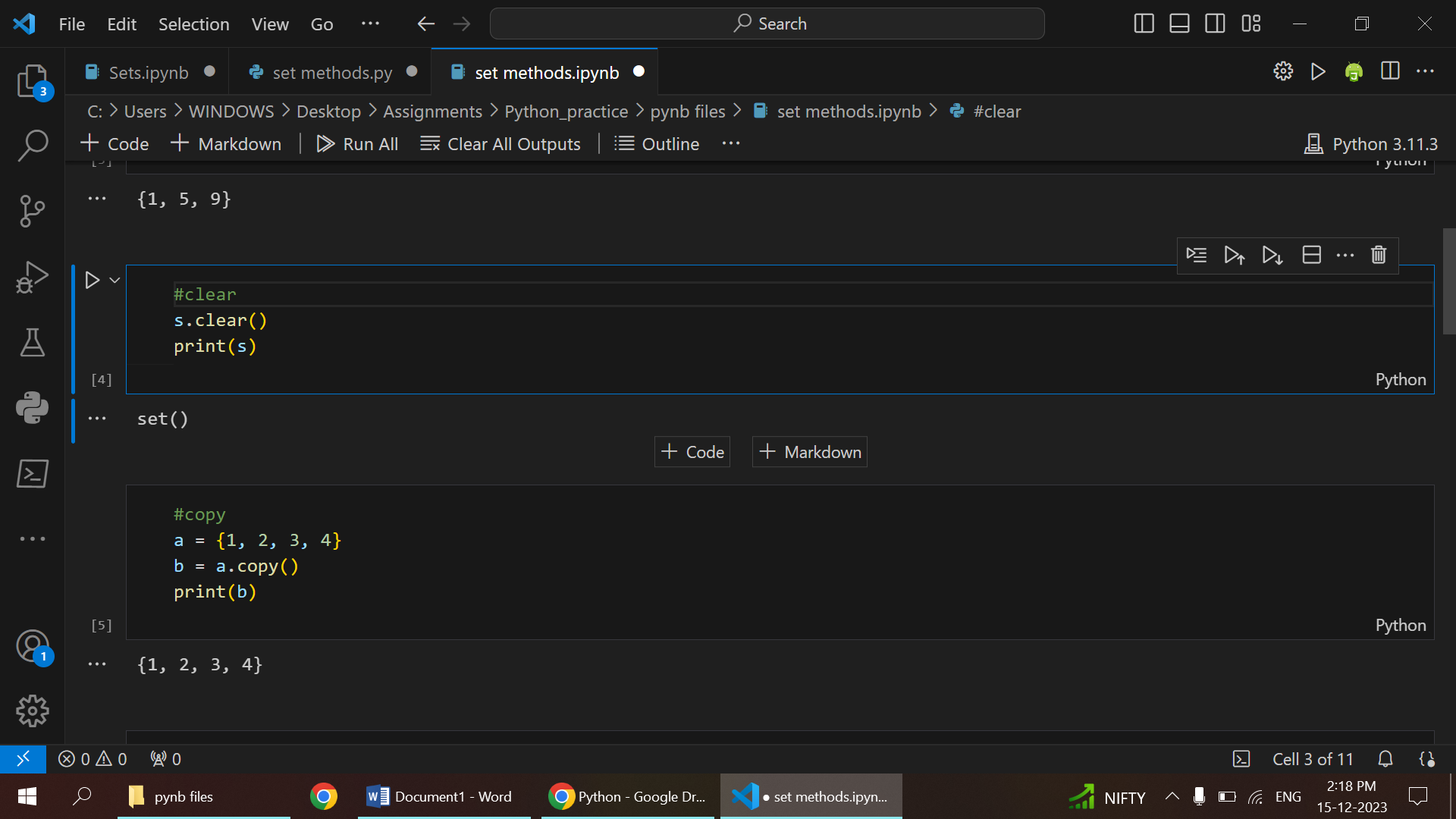
* Operations:

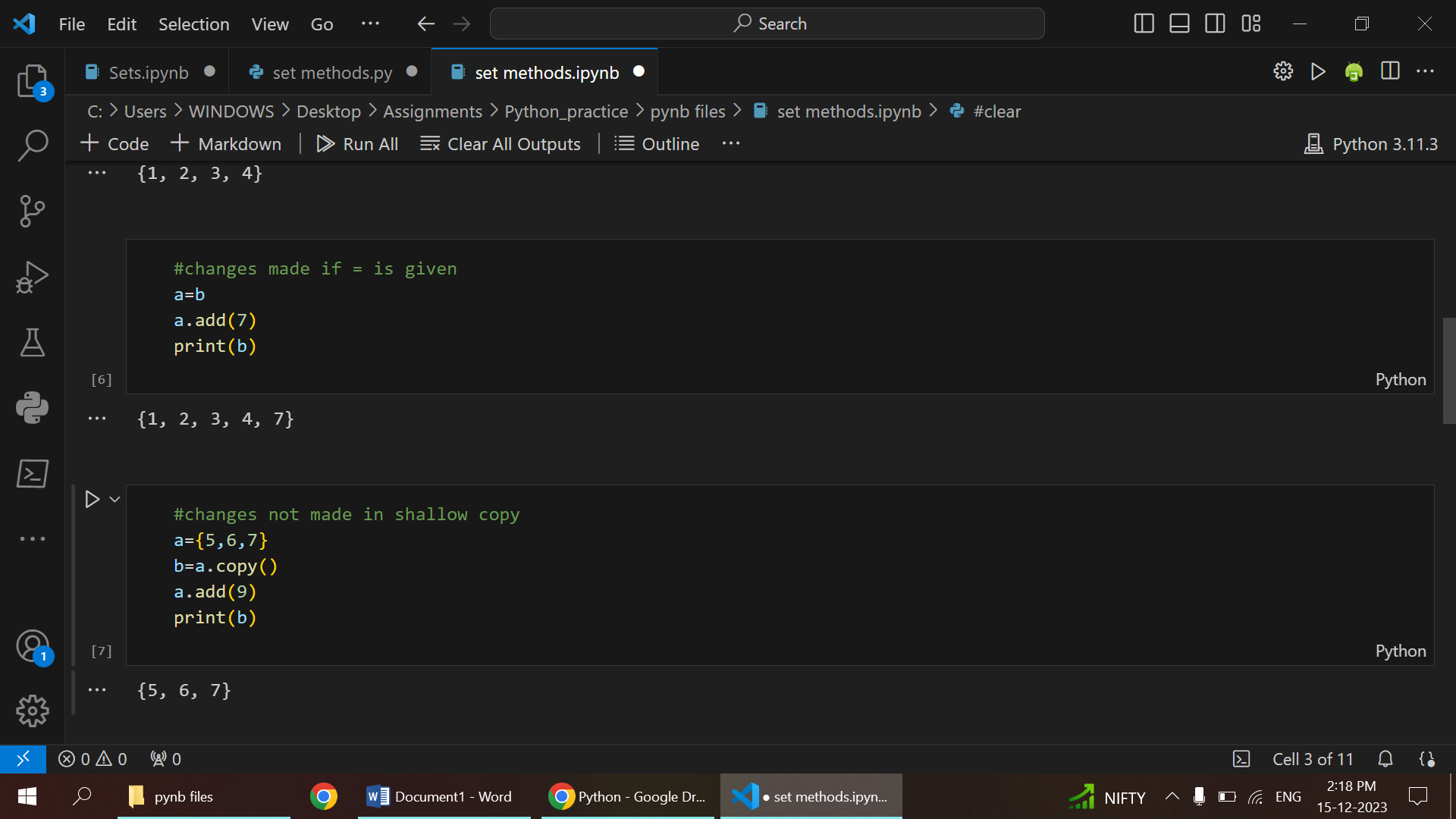
Sets support operations like union (∪∪), intersection (∩∩), and difference (−−).

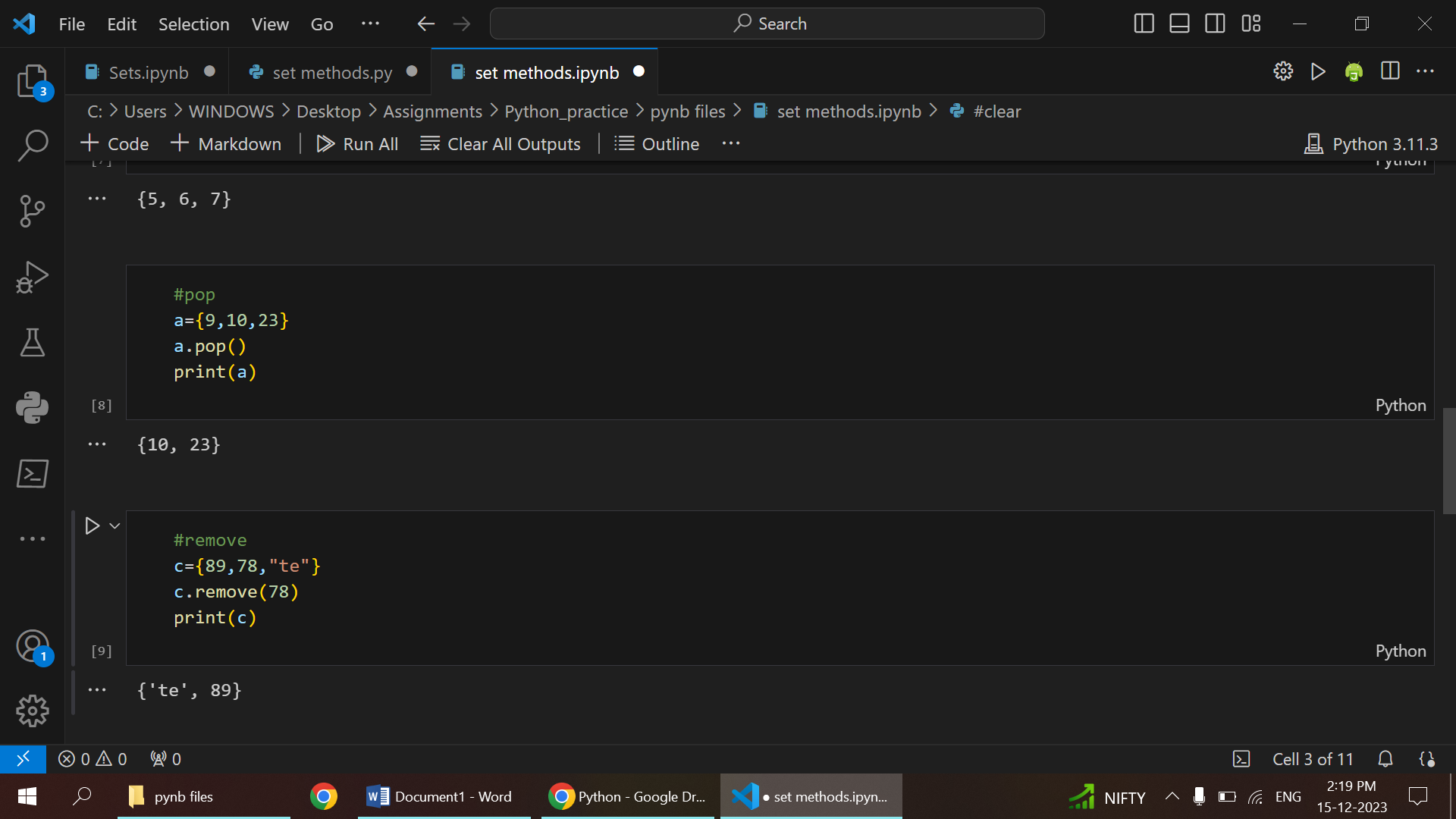
* Membership Testing:

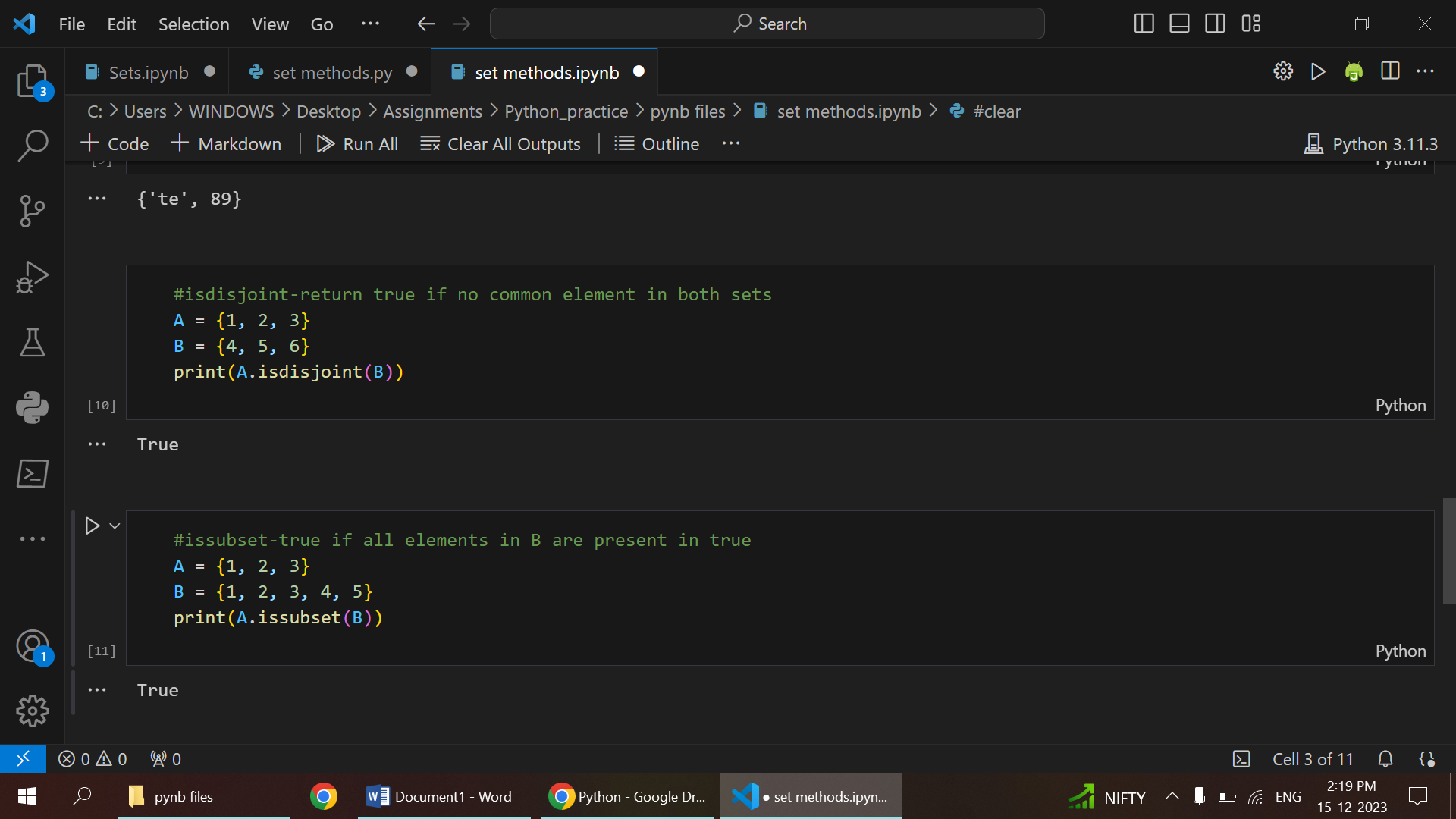
The in keyword or issubset() and issuperset() methods are used to check if an element belongs to a set or if a set is a subset or superset of another.

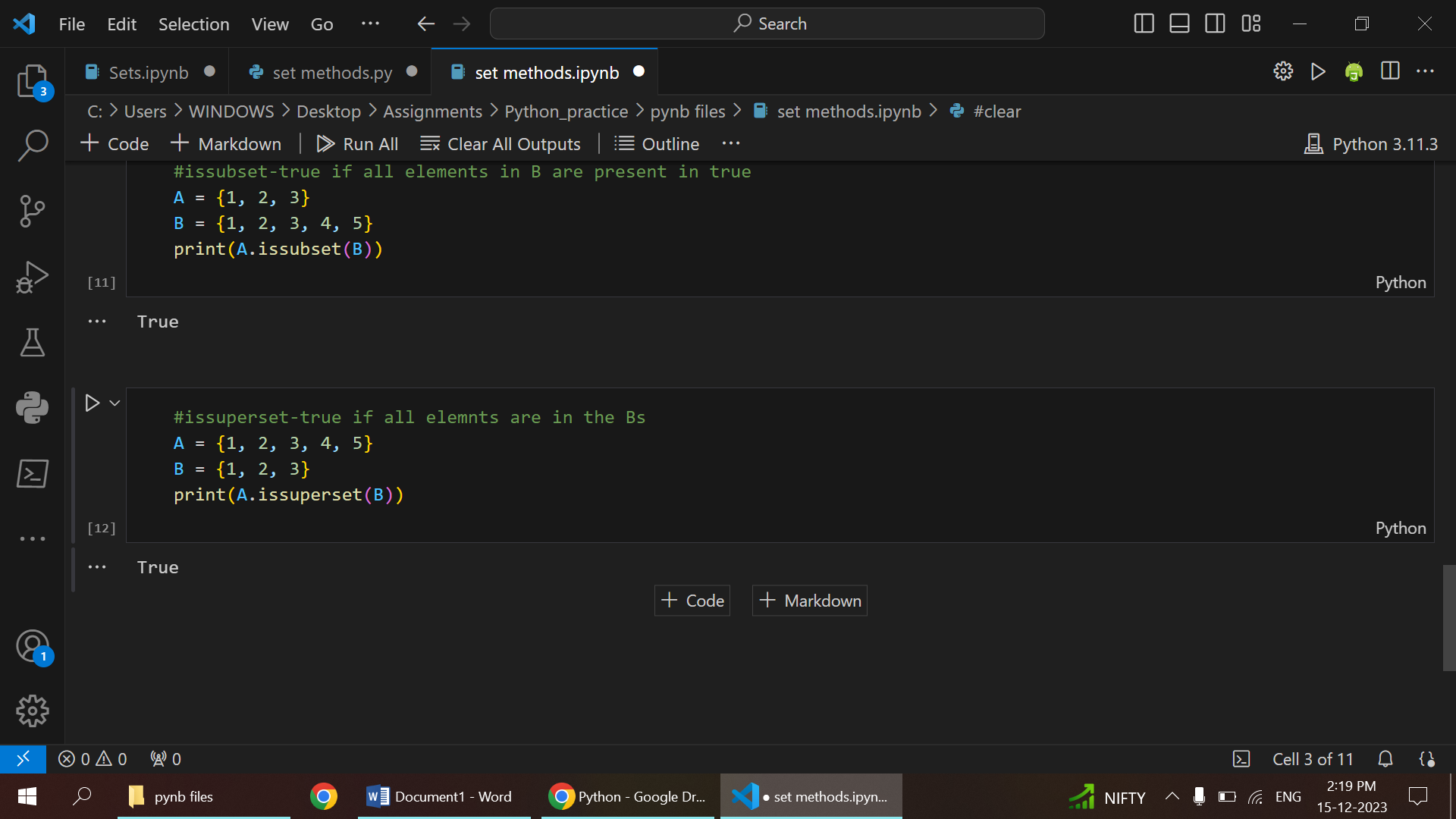






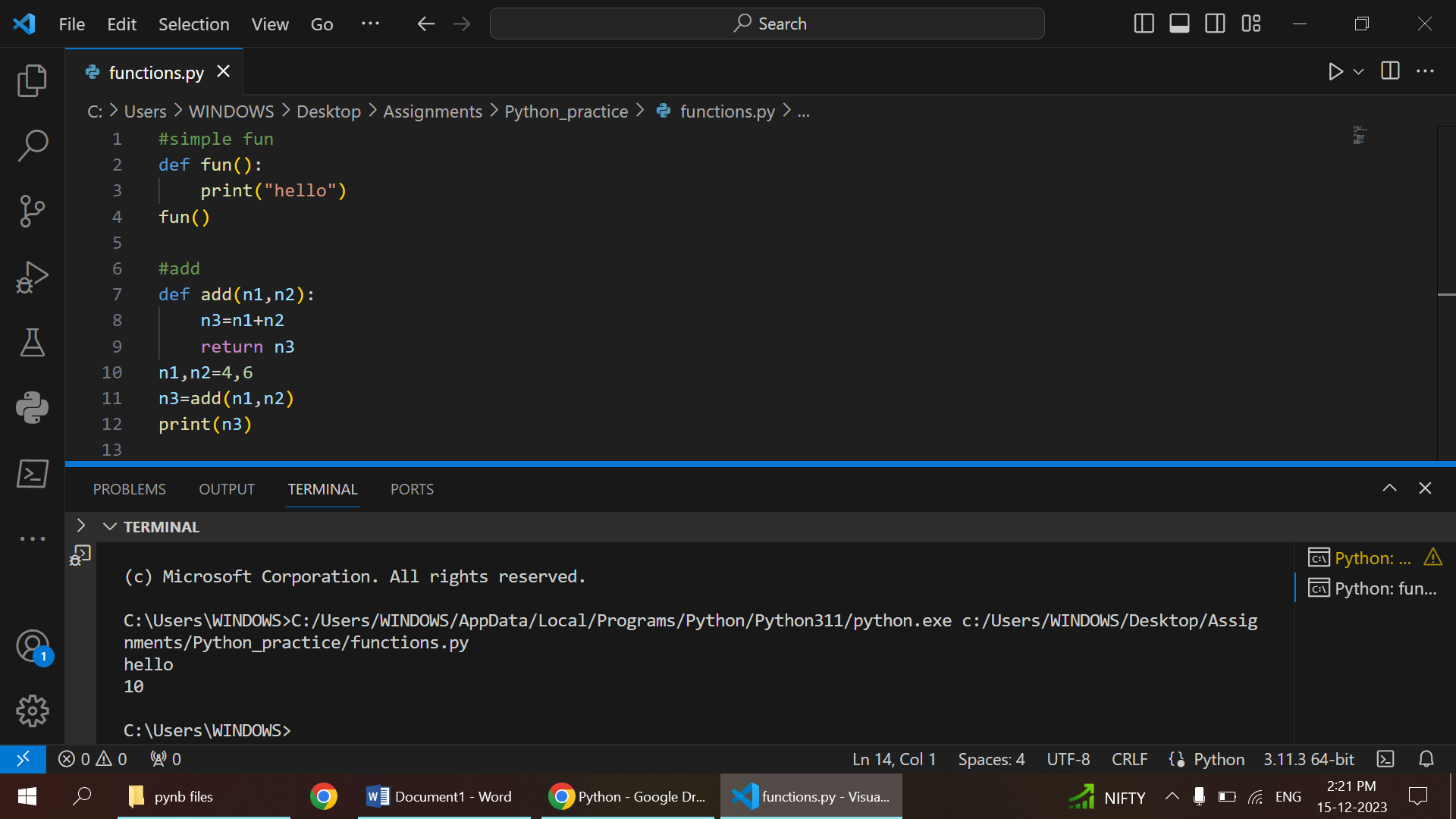


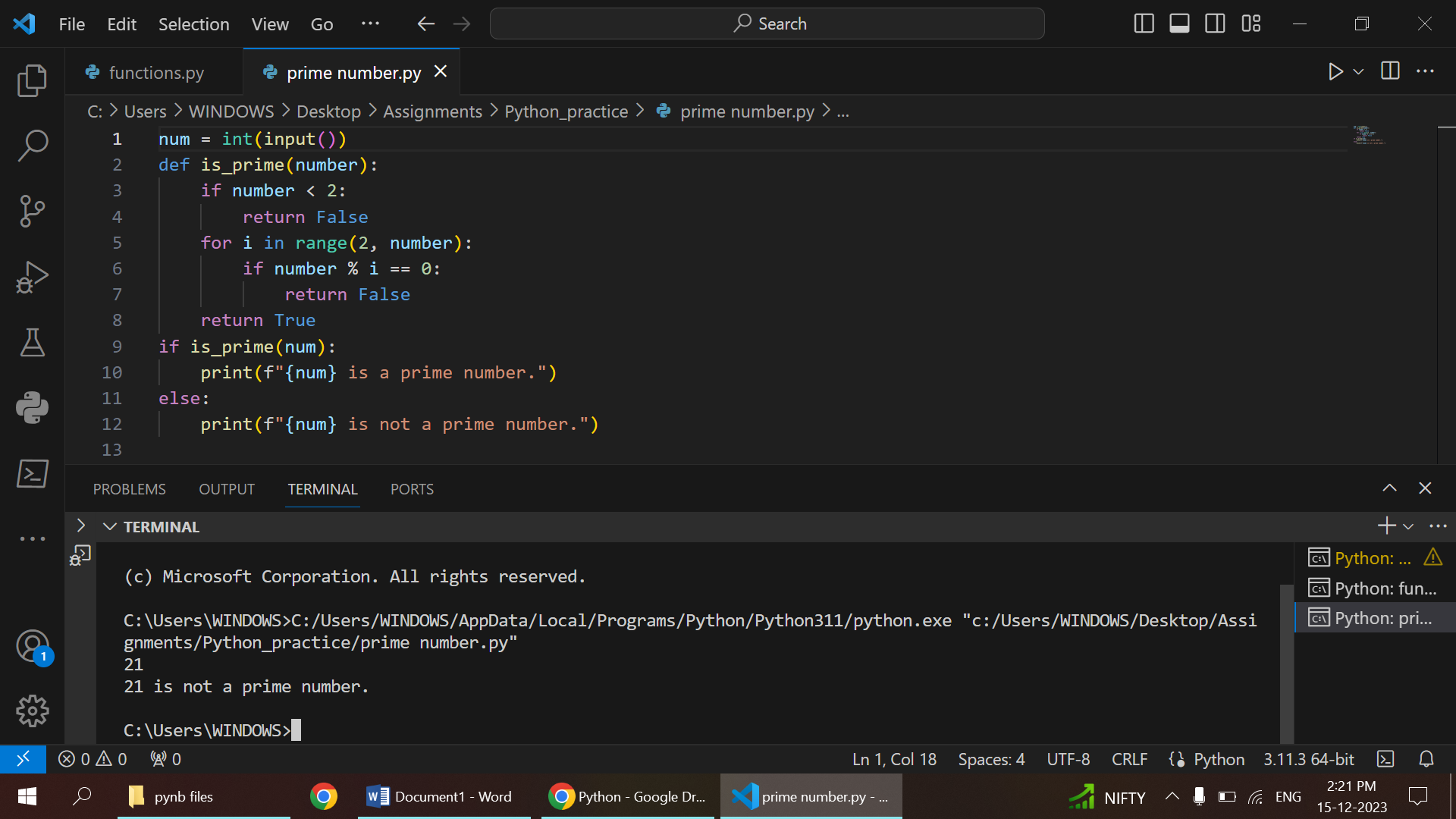




**FUNCTIONS**

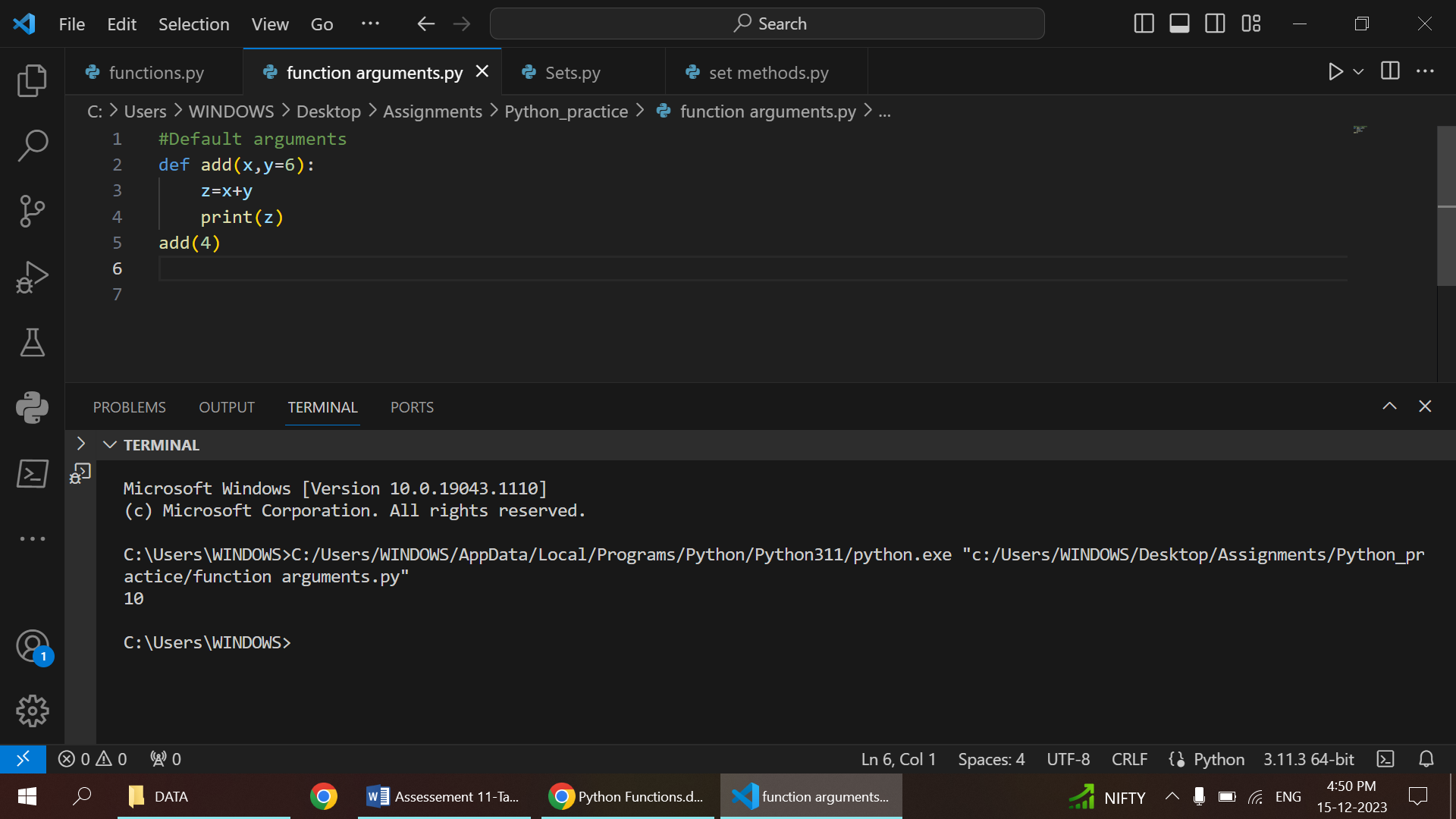
* A Python function is a reusable block of code.
* It is defined using the `def` keyword, followed by a name and parentheses for optional parameters.
* The function body, where the code executes, is indented.
* Functions can accept input values through parameters.
* Output can be produced using the `return` statement.
* To use a function, it is called by name, and any required arguments are passed in parentheses.
* Functions enhance code modularity and readability.





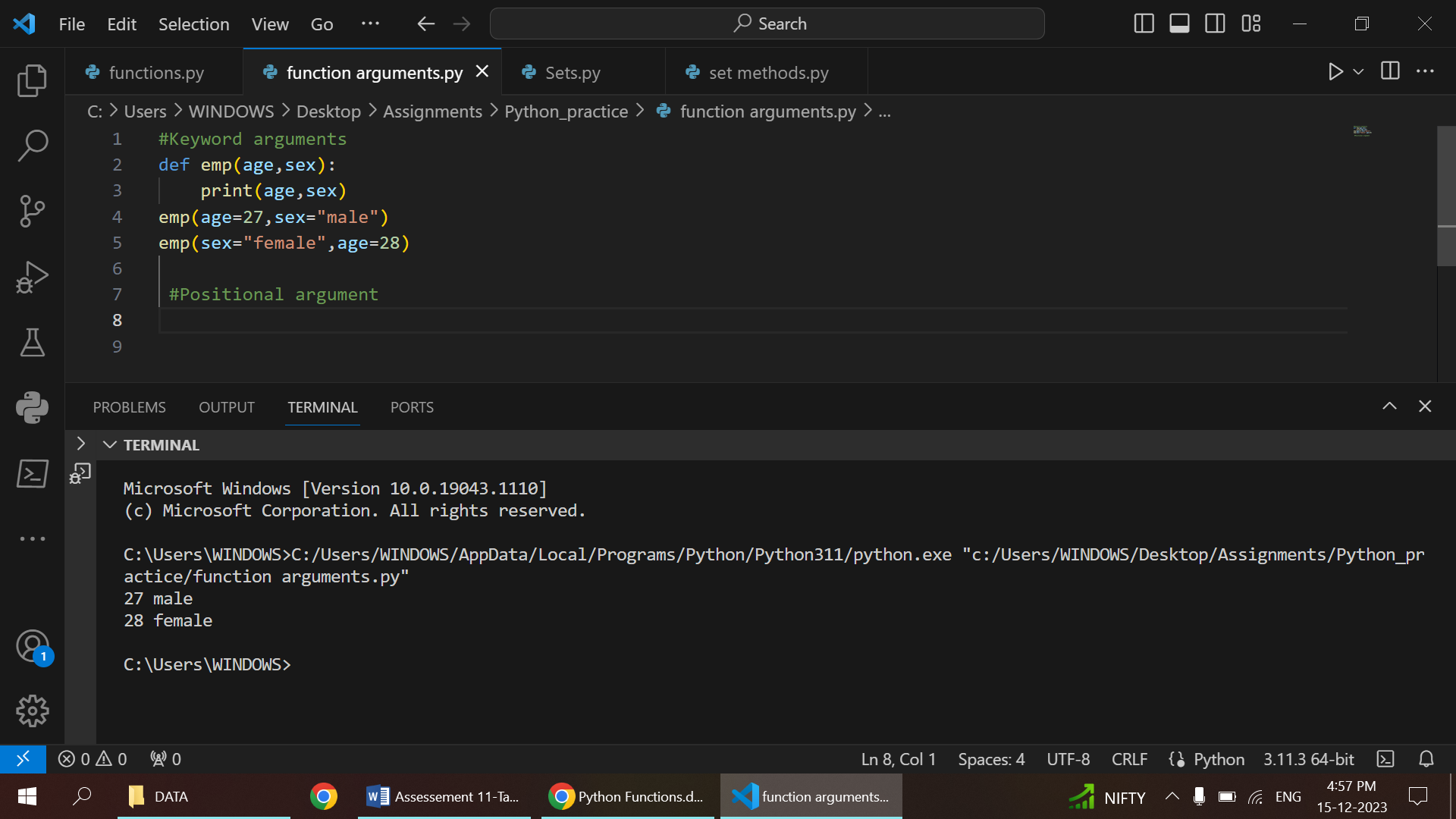
1.DEFAULT ARGUMENTS

* Default arguments provide predefined values for parameters in a Python function.
* Syntax involves assigning default values during function parameter
* Omitting a default argument during function call results in the default value being used.
* enhance a function's adaptability, offering sensible defaults while allowing customization.
* Must follow non-default ones in the function definition; non-defaults must be specified before defaults.



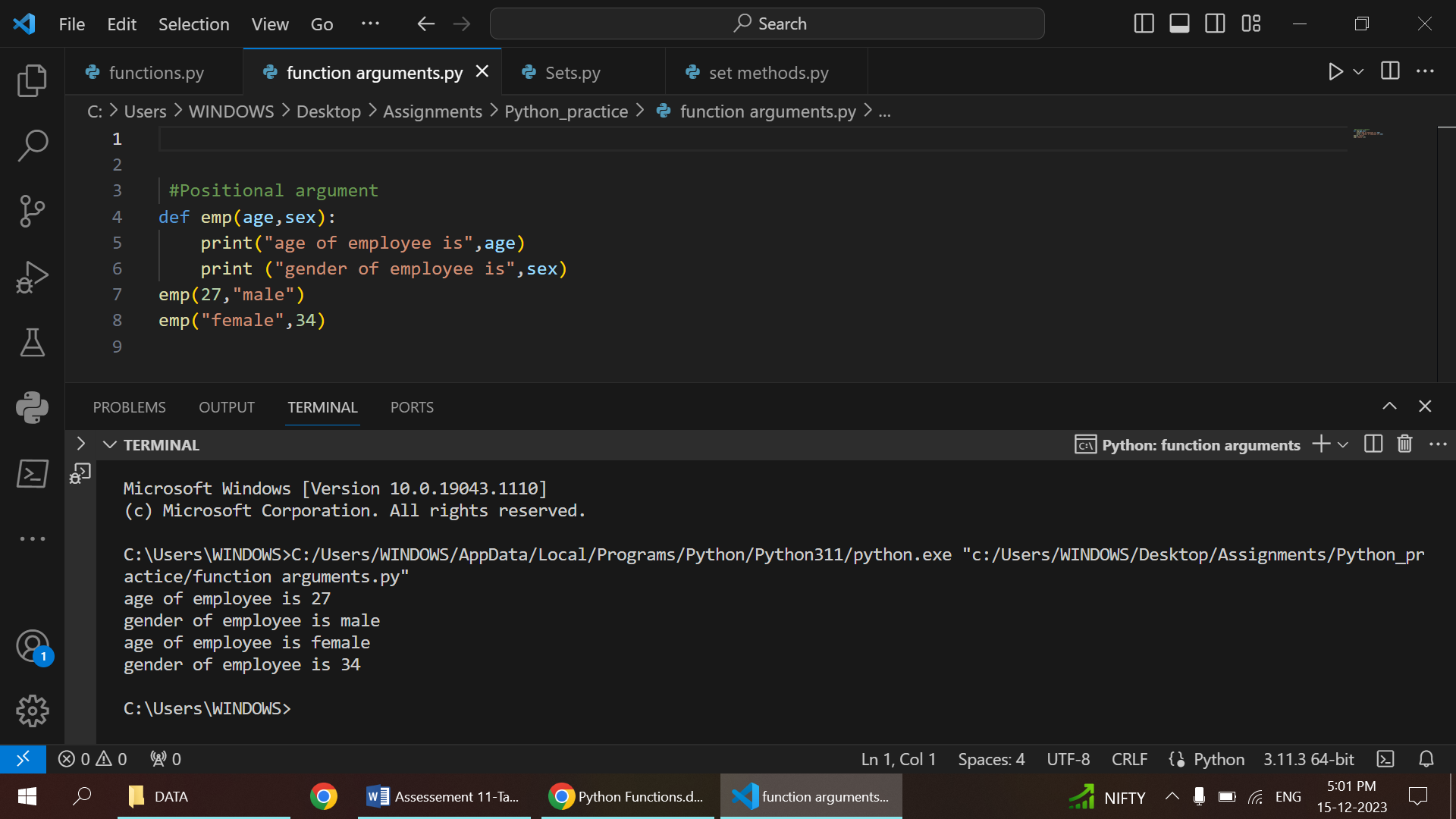
2.KEYWORD ARGUMENTS

* Keyword arguments in Python allow explicit specification of parameter values during function calls.
* Syntax involves assigning values to parameters by name
* Unlike positional arguments, the order in which keyword arguments are provided does not matter.
* Enhances code readability, especially in functions with multiple parameters, making the code more self-documenting.
* Keyword arguments can be used alongside positional arguments in a function call.



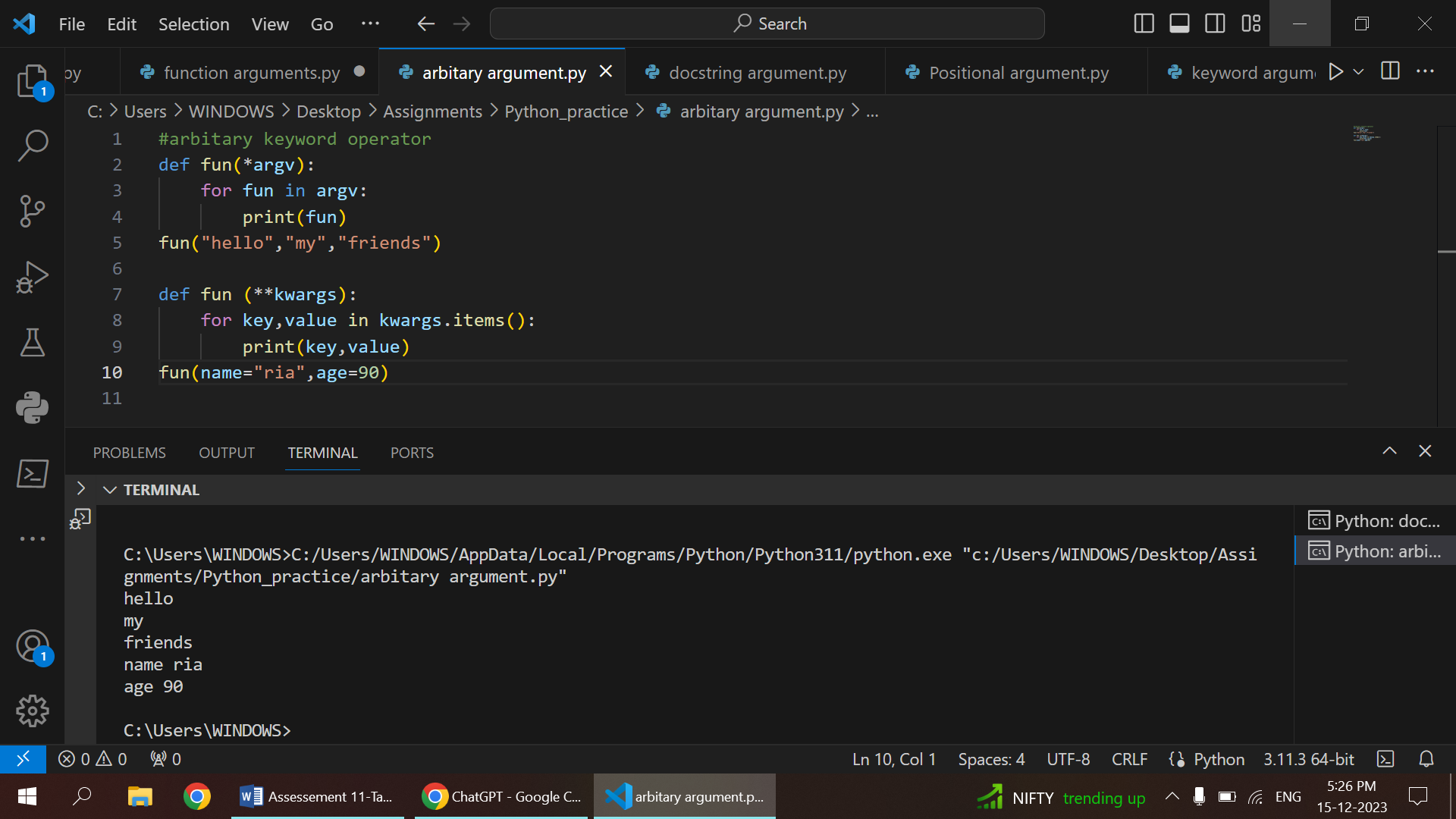
3.POSITIONAL ARGUMENTS

* Values passed to parameters based on their position in the function call are termed positional arguments.
* Syntax involves providing values in the order expected by the function
* Order matters: Positional arguments must align with the order of parameters in the function definition.
* Lack of explicit naming: Values are assigned based on their position; the function relies on the order of input.
* Most common method: Positional arguments are the default method of passing values to parameters in Python functions.



4.ARBITARY KEYWORD ARGUMENTS

* Allow passing a variable number of keyword arguments to a Python function.
* Defined using the \*\*kwargs syntax in the function parameter list.
* Collects excess keyword arguments into a dictionary.
* Provides flexibility when dealing with functions with an unspecified number of named parameters.
* Useful when the number of named parameters is not known beforehand.



5.DOCSTRING ARGUMENTS

* Multi-line strings below a function definition serve as docstring arguments, documenting parameter details.
* Syntax involves triple double or single quotes immediately following the function definition.
* Docstrings clarify parameter types and expected behavior, enhancing code understanding.
* Accessible via the .\_\_doc\_\_ attribute, docstrings contribute to code documentation.
* Writing concise and clear docstrings is a best practice for code readability.

