

Java 8 Streams

A stream represents a sequence of elements and supports different kind of operations to perform computations upon those elements:

```
List<String> myList =  
    Arrays.asList("a1", "a2", "b1", "c2", "c1");
```

```
myList  
    .stream()  
    .filter(s -> s.startsWith("c"))  
    .map(String::toUpperCase)  
    .sorted()  
    .forEach(System.out::println);
```

Stream operations are either intermediate or terminal. Intermediate operations return a stream so we can chain multiple intermediate operations without using semicolons. Terminal operations are either void or return a non-stream result. **In the above example filter, map and sorted are intermediate operations whereas forEach is a terminal operation. For a full list of all available stream operations see the [Stream Javadoc](#). Such a chain of stream operations as seen in the example above is also known as *operation pipeline***

```
Arrays.asList("a1", "a2", "a3")  
    .stream()  
    .findFirst()  
    .ifPresent(System.out::println); // a1
```

<https://www.youtube.com/watch?v=WWXPQUgXagk>