

# Assignment-1

## EXERCISE 1 :

### ERROR:

The error in the code is a variable name typo.

The variable is defined as "number\_of\_apples" but is defined as "number\_of\_apple" in the print statement.

### CORRECTED CODE:

```
number_of_apples = 5  
print(number_of_apples)
```

### ERROR EXPLANATION:

- When you define a variable, you must use the exact same spelling and capitalization when you use it later.
- In this code, "number\_of\_apples" is defined with an "s" at the end, but when printing, it's defined without the "s".

To fix the error, simply add the missing "s" to the variable name in the print statement.

-----

## EXERCISE 2 :

### ERROR:

The error in the code is an IndexError.

The error occurs when trying to access an element in a list using an index that is out of range.

### CORRECTED CODE:

```
fruits = ["apples", "banana", "cherry"]  
print(fruits[2])
```

### ERROR EXPLANATION:

- List indices start at 0, so the third element is at index 2.
- Since the list has only 3 elements, the maximum index is 2.

To avoid this error, ensure that the index you're using is within the range of the list's length. In this case, the list has 3 elements, so the valid indices are 0, 1, and 2.

-----

### **EXERCISE 3 :**

#### **ERROR:**

The error in the code is an `TypeError`.

The issue in the code is that the list "numbers" contains a string ("6") which cannot be added to the sum.

#### **CORRECTED CODE:**

```
def find_average(numbers):  
    sum = 0  
    for number in numbers:  
        if isinstance(number, int) or isinstance(number, float): # Check if the element is a number  
            sum += number  
    average = sum / len(numbers)  
    return average
```

```
numbers = [1, 2, 3, 4, 5, 6] # Removed the quotes around "6"  
average = find_average(numbers)  
print(f"The average is: {average}")
```

#### **ERROR EXPLANATION:**

- Added a check to ensure that only numbers (int or float) are added to the sum.
- Removed the quotes around "6" in the list, so it's now an integer instead of a string.

With these changes, the function should work as expected and calculate the average of the numbers in the list.

-----

## EXERCISE 4 :

### ERROR:

The error in the code is the dictionary values are lists.

Additionally, the dictionary key is case-sensitive, so "Alice" and "alice" are treated as different keys.

### CORRECTED CODE:

```
def update_record(records, name, score):  
    name = name.lower() # Convert name to lowercase for consistency  
    if name in records:  
        records[name].append(score)  
    else:  
        records[name] = [score]  
  
student_records = {"alice": [88, 92], "bob": [70, 85]}  
update_record(student_records, "charlie", 91)  
update_record(student_records, "alice", 95) # Now treated as "alice"  
print(student_records)
```

### ERROR EXPLANATION:

- Added name = name.lower() to convert the name to lowercase, ensuring consistency in key names.
- Initialized the new key with a list [score] instead of just score.

Now, the code should update the records correctly and

```
print: {'alice': [88, 92, 95], 'bob': [70, 85], 'charlie': [91]}
```

-----