Stream Types May Change As the Pipeline Process Progresses

An intermediate operation can usually be recognized by it's signature, because it returns a stream.

I want to point out, that this doesn't mean the element type of the stream can't change.

In practice, you'll be regularly transforming your stream element to a different type.

Stream Sources

This slide shows the eight methods I covered in this video.

Two can produce infinite streams, the Stream.generate method as well as Stream.iterate, which doesn't include a Predicate parameter.

Method	Finite	Infinite
`Collection.stream()`	X	
`Arrays.stream(T[])`	X	
`Stream.of(T...)`	X	
`Stream.iterate(T seed, UnaryOperator<T> f)`	X	X
`Stream.iterate(T seed, Predicate<? super T> p, UnaryOperator<T> f)`	X	
`Stream.generate(Supplier<? extends T> s)`		X
`**IntStream.range(int startInclusive, int endExclusive)`	X	
`**IntStream.rangeClosed(int startInclusive, int endExclusive)`	X	

```
** range and rangeClosed also available on DoubleStream and LongStream, with double and long types produced.
```