def function in python

The def function in Python is not a specific function but rather a keyword used to define functions. When you use the def keyword, you're creating a new function that can perform specific tasks and be reused throughout your code.

What Does def Do?

The def keyword tells Python that you are defining a new function. It is followed by:

1)The name of the function.

2) Parentheses (), which may contain parameters (inputs).

3) A colon :, which marks the start of the function body.

4) An indented block of code that specifies what the function does.

1) find the maximum of from given

In [1]:
```
numbers=[23,55,78,98,90,60]
def find_max_from_list(numbers):
    if not numbers:
        return None  # Handle empty list
    return max(numbers)
print(find_max_from_list(numbers))
```

```
98
```

2) sum all the numbers in a list.

Sample List : [8, 2, 3, 0, 7]

In [2]:
```
sample_list = [8, 2, 3, 0, 7]
def sum_of_list(numbers):
    return sum(numbers)
print(sum_of_list(sample_list))
```

```
20
```

3) takes a list and returns a new list with distinct elements from the first list.

Sample List : [1,2,3,3,3,3,4,5]

In [14]:
```
sample_list = [1, 2, 3, 3, 3, 3, 4, 5]
def get_distinct_elements(sample_list):
    return list(set(sample_list))
print(get_distinct_elements(sample_list))
```

```
[1, 2, 3, 4, 5]
```

4) total number of Combinations.

In [21]:
```
from itertools import combinations
elements = ['x', 'y', 'z']
count, comb_list = total_combinations(elements)

def total_combinations(elements):
    comb = list(combinations(elements, 2))  # Generate all combinations of size
    count = len(comb)  # Count the combinations
    return count, [''.join(pair) for pair in comb]  # Return count and formattea

print(f"Total combinations: {count}")
print(f"Combinations: {comb_list}")
```

```
Total combinations: 3
Combinations: ['xy', 'xz', 'yz']
```

5) total number of permutation

In [24]:
```
from itertools import permutations
elements = ['x', 'y', 'z']
count, perm_list = total_permutations(elements)

def total_permutations(elements):
    perm = list(permutations(elements, 2))  # Generate all permutations of size
    count = len(perm)  # Count the permutations
    return count, [''.join(pair) for pair in perm]  # Return count and formattea

elements = ['x', 'y', 'z']
count, perm_list = total_permutations(elements)

print(f"Total permutations: {count}")
print(f"Permutations: {perm_list}")
```

```
Total permutations: 6
Permutations: ['xy', 'xz', 'yx', 'yz', 'zx', 'zy']
```

6) counts vowels and consonant in a word.

input : statistics

In [23]:
```
word = "statistics"
vowels, consonants = count_vowels_consonants(word)

def count_vowels_consonants(word):
    vowels = "aeiouAEIOU"
    vowel_count = 0
    consonant_count = 0

    for char in word:
        if char.isalpha():  # Check if the character is a letter
            if char in vowels:
                vowel_count += 1
            else:
                consonant_count += 1

    return vowel_count, consonant_count

print(f"Vowels: {vowels}")
print(f"Consonants: {consonants}")
```

```
Vowels: 3
Consonants: 7
```

7) lowercase words and returns uppercase words.

Input:= STATISTICS

In [9]:
```python
def convert_to_uppercase(word):
    return word.upper()

# Call the function after defining it
word = "statistics"
result = convert_to_uppercase(word)

print(f"Uppercase: {result}")
```

```
Uppercase: STATISTICS
```

8) count lower case and upper case letter.

Ex : STatiStiCS

In [7]:
```python
def count_case_letters(word):
    lowercase_count = sum(1 for char in word if char.islower())
    uppercase_count = sum(1 for char in word if char.isupper())
    return lowercase_count, uppercase_count

# Now call the function
word = "STatiStiCS"
lowercase, uppercase = count_case_letters(word)

print(f"Lowercase: {lowercase}")
print(f"Uppercase: {uppercase}")
```

```
Lowercase: 5
Uppercase: 5
```

In [ ]:
```python

```