

1. Show that a List in Python can:

```
python
Copy code
# Store elements of different types
my_list = [1, 3.14, "Hello", True]
print("List with different types:", my_list)

# Store duplicate elements
duplicate_list = [1, 2, 2, 3, 3, 3]
print("List with duplicates:", duplicate_list)
```

2. Check if an Element Exists in a List

```
python
Copy code
my_list = [1, 3.14, "Hello", True]
element = "Hello"
print(f"Is '{element}' in the list? {'Yes' if element in my_list else 'No'}")
```

3. Find the Length of the List

```
python
Copy code
my_list = [1, 3.14, "Hello", True]
print("Length of the list:", len(my_list))
```

4. Create a List with Values n^2 for n from 1 to 5

```
python
Copy code
squares = [n ** 2 for n in range(1, 6)]
print("List of squares:", squares)
```

5. Print the Elements of an Array in Reverse Order

```
python
Copy code
array = [10, 20, 30, 40]
print("Array in reverse order:", array[::-1])
```

7. Sort the Elements of an Array in Descending Order

```
python
Copy code
array = [10, 20, 30, 40]
print("Array sorted in descending order:", sorted(array, reverse=True))
```

8. Compare 2 Lists

```
python
Copy code
list1 = [1, 2, 3]
list2 = [1, 2, 3]

# Using cmp() function (not available in Python 3, use manual comparison)
cmp_result = (list1 > list2) - (list1 < list2) # Equivalent to cmp
print("Comparison result using cmp:", cmp_result)
```

```
# Using set() function and ==
print("Are lists equal using set:", set(list1) == set(list2))

# Using sort() and ==
print("Are lists equal using sorted:", sorted(list1) == sorted(list2))
```

9. Remove a Specific Item from a List

```
python
Copy code
my_list = [10, 20, 30, 40]

# Using remove()
my_list.remove(20)
print("List after remove(20):", my_list)

# Using pop()
my_list.pop(1) # Removes element at index 1
print("List after pop(1):", my_list)

# Using clear()
my_list.clear()
print("List after clear():", my_list)
```

10. Convert an Integer Array to Floating Type (Using NumPy)

```
python
Copy code
import numpy as np
a = [1, 2, 3]
x = np.asfarray(a)
print("Floating type array:", x)
```

11. Convert a List and Tuple into Arrays (Using NumPy)

```
python
Copy code
import numpy as np
a_list = [1, 2, 3]
a_tuple = (4, 5, 6)

list_array = np.asarray(a_list)
tuple_array = np.asarray(a_tuple)

print("Array from list:", list_array)
print("Array from tuple:", tuple_array)
```