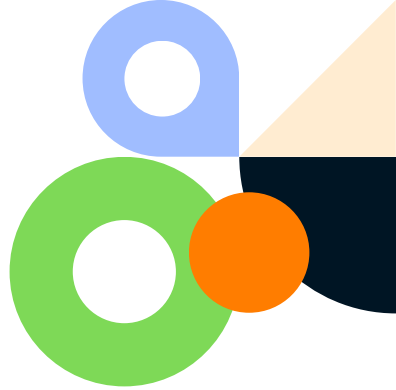
Java Streams - takeWhile



This method allow you to control stream processing based on a condition. ***takeWhile*** keeps processing elements until the predicate fails

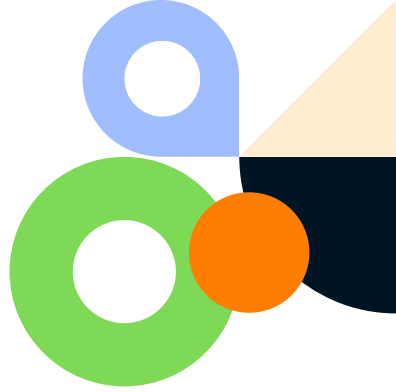
```java
public static void main(String[] args) {
    List<Integer> numbers = List.of(1, 2, 3, 4, 5, 6, 7, 8, 9, 10);

    // Use takeWhile to get numbers until they are less than 6
    List<Integer> takenNumbers = numbers.stream()
            .takeWhile(num → num < 6)
            .toList();
    System.out.println("Taken while < 6: " + takenNumbers);
}
```

Output:

Taken while < 6: [1, 2, 3, 4, 5]

Palmurugan C

Java Streams - dropWhile



This method allow you to control stream processing based on a condition. ***dropWhile*** skips elements until the predicate fails, then includes all subsequent elements

```java
public static void main(String[] args) {
    List<Integer> numbers = List.of(1, 2, 3, 4, 5, 6, 7, 8, 9, 10);

    // Use dropWhile to skip numbers until they are greater than or equal to 6
    List<Integer> droppedNumbers = numbers.stream()
            .dropWhile(num -> num < 6)
            .toList();
    System.out.println("Dropped while < 6: " + droppedNumbers);
}
```

Output:

Dropped while < 6: [6, 7, 8, 9, 10]

Palmurugan C