

Assignment Week – 1

Data Types

Objective: Convert between different data types.

1. **Task:** Convert the following values to the specified types and print the results

1. Convert **3.75** to an integer and print the value

1. `print(int(3.75)) = 3`

2. Convert **"123"** to a float and print the value

2. `print(float("123")) = 123.0`

3. Convert **0** to a boolean and print the value

3. `print(bool(0)) = False`

4. Convert **False** to a string and print the value

4. `print(str(False)) = False`

2. Convert all characters in the string to uppercase. `x = "hello"`

2. `x = "hello"`

`print(x.upper())`

`HELLO`

3. Given **x = 5** and **y = 3.14**, calculate **z = x + y** and determine the data type of **z**. And convert it to integer.

3. `x = 5`

`y = 3.14`

`z = x + y`

`print(z) // 8.14`

`print(type(z)) // <class 'float'>`

`print(int(z)) //8`

4. Given the string `s = 'hello'`, perform the following operations:

- Convert the string to uppercase.
- Replace 'e' with 'a'.
- Check if the string starts with 'he'.
- Check if the string ends with 'lo'.

4. `s = 'hello'`

```
print(s.upper())      // HELLO
```

```
print(s.replace('e', 'a')) //hallo
```

```
print(s.startswith('he')) //False
```

```
print(s.endswith('lo')) //True
```

Input & Print

Objective: Ask the user for their name and greet them.

Task: Write a program that asks the user for their name and then prints a greeting message using their name.

2. **Objective:** Perform basic arithmetic operations based on user input.

Task: Ask the user to enter two numbers from the user and print their sum, multiplication, and division.

3. **Task:** Ask the user to enter input names separated by commas, split the string from comma and copy to a list and print.

4. **Task:** Ask the user to enter their age and check if they are eligible to vote based on their age.

5. For value = 3.14159, Using f-string print output for only up to 2 decimal places. Output: 3.14

Solutions :

1.)

```
name = input("Enter your name: ") // Enter your name: Nitin  
print("Hello,", name) // Hello, Nitin
```

2.)

```
num1 = float(input("Enter first number: ")) // Enter first number: 10  
num2 = float(input("Enter second number: ")) //Enter second number: 5
```

```
print(num1 + num2) // 15.0  
print(num1 * num2) //50.0  
print(num1 / num2) //2.0
```

3.)

```
names = input("Enter names separated by commas: ") //Enter names separated by commas:  
Nitin,Rahul,Amit  
name_list = names.split(",")  
print(name_list) // ['Nitin', 'Rahul', 'Amit']
```

4.)

```
age = int(input("Enter your age: "))  
if age >= 18:  
    print("Eligible to vote")  
else:  
    print("Not eligible to vote")
```

5.)

```
value = 3.14159  
print(f"{value:.2f}")
```

Built In – Data Structure

Given a list of numbers, find and print the maximum and minimum values.

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

2. Given two lists below, merge the values from both lists to one and print

```
a = [1,2,3,4]      b = [5,6,7,8]
```

3. From a list, print the number of times the value 3 appears in the list:

```
a = [1,3,4,5,2,1,3,9,3]
```

4. From below list, Sort the list and print

```
a = [1,3,4,5,2,1,3,9,3]
```

5. Given a set, add the element 6 to it and print the updated set.

```
numbers = {1, 2, 3, 4, 5}
```

6. Given a set, remove the element 3 from it and print the updated set.

```
numbers = {1, 2, 3, 4, 5}
```

7. Given two sets, find and print their intersection.

```
set1 = {1, 2, 3}  set2 = {3, 4, 5}
```

8. Given a tuple, count and print the number of occurrences of the element 'apple'.

```
fruits = ('apple', 'banana', 'apple', 'cherry')
```

9. Given two tuples, concatenate them and print the result.

```
tuple1 = (1, 2, 3) tuple2 = (4, 5, 6)
```

10. Access and print the value associated with the key "age" from the dictionary.

```
person = {"name": "Alice", "age": 30, "city": "New York"}
```

11. Add new key, gender to dictionary and assign "M" to it and print

```
person = {"name": "Alice", "age": 30, "city": "New York"}
```

12. Remove the key "city" from the above Dict and print

13. Given two dictionaries, merge them into one

```
dict1 = {"a": 1, "b": 2} dict2 = {"c": 3, "d": 4}
```

Solutions :

1.)

```
nums = [1, 2, 3, 4, 5]  
print(max(nums))  
print(min(nums))
```

2.)

```
a = [1, 2, 3, 4]  
b = [5, 6, 7, 8]
```

```
merged_list = a + b  
print(merged_list)
```

3.)

```
a = [1, 3, 4, 5, 2, 1, 3, 9, 3]  
print(a.count(3)) //3
```

4.)

```
a = [1, 3, 4, 5, 2, 1, 3, 9, 3]
a.sort()
print(a) // [1, 1, 2, 3, 3, 3, 4, 5, 9]
```

5.)

```
numbers = {1, 2, 3, 4, 5}
numbers.add(6)
print(numbers) // {1, 2, 3, 4, 5, 6}
```

6.)

```
numbers = {1, 2, 3, 4, 5}
numbers.remove(3)
print(numbers) // {1, 2, 4, 5}
```

7.)

```
set1 = {1, 2, 3}
set2 = {3, 4, 5}

print(set1.intersection(set2)) // {3}
```

8.)

```
fruits = ('apple', 'banana', 'apple', 'cherry')
print(fruits.count('apple')) // 2
```

9.)

```
tuple1 = (1, 2, 3)
tuple2 = (4, 5, 6)

print(tuple1 + tuple2) //(1, 2, 3, 4, 5, 6)
```

10.)

```
person = {"name": "Alice", "age": 30, "city": "New York"}  
print(person["age"]) // 30
```

11.)

```
person = {"name": "Alice", "age": 30, "city": "New York"}  
person["gender"] = "M"  
print(person) //{'name': 'Alice', 'age': 30, 'city': 'New York', 'gender': 'M'}
```

12.)

```
person = {"name": "Alice", "age": 30, "city": "New York"}  
person.pop("city")  
print(person) //{'name': 'Alice', 'age': 30}
```

13.)

```
dict1 = {"a": 1, "b": 2}  
dict2 = {"c": 3, "d": 4}  
  
merged_dict = {**dict1, **dict2}  
print(merged_dict) //{'a': 1, 'b': 2, 'c': 3, 'd': 4}
```

Controls

For loop

1. Write a program that takes the input from the user and checks if a number is even or odd.
2. Reverse a string using a for loop and check it is a palindrome. - Strings = “civic”, “hello”
3. Using the input from the user, Generate the first **N** numbers of the Fibonacci sequence.
4. From list [1,2,3,4,5]. Write code to find two values from the list when added the result is 9.
#Expected output : [4, 5]

While loop

5. Print all even numbers between 1 and 20 using a **while** loop.

Break

6. Find the first occurrence of a number in a list and stop further searching.

```
numbers = [10, 20, 30, 40, 50]  
search_for = 30
```

Continue

7. Using continue statement, print only the odd numbers from 1 to 10.

Pass

8. What will be the output

```
for i in range(5):  
    if i == 3:  
        pass  
    print(i)
```

Match

9. Write a program that takes a day of the week as input and prints whether it's a weekday or weekend using match conditional statements.

Solutions :

1.)

```
num = int(input("Enter a number: "))  
if num % 2 == 0:  
    print("Even")  
else:  
    print("Odd")
```

2.)

```
s = input("Enter a string: ")
```

```
rev = ""
```

```
for char in s:
```

```
    rev = char + rev
```

```
print(rev)
```

```
if s == rev:
```

```
print("Palindrome")
else:
    print("Not Palindrome")
```

3.)

```
n = int(input("Enter N: "))
a, b = 0, 1
for i in range(n):
    print(a, end=" ")
    a, b = b, a + b
```

4.)

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

for i in range(len(lst)):
    for j in range(i + 1, len(lst)):
        if lst[i] + lst[j] == 9:
            print([lst[i], lst[j]])
```

5.)

```
i = 1
while i <= 20:
    if i % 2 == 0:
        print(i)
    i += 1
```

6.)

```
numbers = [10, 20, 30, 40, 50]
search_for = 30
for num in numbers:
    if num == search_for:
        print(num)
```

```
break
```

7.)

```
for i in range(1, 11):
```

```
    if i % 2 == 0:
```

```
        continue
```

```
    print(i)
```

8.)

```
for i in range(5):
```

```
    if i == 3:
```

```
        pass
```

```
    print(i)
```

9.)

```
day = input("Enter day: ").lower()
```

```
match day:
```

```
    case "saturday" | "sunday":
```

```
        print("Weekend")
```

```
    case "monday" | "tuesday" | "wednesday" | "thursday" | "friday":
```

```
        print("Weekday")
```

```
    case _:
```

```
        print("Invalid day")
```

Functions

1.) Define a function `calculate_area` that calculates the area of a rectangle and return the result. If no width is provided, it defaults to 10.

2. Write a recursive function to compute the factorial of a non-negative integer.

3. Write a function that takes one parameter as a string and reverse it and return.

4. Write a Python function that takes two parameters as lists and to sum all the numbers in a list.

a = [8, 2, 3, 0, 7], b = [3, -2, 5, 1] and return a value.

5. Write a Python function that takes a list and returns a new list with distinct and sorted elements from the first list. a = [4,1,2,3,3,1,3,4,5,1,7]

Output = [1,2,3,4,5,7]

Solutions :

1.)

```
def calculate_area(length, width=10):  
    return length * width
```

```
print(calculate_area(5)) //50
```

```
print(calculate_area(5, 4)) //20
```

2.)

```
def factorial(n):  
    if n == 0:  
        return 1  
    return n * factorial(n - 1) //120
```

```
print(factorial(5))
```

3.)

```
def reverse_string(s):  
    return s[::-1]  
  
print(reverse_string("hello")) //olleh
```

4.)

```
def sum_lists(a, b):  
    return sum(a) + sum(b)
```

```
a = [8, 2, 3, 0, 7]
b = [3, -2, 5, 1]
print(sum_lists(a, b)) //27
5.)
def distinct_sorted(lst):
    return sorted(set(lst))
```

```
a = [4,1,2,3,3,1,3,4,5,1,7]
print(distinct_sorted(a)) // [1, 2, 3, 4, 5, 7]
```

List Comprehension

- Given a list of numeric strings, convert them into integers. Using List Comprehensions

```
strings = ["1", "2", "3", "4", "5"]
```

```
#Expected output : [1, 2, 3, 4, 5]
```

- Extract all integers from a list that are greater than 10. Using List Comprehensions

```
numbers = [1, 5, 13, 4, 16, 7]
```

```
#Expected output : [13, 16]
```

- Create a list of squares for numbers from 1 to 5. Using List Comprehensions

```
#Expected output : [1, 4, 9, 16, 25]
```

- Convert a 2D list into a 1D list.Using List Comprehensions

```
matrix = [[1, 3, 4], [23, 32, 56, 74], [-2, -6, -9]]
```

```
#Expected output : [1, 3, 4, 23, 32, 56, 74, -2, -6, -9]
```

- Given two lists, **keys** = ['a', 'b', 'c'] and **values** = [1, 2, 3], create a dictionary using dictionary comprehension.

```
#Expected output : {'a': 1, 'b': 2, 'c': 3}
```

6. Given the dictionary `scores = {'Alice': 85, 'Bob': 70, 'Charlie': 90}`, create a new dictionary containing only the students who scored above 80

#Expected output : {'Alice': 85, 'Charlie': 90}

Solutions :

1.)

```
strings = ["1", "2", "3", "4", "5"]
result = [int(x) for x in strings]
print(result)
```

2.)

```
numbers = [1, 5, 13, 4, 16, 7]
result = [x for x in numbers if x > 10]
print(result)
```

3.)

```
result = [x**2 for x in range(1, 6)]
print(result)
```

4.)

```
matrix = [[1, 3, 4], [23, 32, 56, 74], [-2, -6, -9]]
result = [num for row in matrix for num in row]
print(result)
```

5.)

```
keys = ['a', 'b', 'c']
values = [1, 2, 3]
result = {keys[i]: values[i] for i in range(len(keys))}
print(result)
```

6.)

```
scores = {'Alice': 85, 'Bob': 70, 'Charlie': 90}
result = {name: score for name, score in scores.items() if score > 80}
print(result)
```