**Today Agenda**

**List in Kotlin**

1. Creating a List

2. Accessing Elements

3. Adding Elements

4. Removing Elements

5. Checking if an Element Exists

6. Sorting the List

7. Mapping Elements

8. Filtering Elements

9. Finding Elements

10. Checking if the List is Empty

11. Counting Elements

12. Iterating Over the List

I'll **explain each step** of the code and its function:

1. Creating a List:

- `val myList = listOf(1, 2, 3, 4, 5)` creates an immutable list named `myList` containing five integers (1, 2, 3, 4, 5). The `listOf` function is used to create a read-only list.

2. Accessing Elements:

- `val firstElement = myList[0]` accesses the first element (index 0) of the list and assigns it to the variable `firstElement`.

- `val lastElement = myList.last()` retrieves the last element of the list using the `last()` function.

3. Adding Elements:

- `val newList = myList + 6` creates a new list `newList` by adding the integer 6 to the original list. This operation doesn't modify the original list; it creates a new one.

4. Removing Elements:

- `val filteredList = myList.filter { it != 3 }` creates a new list `filteredList` by filtering out elements that are equal to 3 from the original list. It uses the `filter` function with a lambda expression.

5. Checking if an Element Exists:

- `val containsElement = myList.contains(2)` checks if the element 2 exists in the list and assigns the result to `containsElement`.

6. Sorting the List:

- `val sortedList = myList.sorted()` creates a new list `sortedList` that contains the elements from the original list, sorted in ascending order.

7. Mapping Elements:

- `val squaredList = myList.map { it \* it }` creates a new list `squaredList` where each element is the square of the corresponding element in the original list. It uses the `map` function to apply the provided lambda to each element.

8. Filtering Elements:

- `val evenNumbers = myList.filter { it % 2 == 0 }` creates a new list `evenNumbers` containing only the even numbers from the original list. It uses the `filter` function with a lambda expression.

9. Finding Elements:

- `val firstEven = myList.find { it % 2 == 0 }` searches for the first element in the list that satisfies the given condition (even number) and assigns it to `firstEven`.

10. Checking if the List is Empty:

- `val isEmpty = myList.isEmpty()` checks if the list `myList` is empty and assigns the result to `isEmpty`.

11. Counting Elements:

- `val count = myList.count()` counts the number of elements in the list and assigns it to the variable `count`.

12. Iterating Over the List:

- `myList.forEach { println(it) }` iterates through each element in the list and prints it using the `forEach` function with a lambda expression.

These code examples demonstrate various common operations you can perform on lists in Kotlin, including creation, modification, filtering, and iteration. Each operation serves a different purpose and can be useful in different scenarios when working with lists of data.

Certainly! Here's the full code for each list operation in Kotlin:

1. Creating a List:

fun main() {

val myList = listOf(1, 2, 3, 4, 5)

println(myList)

}

2. Accessing Elements:

fun main() {

val myList = listOf(1, 2, 3, 4, 5)

val firstElement = myList[0]

val lastElement = myList.last()

println("First: $firstElement, Last: $lastElement")

}

3. Adding Elements:

fun main() {

val myList = listOf(1, 2, 3, 4, 5)

val newList = myList + 6

println(newList)

}

4. Removing Elements:

fun main() {

val myList = listOf(1, 2, 3, 4, 5)

val filteredList = myList.filter { it != 3 }

println(filteredList)

}

5. Checking if an Element Exists:

fun main() {

val myList = listOf(1, 2, 3, 4, 5)

val containsElement = myList.contains(2)

println("Contains 2: $containsElement")

}

6. Sorting the List:

fun main() {

val myList = listOf(5, 2, 4, 1, 3)

val sortedList = myList.sorted()

println(sortedList)

}

7. Mapping Elements:

fun main() {

val myList = listOf(1, 2, 3, 4, 5)

val squaredList = myList.map { it \* it }

println(squaredList)

}

8. Filtering Elements:

fun main() {

val myList = listOf(1, 2, 3, 4, 5)

val evenNumbers = myList.filter { it % 2 == 0 }

println(evenNumbers)

}

9. Finding Elements:

fun main() {

val myList = listOf(1, 2, 3, 4, 5)

val firstEven = myList.find { it % 2 == 0 }

println("First even number: $firstEven")

}

10. Checking if the List is Empty:

fun main() {

val myList = listOf<Int>()

val isEmpty = myList.isEmpty()

println("Is the list empty? $isEmpty")

}

11. Counting Elements:

fun main() {

val myList = listOf(1, 2, 3, 4, 5)

val count = myList.count()

println("Number of elements: $count")

}

12. Iterating Over the List:

fun main() {

val myList = listOf(1, 2, 3, 4, 5)

myList.forEach { println(it) }

}

.