

# Essential Streams



# DataStreams

The fundamental abstraction of a stream in Flink

Can be transformed with FP

- map
- flatMap
- filter
- reduce
- process

```
// map
val doubledNumbers: DataStream[Int] = numbers.map(_ * 2)

// flatMap
val expandedNumbers: DataStream[Int] = numbers.flatMap(n => List(n, n + 1))

// filter
val filteredNumbers: DataStream[Int] = numbers.filter(_ % 2 == 0)
```

# Flink Application

Needs an environment to run

- rich data structure with access to all Flink APIs
- description of all streams and transformations
- lazy evaluation

```
// execution environment
val env: StreamExecutionEnvironment = StreamExecutionEnvironment.getExecutionEnvironment

// in between, add any sort of computations
val simpleNumberStream: DataStream[Int] = env.fromElements(1,2,3,4)

// perform some actions
simpleNumberStream.print()

// at the end
env.execute() // trigger all the computations that were DESCRIBED earlier
```

**Flink rocks**

