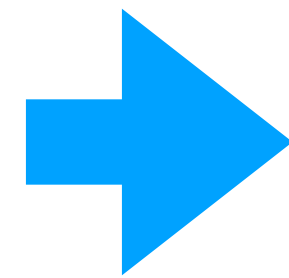


Stream API

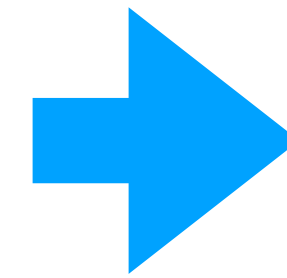
java.util.stream.Stream

- 单向迭代，不可往复
- 可并行操作
- 数据聚合操作
- Stream<T> IntStream DoubleStream LongStream

生成Stream



数据转换



数据操作

生成 Stream

- `Collection.stream()`
- `Collection.parallelStream()`
- `Arrays.stream(T[] array)`
- `BufferedReader.lines()`

Stream的操作

- Intermediate // 中间操作、中级操作； 可多个中间操作； 惰性

• filter map flatMap peek distinct sorted limit skip
过滤 一对一 一对多转换 逐项操作 去重 排序 返回前N个 舍弃前N个

- Terminal // 末端操作 终极操作； 只可一个； 立即执行； 执行后流不可再次被使用

• forEach toArray reduce collect min max count
.collect(Collectors.toList())

anyMatch allMatch noneMatch findFirst findAny

返回值boolean类型， 判断流内数据是否符合条件

map vs. flatMap

```
Stream<String> stream = Stream.of("hello", "world");  
stream.map(s -> s.split("")).foreach(System.out::println);
```

// *map*是一对一，结果是`[["h","e","l","l","0"], ["w","o","r","l","d"]]`

// *map*的参数*lambda*表达式的返回结果可以是任意*Object*，实现原流到此*Object*流的转换

```
Stream<String> stream = Stream.of("hello", "world");  
stream.flatMap(s -> Arrays.stream(s.split("))).foreach(System.out::println);
```

// *flatMap*是一对多，结果是`["h","e","l","l","0", "w","o","r","l","d"]`

// *flatMap*的参数*lambda*表达式返回结果是*Stream<T>*，

// 每个转换结果形成的多个*Stream<T>*会被合并到一起，实现原流到*T*流的转换