Intermediate Operations that operate on every element

Return Type	Operation	Description
Stream<R>	map(Function<? super T,? extends R> mapper)	This is a function applied to every element in the stream. Because it's a function, the return type can be different, which has the effect of transforming the stream to a different stream of different types.
Stream<T>	peek(Consumer<? super T> action)	This function doesn't change the stream, but allows you to perform some interim consumer function while the pipeline is processing.
Stream<T>	sorted() sorted(Comparator<? super T> comparator)	There are two versions of sorted. The first uses the naturalOrder sort, which means elements in the stream must implement Comparable. If your elements don't use Comparable, you'll want to use sorted and pass a Comparator.

{LP} LearnProgramming .academy

Primitive Streams

In addition to the generic Stream, that lets you stream any reference type, Java has three primitive streams.

Special Primitve Streams	Mapping from Reference Type to Primitive	Mapping from Primitive Stream to Reference Type
DoubleStream	mapToDouble(ToDoubleFunction<? super T> mapper)	mapToObj(DoubleFunction<? extends U> mapper) boxed()
IntStream	mapToInt(ToIntFunction<? super T> mapper)	mapToObj(IntFunction<? extends U> mapper) boxed()
LongStream	mapToLong(ToLongFunction<? super T> mapper)	mapToObj(LongFunction<? extends U> mapper) boxed()

{LP} LearnProgramming .academy