Table of Contents

## 1 FunctionDef num(a, b)

The function of **num** is to calculate and return the sum of two input values.

**parameters**: The parameters of this Function. · a: The first value to be used in the addition. · b: The second value to be used in the addition.

**Code Description**: The function num is defined to accept two parameters, a and b. It then calculates the sum of a and b using the + operator and returns the resulting value.

**Note**: This function will work with any data types that support the addition (+) operator, such as integers, floats, or strings (for concatenation). If incompatible types are provided, a TypeError will be raised.

**Output Example**:

5

## 2 FunctionDef generate\_random\_integers(count, start, end)

The function of **generate\_random\_integers** is to return a list of a specified number of pseudo-random integers within a given inclusive range.

**parameters**: The parameters of this Function. · count: An integer specifying the number of random integers to generate. · start: An optional integer representing the inclusive lower bound for the generated values, with a default of 0. · end: An optional integer representing the inclusive upper bound for the generated values, with a default of 100.

**Code Description**: The function first validates the count parameter. If count is a negative number, it raises a ValueError. Next, it checks if the start value is greater than the end value. If it is, the function swaps the two values to ensure the range is valid. Finally, it uses a list comprehension to generate a list of integers. This comprehension iterates count times, and in each iteration, it calls random.randint(start, end) to produce a single random integer within the inclusive range of start and end. The resulting list of random integers is then returned.

**Note**: This function depends on the random module, which must be imported for it to work. If the start value is provided as greater than the end value, the function automatically corrects the range by swapping them instead of raising an error.

**Output Example**: Calling generate\_random\_integers(5, 1, 10) might return a list similar to [3, 10, 1, 7, 5]. ## FunctionDef choose\_random\_item(items) The function of **choose\_random\_item** is to select and return a single random string from a provided list of strings.

**parameters**: The parameters of this Function. · items: A list of strings from which a random item will be chosen.

**Code Description**: The function first evaluates if the input list items is empty. If the list has no elements, it raises a ValueError with the message “items must not be empty” to prevent errors in the subsequent step. If the list is not empty, the function proceeds to use the random.choice() method to select a single item uniformly at random from the items list and returns that chosen string.

**Note**: This function requires the random module to be imported. It will raise a ValueError if an empty list is provided as an argument.

**Output Example**: If items is ['apple', 'banana', 'cherry'], a possible return value is 'banana'. ## FunctionDef shuffle\_copy(items) The function of **shuffle\_copy** is to return a new, randomly shuffled copy of a given list of integers without modifying the original list.

**parameters**: The parameters of this Function. · items: A list of integers that will be copied and shuffled.

**Code Description**: The function first creates a shallow copy of the input list items and assigns it to a new variable named copy. It then uses the random.shuffle() method to rearrange the elements of the copy list into a random order. This shuffling operation is performed in-place on the copy, leaving the original items list unchanged. Finally, the function returns the shuffled copy.

**Note**: This function requires the random module to be imported to work correctly. It is designed to be non-mutating, meaning the list passed as an argument will not be altered.

**Output Example**: If you call shuffle\_copy([1, 2, 3, 4, 5]), a possible return value could be [3, 1, 5, 2, 4]. The exact order will be different each time the function is called.