LISTS AND OBJECTS

Multiline Comments

Syntax:

```
/*
This is a multiline comment
that spans multiple lines.
*/
```

ListOf and MutableListOf

Syntax:

```
val immutableList = listOf("item1", "item2")
val mutableList = mutableListOf("item1", "item2")
```

Function vs. Method

Function: Classes primarily used for holding data.

```
fun main() {
println("Hello, Kotlin!")
}
```

Method: Associated with objects, called on objects.

```
fun main() {
val text = "hello, kotlin!"
val uppercasedText = text.toUpperCase()
println(uppercasedText)
}
```

Understanding Indexing in Lists

Syntax:

```
val element = myList[index]
```

Examples:

Access: println(myList[1])
Add: mutableList.add("date")
Remove: mutableList.remove("banana")
Replace: mutableList[1] = "blueberry"

Using the set Method

Syntax:

```
mutableList.set(index, element)
```

Example:

```
val mutableList =
mutableListOf("apple", "banana",
"cherr
y")
mutableList.set(1, "blueberry")
```

Using the removeLast() Method

Syntax:

```
mutableCollection.removeLast()
```

Example:

```
val numbers = mutableListOf(1, 2, 3, 4, 5)
val removedNumber = numbers.removeLast()
```

Using the contains() Method

Syntax:

```
collection.contains(element)
```

Example:

```
val fruits = listOf("apple", "banana", "cherry")
val isBananaPresent = fruits.contains("banana")
```

What is a For Loop?

Basic For Loop:

```
for (fruit in listOf("apple", "banana", "cherry")) {
println(fruit)
}
```

Using Index in For Loop:

```
val fruits = listOf("apple", "banana", "cherry")
for ((index, fruit) in fruits.withIndex()) {
println("Fruit at position $index is $fruit")
}
```

Using Break in For Loop:

```
for (number in 1..10) {
println(number)
if (number == 5) break
}
```

Understanding private val

```
class SecretKeeper {
private val secretMessage: String = "The cake is a li
e!"
fun shareSecret(): String {
return "I can tell you that: $secretMessage"
}
}
```

Benefits:

- **Encapsulation:** Keeps internal state hidden and unmodifiable.
- **Safety:** Prevents unintentional modifications and access.