

```
# Write a function to count the number of elements of each data type in the list.
```

```
def count_data_types(lst):
    type_counts = {}

    for element in lst:
        element_type = type(element)
        if element_type in type_counts:
            type_counts[element_type] += 1
        else:
            type_counts[element_type] = 1

    return type_counts
```

```
heterogeneous_list = [42, 46, "Hello", 3.14, True, None, [1, 2, 3], {"key": "value"}, (4, 5), {9, 8, 7}]
result = count_data_types(heterogeneous_list)
print(result)
```

```
# Create a function that takes a data type as an argument and returns a new list containing only elements of that type from the original list.
```

```
def filter_by_type(lst, data_type):
    return [element for element in lst if isinstance(element, data_type)]
```

```
heterogeneous_list = [42, "Hello", 3.14, True, None, [1, 2, 3], {"key": "value"}, (4, 5), {9, 8, 7}]
filtered_list = filter_by_type(heterogeneous_list, int)
print(filtered_list) # Output: [42]
```

```
filtered_strings = filter_by_type(heterogeneous_list, str)
print(filtered_strings) # Output: ["Hello"]
```

```
filtered_floats = filter_by_type(heterogeneous_list, float)
print(filtered_floats) # Output: [3.14]
```

```
filtered_lists = filter_by_type(heterogeneous_list, list)
```

```

print(filtered_lists) # Output: [[1, 2, 3]]

filtered_lists = filter_by_type(heterogeneous_list, tuple)

print(filtered_lists)

```

```

# Write a function to sum all numerical values (integers and floats) in the list.
def sum_numerical_values(input_list):
    total = 0
    for item in input_list:
        # Check if the type of the item is either int or float
        if type(item) == int or type(item) == float:
            total += item
    return total

# Test the function with the given list
list1 = [2, 4, 5.4, 6.5, 7, 3, "str1"]
LIST1=[1,1,1.1]
heterogeneous_list = [42, "Hello", 3.14, True, None, [1, 2, 3], {"key": "value"}, (4, 5),
{9, 8, 7}]
print(sum_numerical_values(list1))
print(sum_numerical_values(heterogeneous_list))

```

```

# Write a function to concatenate all string elements in the list into a single string.

heterogeneous_list = [42, "Hello", 3.14, True, None, [1, 2, 3], {"key": "value"}, (4, 5),
{9, 8, 7}]

def concatenate_strings(input_list):
    result = ""
    for item in input_list:
        # Check if the item is a string

```

```

    # we can also use isinstance()
    if type(item) == str:
        result += item
    return result

```

Test the function with the given list

```

heterogeneous_list = [42, "Hello", "world", 3.14, True, None, [1, 2, 3], {"key": "value"},
(4, 5), {9, 8, 7}]
print(concatenate_strings(heterogeneous_list))

```

Write a function to flatten the list, where nested lists, tuples, and sets are expanded into individual elements in a single list.

```

def flatten_list(heterogeneous_list):
    flattened_list = []
    for item in heterogeneous_list:
        if isinstance(item, (list, tuple, set)):
            flattened_list.extend(flatten_list(item))
        else:
            flattened_list.append(item)
    return flattened_list

```

Test the function with the given heterogeneous list

```

heterogeneous_list = [42, "Hello", "world", 3.14, True, None, [1, 2, 3], {"key":
"value"}, (4, 5), {9, 8, 7}]
print(flatten_list(heterogeneous_list))

```

Write a function to check if a specific key exists in any dictionary elements within the list.

```

def key_exists(heterogeneous_list, key):
    for item in heterogeneous_list:
        if type(item) == dict:
            for KEY in item.keys():
                if KEY == key:
                    return True
    return False

```

```

heterogeneous_list = [42, "Hello", "world", 3.14, True, None, [1, 2, 3], {"key":
"value"}, (4, 5), {9, 8, 7}]

```

```

print(key_exists(heterogeneous_list,"key"))
# Write a function to count the number of True and False boolean values in the list

def boolean_values(heterogeneous_list):
    tc=0
    fc=0
    for item in heterogeneous_list:
        if item==True:
            tc+=item
        elif item ==False:

            fc += item
    return tc,fc
heterogeneous_list = [42, "Hello", "world", 3.14, True, None, [1, 2, 3], {"key":
"value"}, (4, 5), {9, 8, 7}]
tc1,fc1=boolean_values(heterogeneous_list)
print("Number of True count",tc1)
print("number of false count",fc1)

```

Write a function to convert all elements in the list to their string representations and return a new list of these strings.

```

def convert_to_strings(input_list):
    return [str(item) for item in input_list]

# Test the function with a list containing various types of elements
heterogeneous_list = [42, "Hello", "world", 3.14, True, None, [1, 2, 3], {"key":
"value"}, (4, 5), {9, 8, 7}]
string_list = convert_to_strings(heterogeneous_list)
print(string_list)

```

Write a function to find the maximum value among all numeric elements in the list.

```

def find_max_numeric(lst):

    # numeric_elements = [x for x in lst if isinstance(x, (int, float))]
    numeric_elements = []
    for x in lst:
        if isinstance(x, (int, float)):
            numeric_elements.append(x)

```

```
    return max(numeric_elements, default=None)
```

```
heterogeneous_list = [42, "Hello", "world", 3.14, True, None, [1, 2, 3], {"key":  
"value"}, (4, 5), {9, 8, 7}]
```

```
result = find_max_numeric(heterogeneous_list)
```

```
print(result) # Output: 4
```

```
# Write a function to replace all None values in the list with a default value provided  
as an argument.
```

```
def replace_none_with_default(lst, default_value):
```

```
    for i in range(len(lst)):
```

```
        if lst[i] is None:
```

```
            lst[i] = default_value
```

```
    return lst
```

```
heterogeneous_list = [42, "Hello", "world", 3.14, True, None, [1, 2, 3], {"key":  
"value"}, (4, 5), {9, 8, 7}]
```

```
default_value = 6
```

```
result = replace_none_with_default(heterogeneous_list, default_value)
```

```
print(result)
```