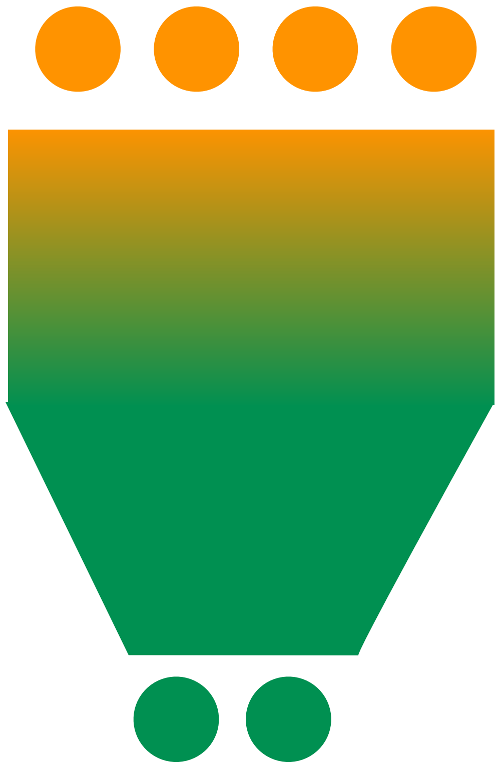
Streams: Coding in a Context

Jessica Kerr
jessitron@jessitron.com
@jessitron

lazy Streams

parallel Stream

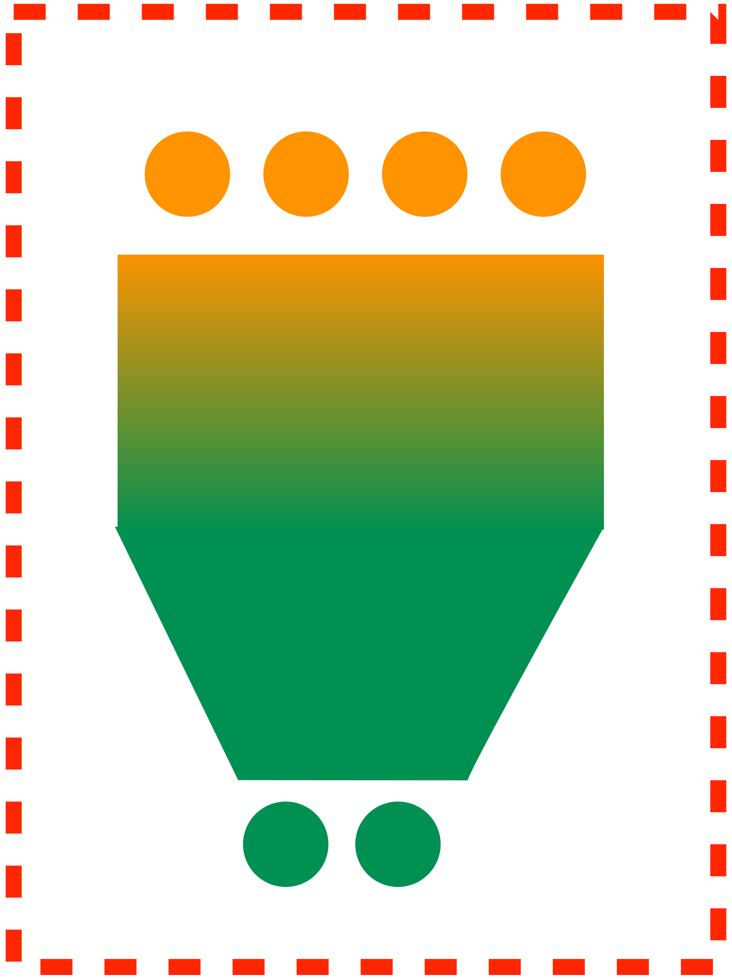
.methods()



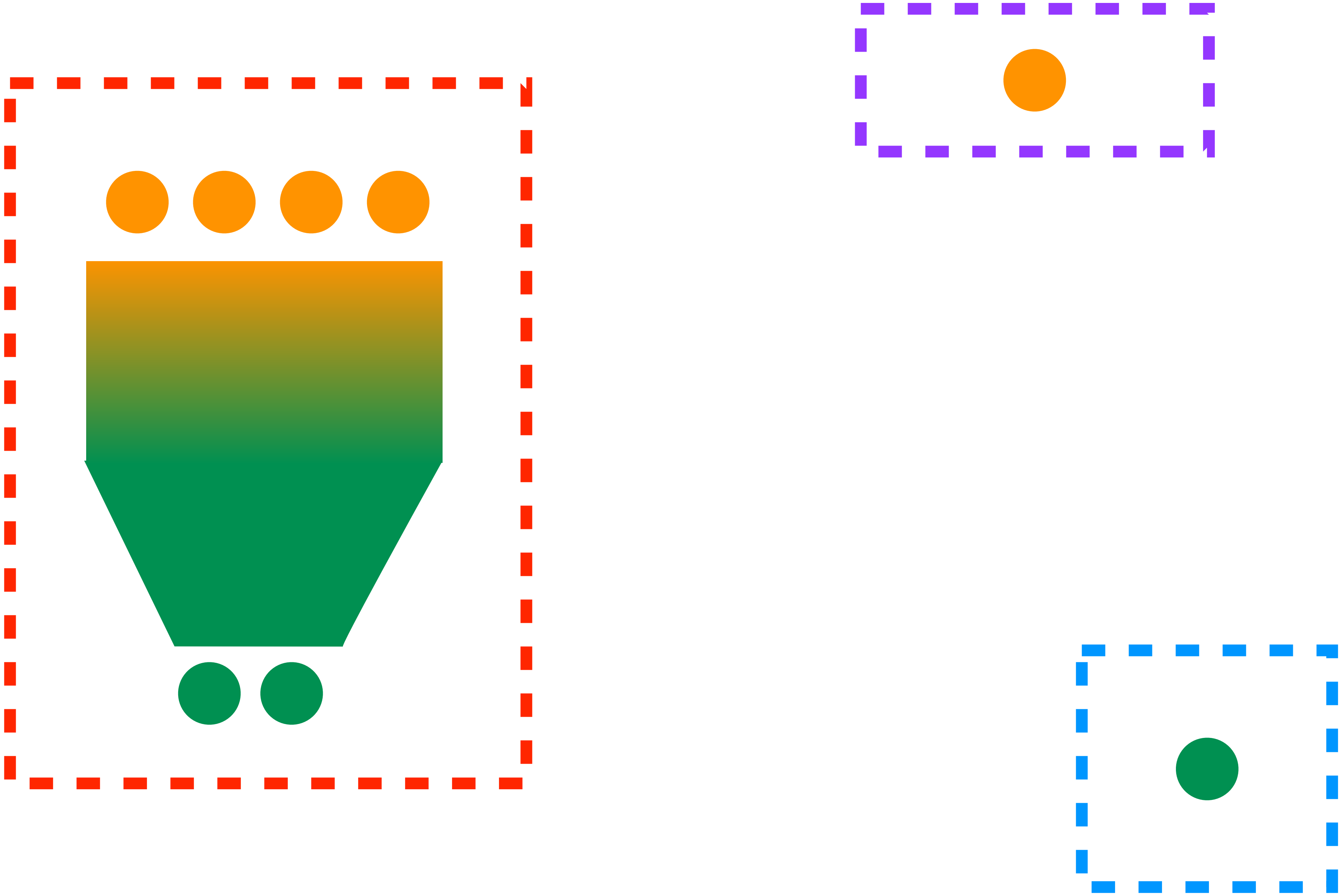
Stream.map( )

.filter( )

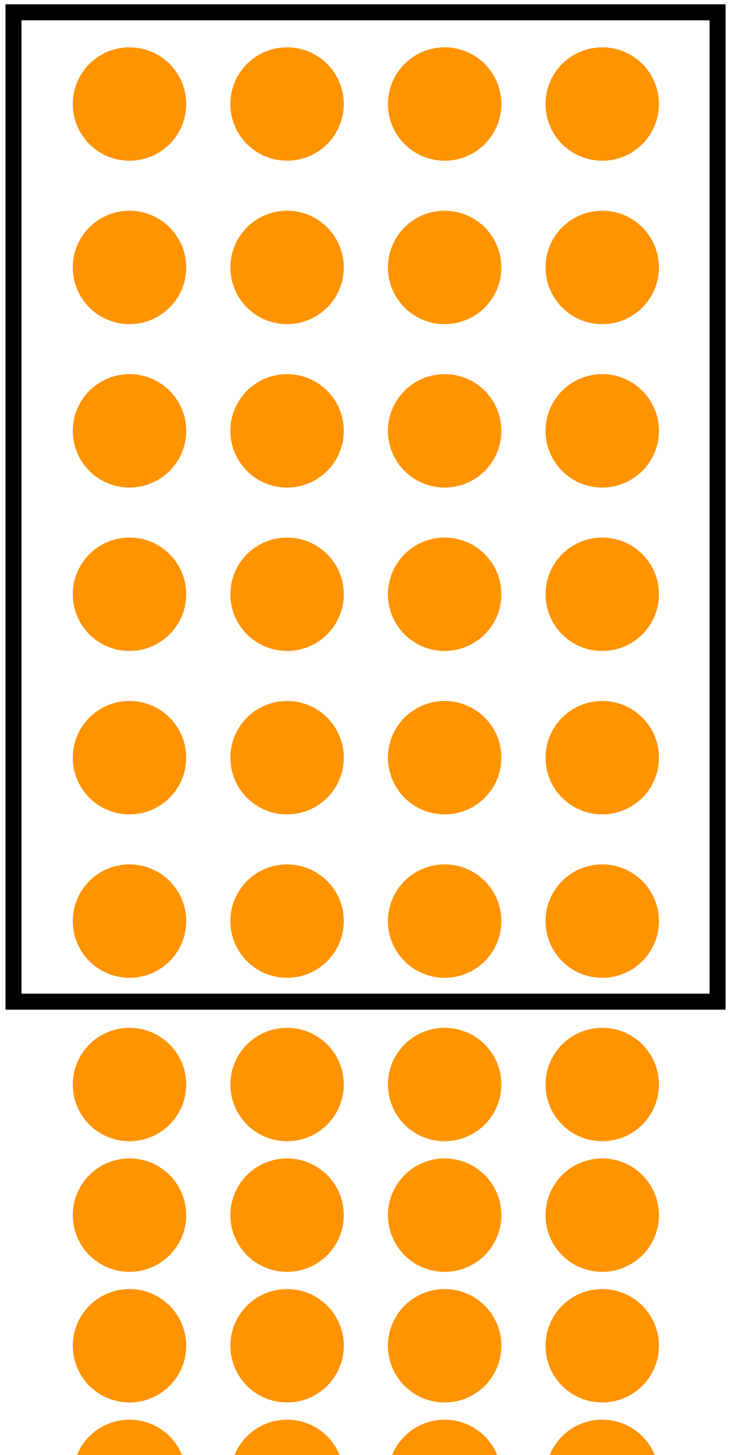
Context



Contexts



BONUS!

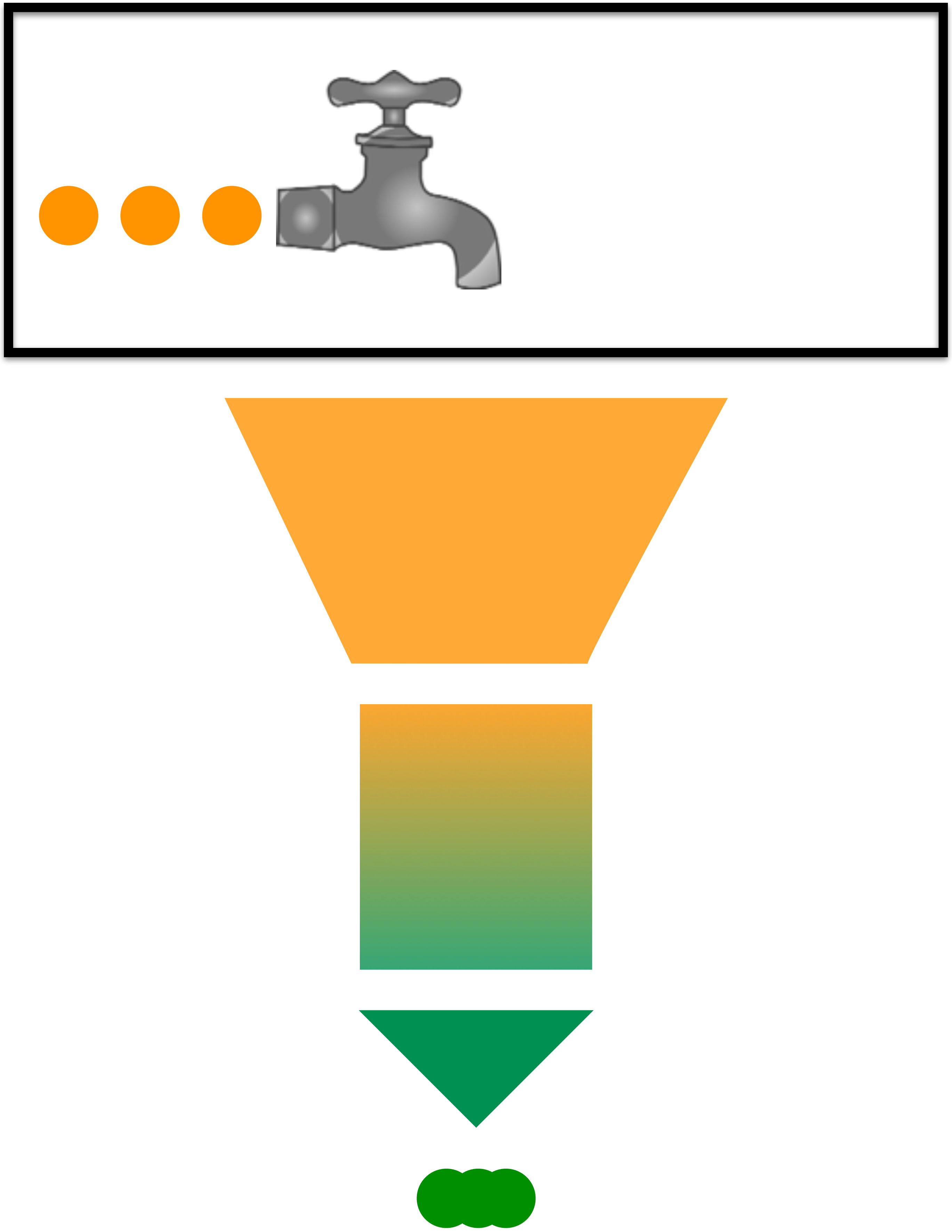


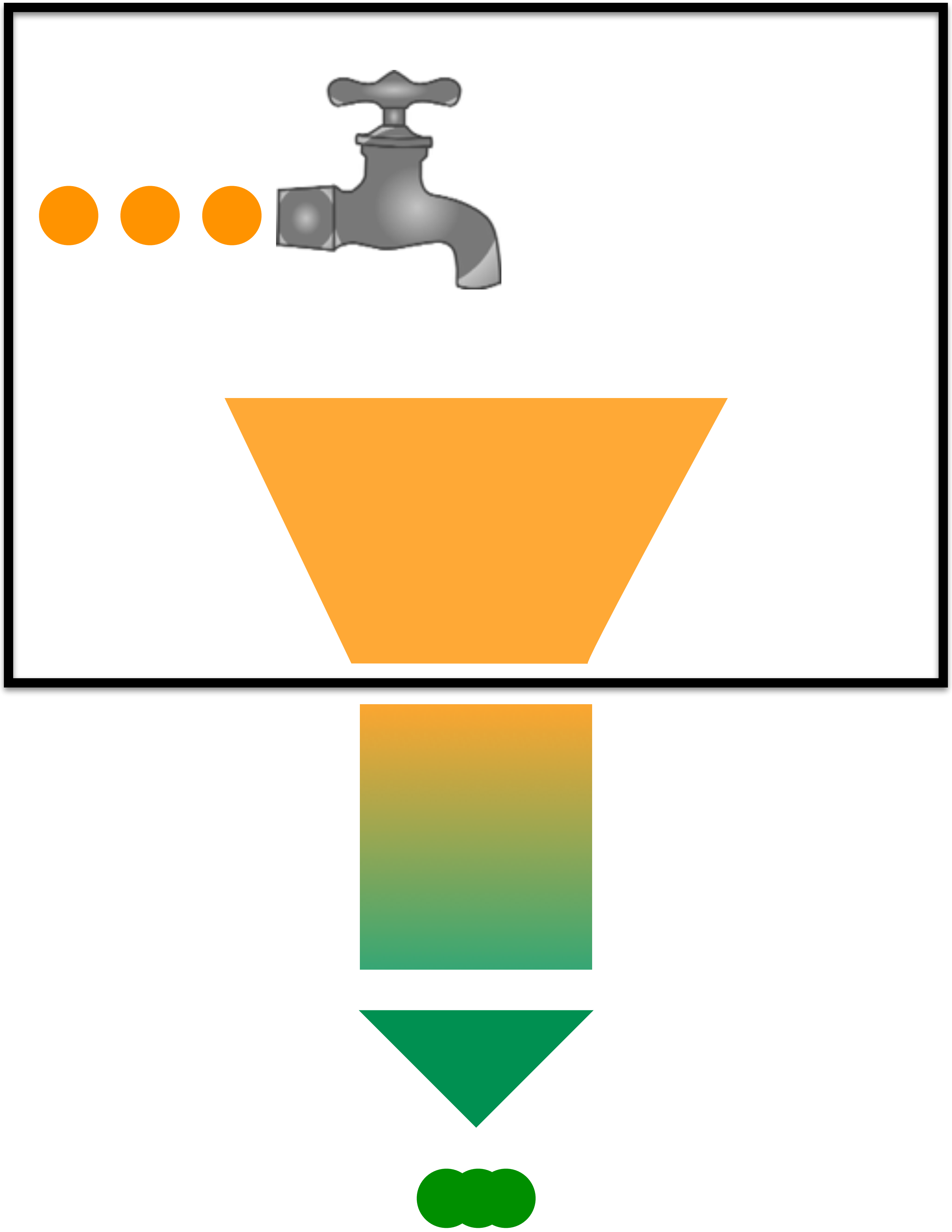
How can Streams be
more efficient than loops?

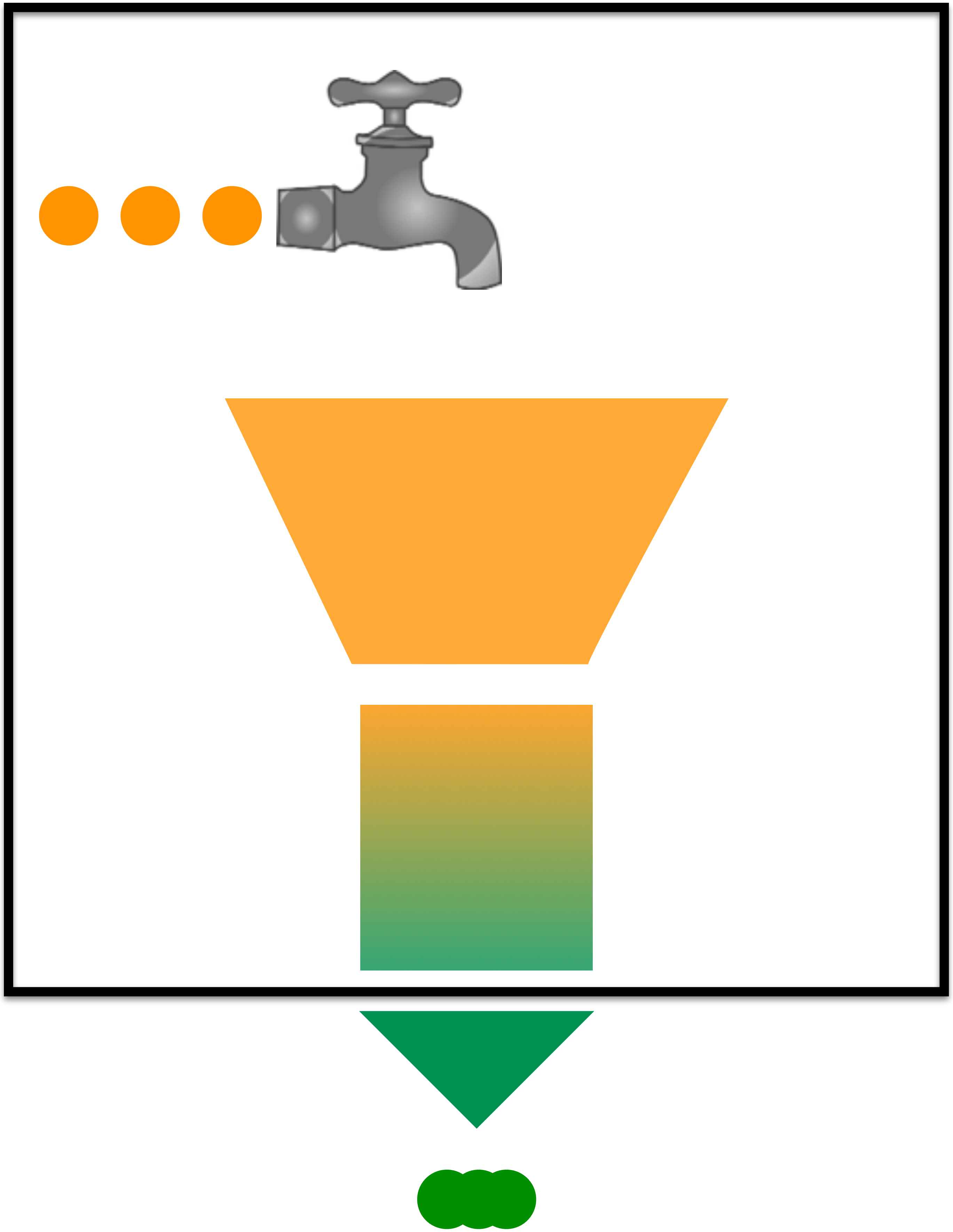
filter

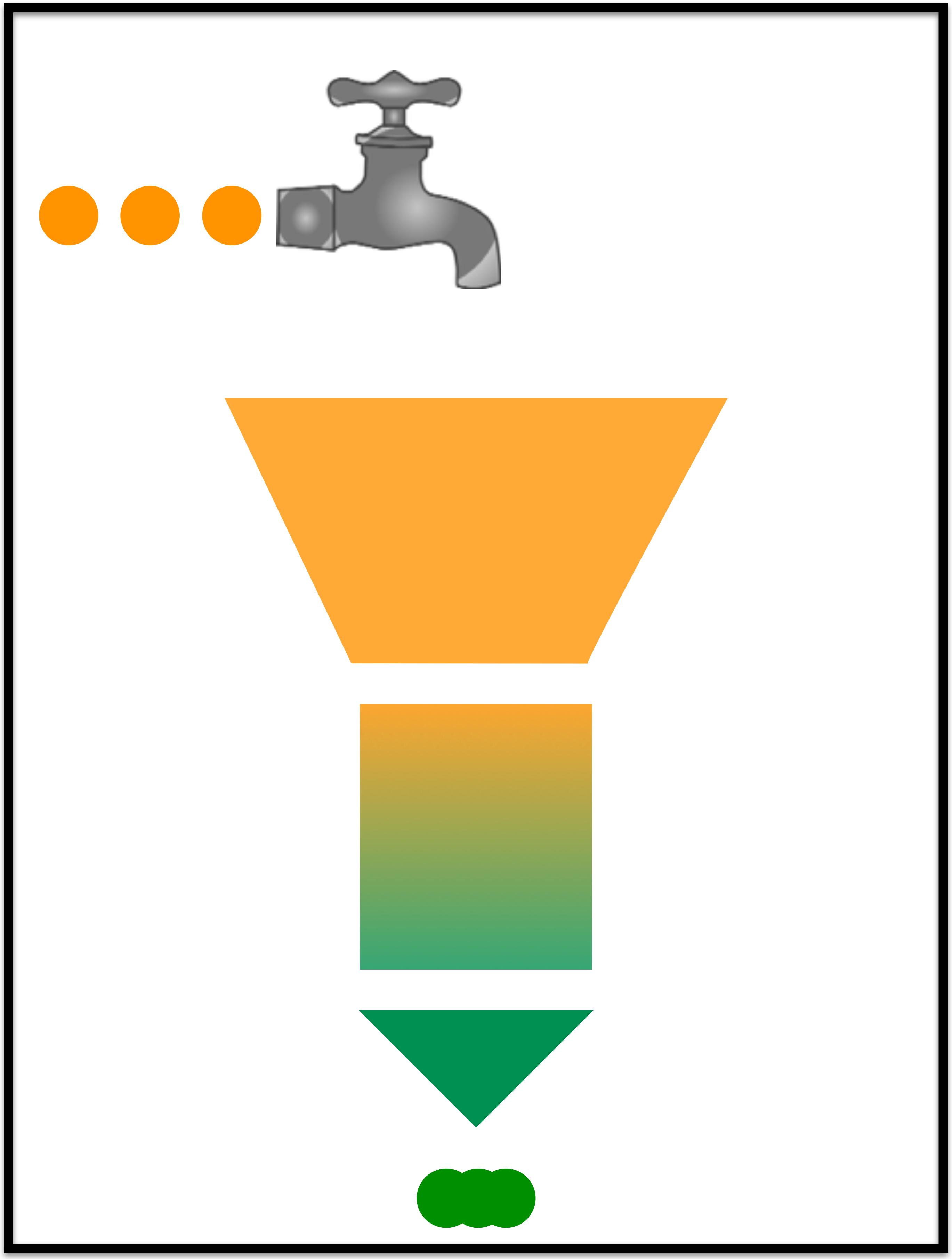
map

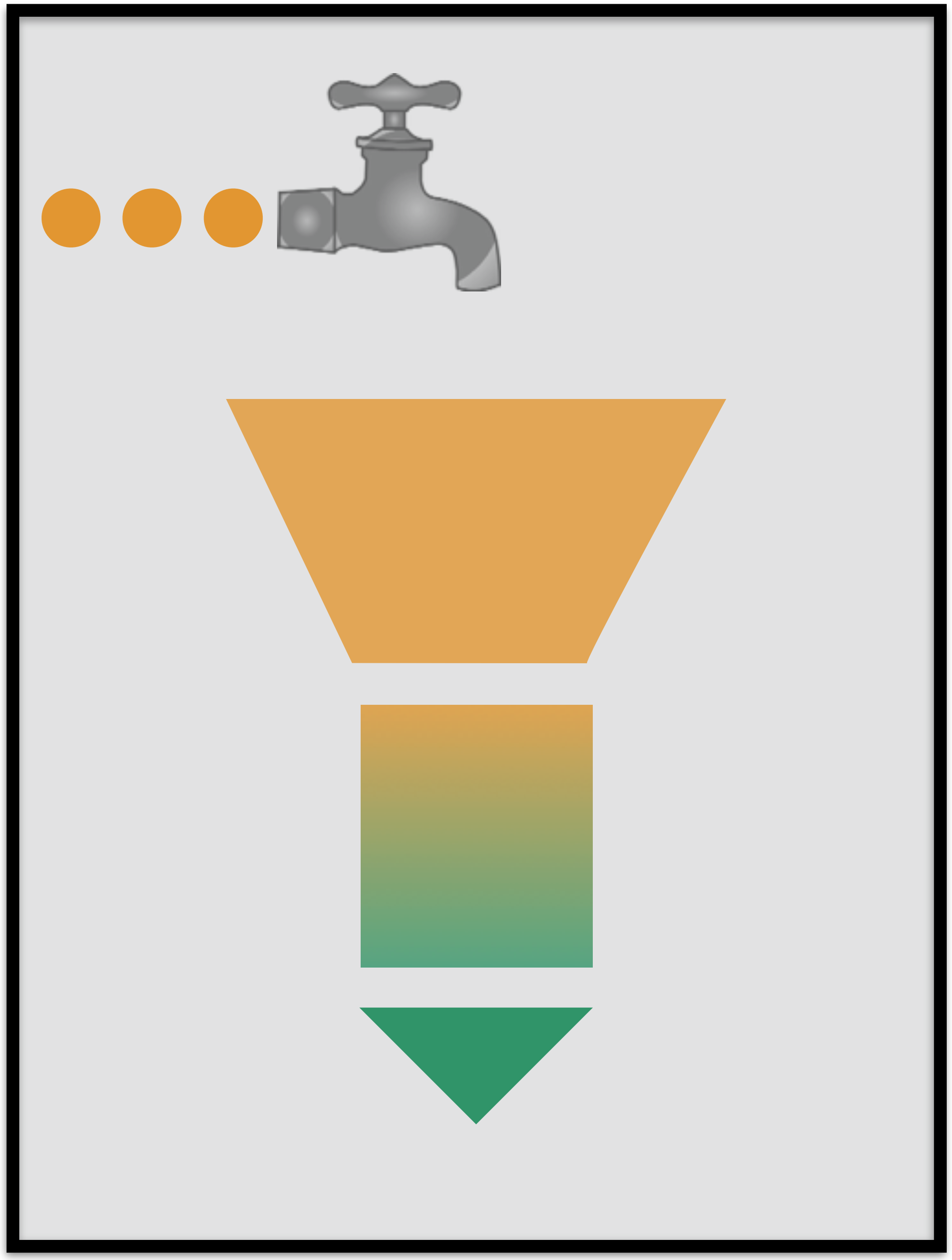
reduce

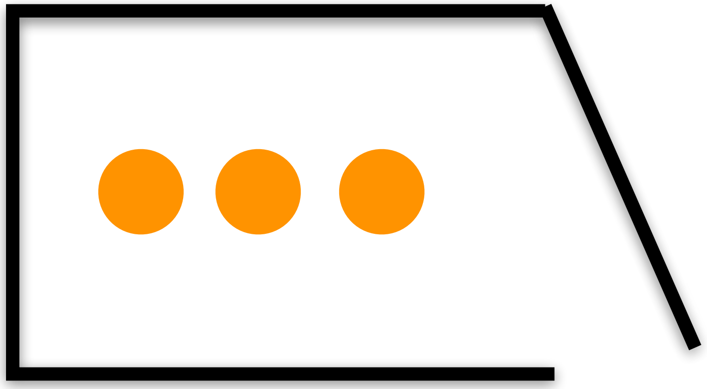


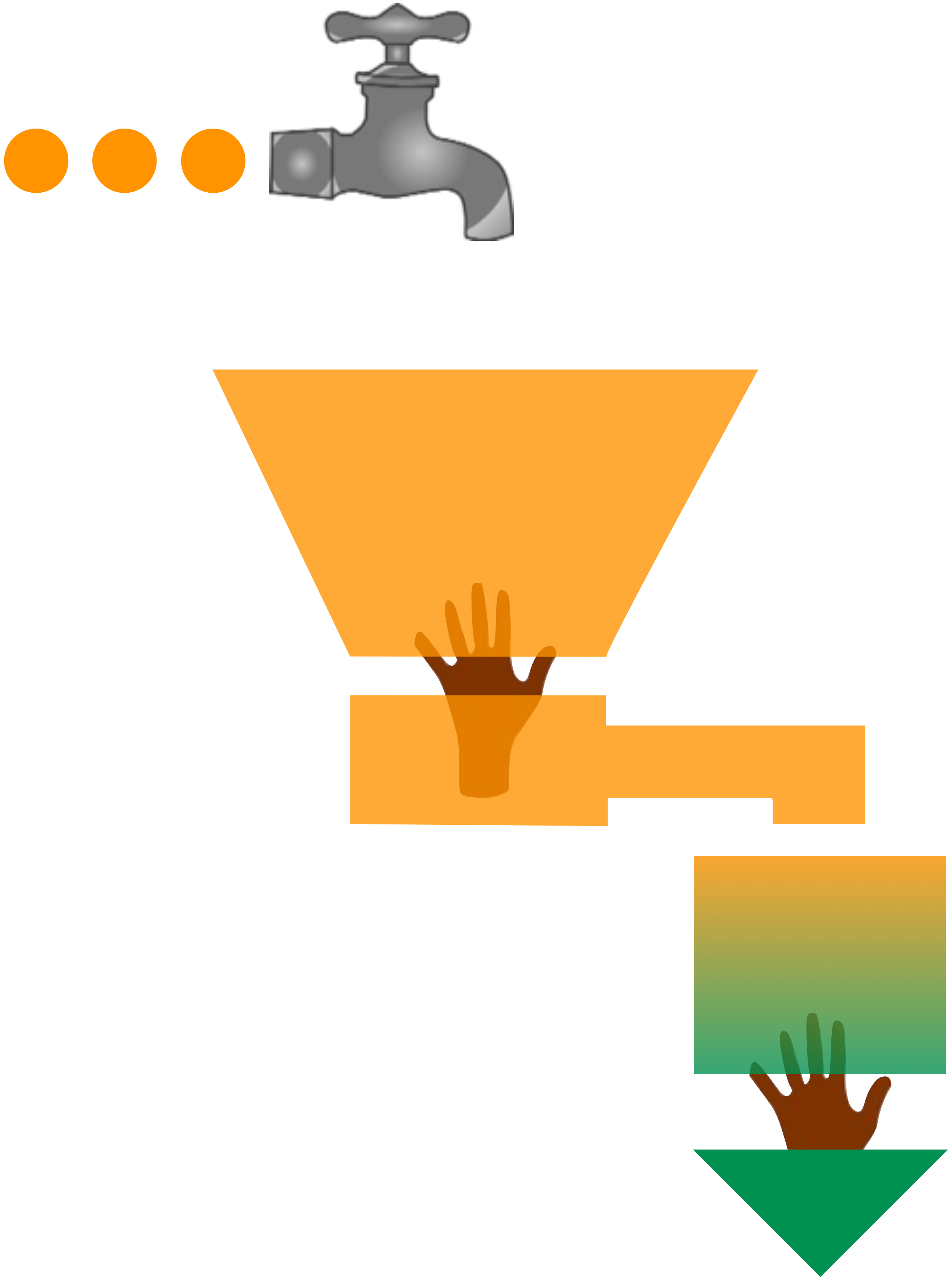












filter

limit

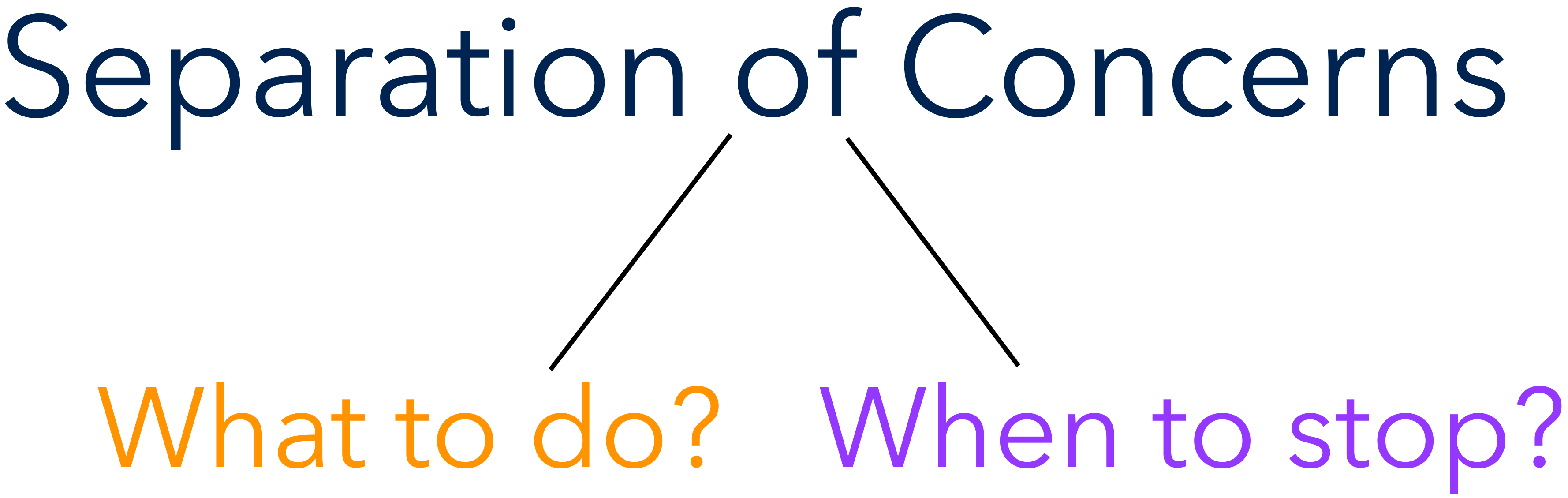
map

reduce

Efficiency

Separation of Concerns

Efficiency



Separation of Concerns

What to do?

When to stop?

BUT FIRST…

Let's see some **Stream operations**!

Terminal Operators

trigger processing

return a value

terminate the Stream instance

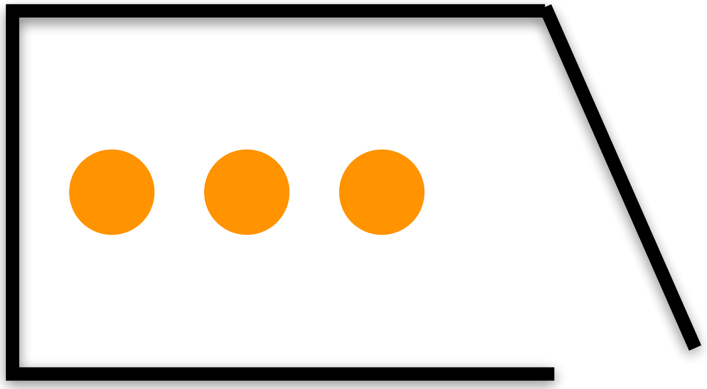
Intermediate Operators

trigger no processing

return a Stream

let you keep going

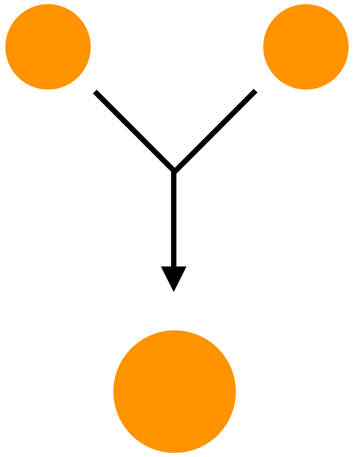
Stream



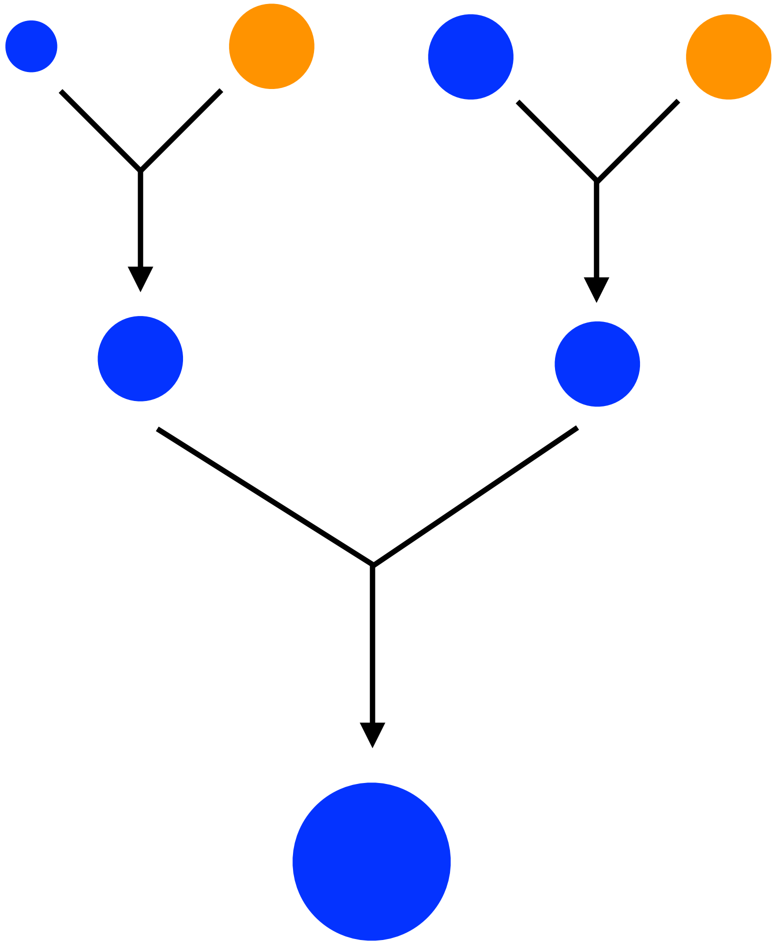
Iterator

one-time traversal

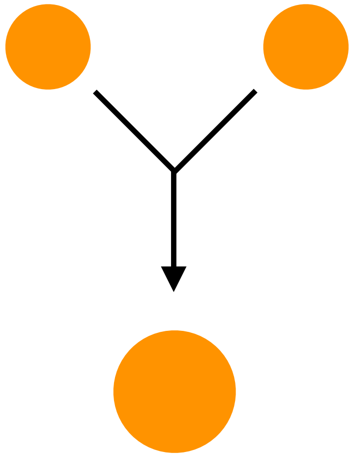
reduce



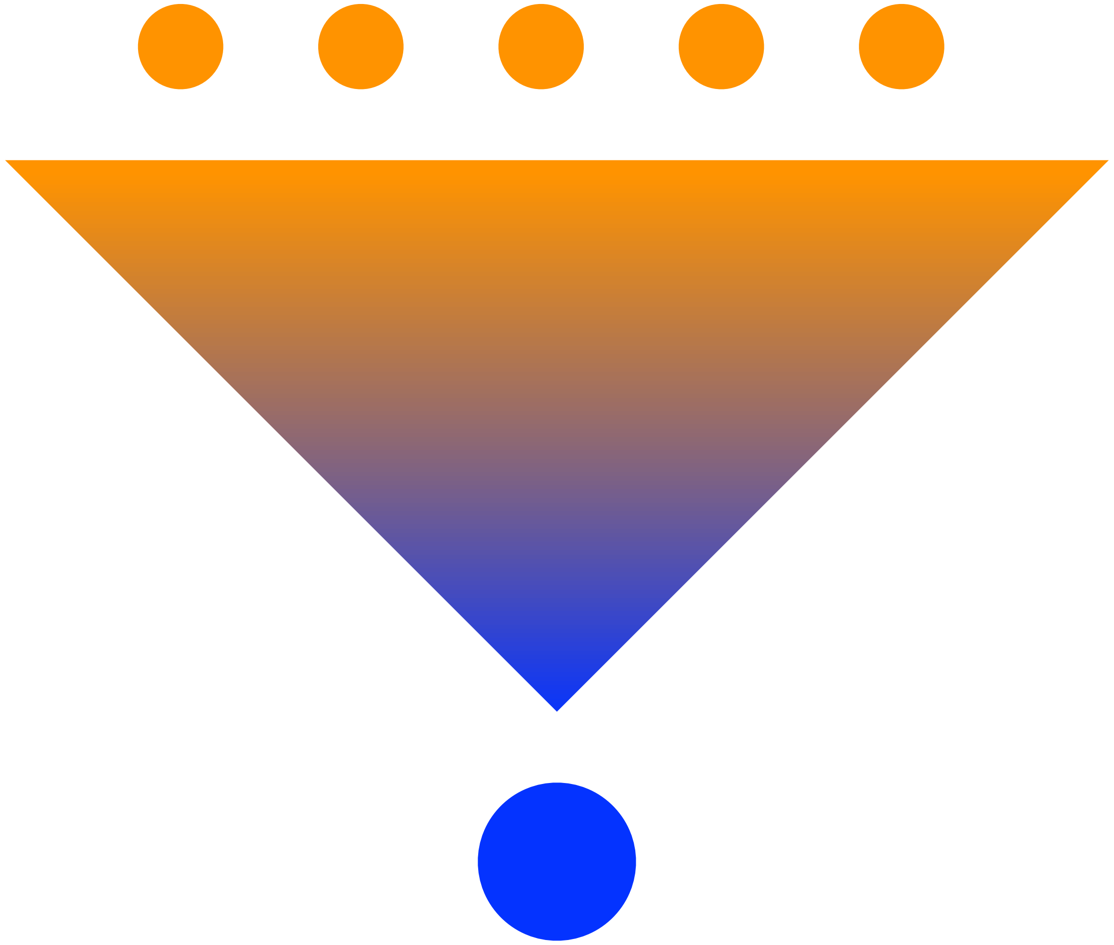
collect



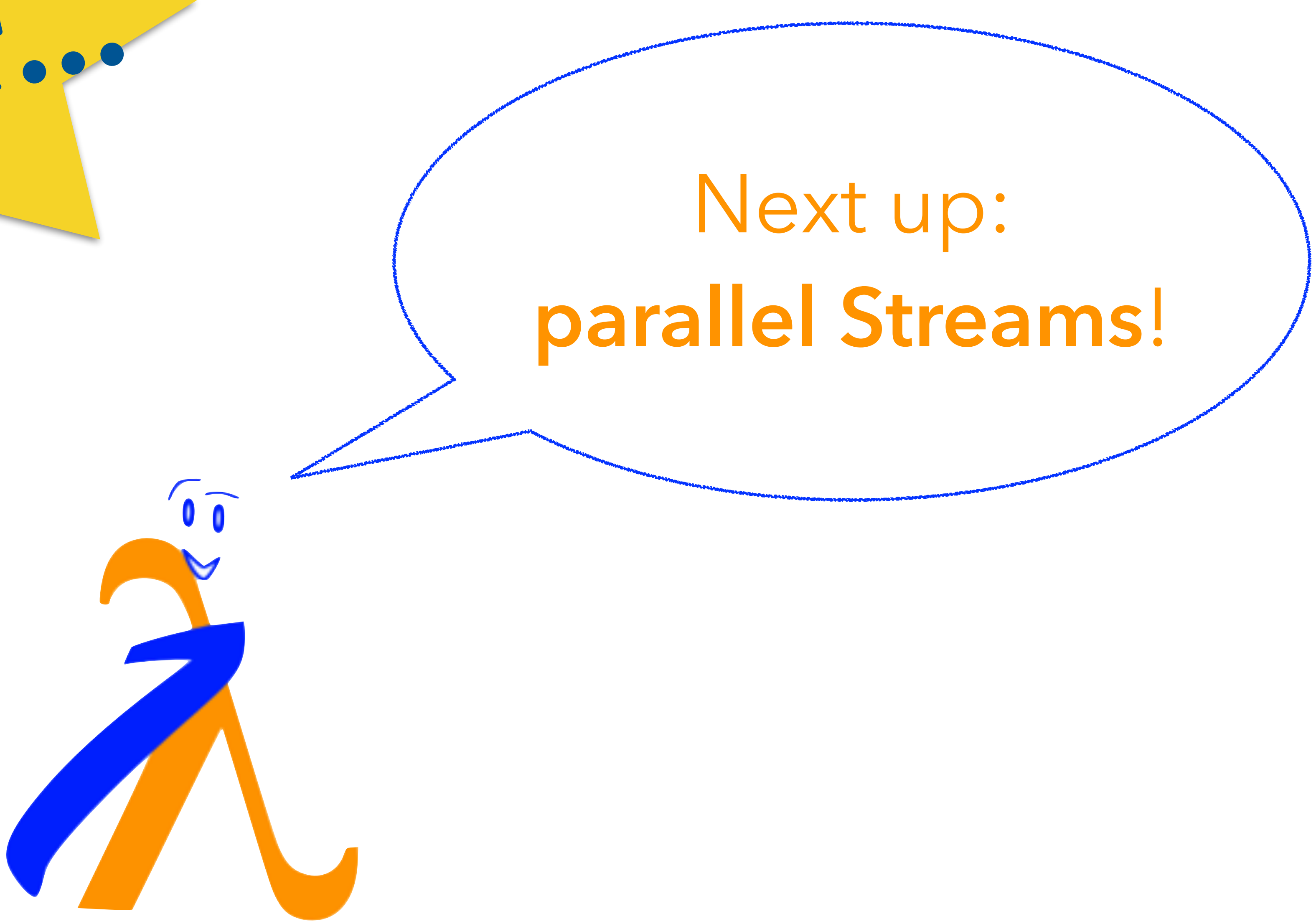
reduce



collect



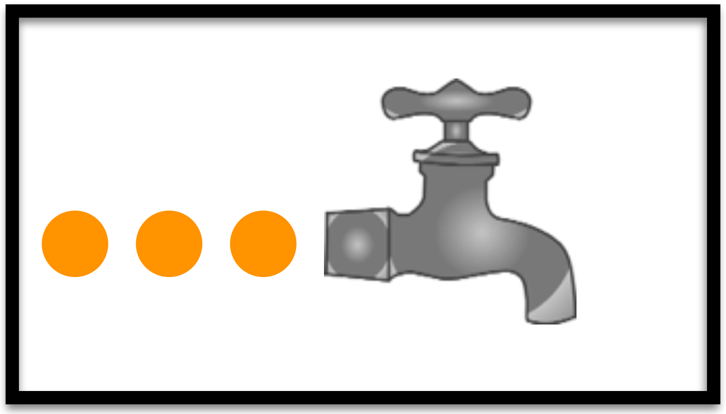
BUT FIRST…



BUT FIRST…

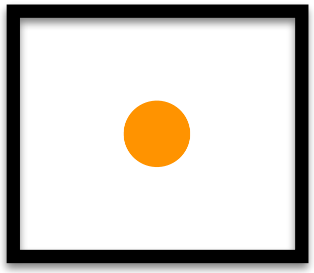
Infinite streams need short-circuiting operations.

Stream

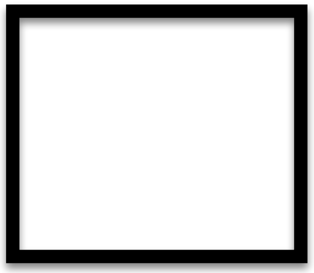


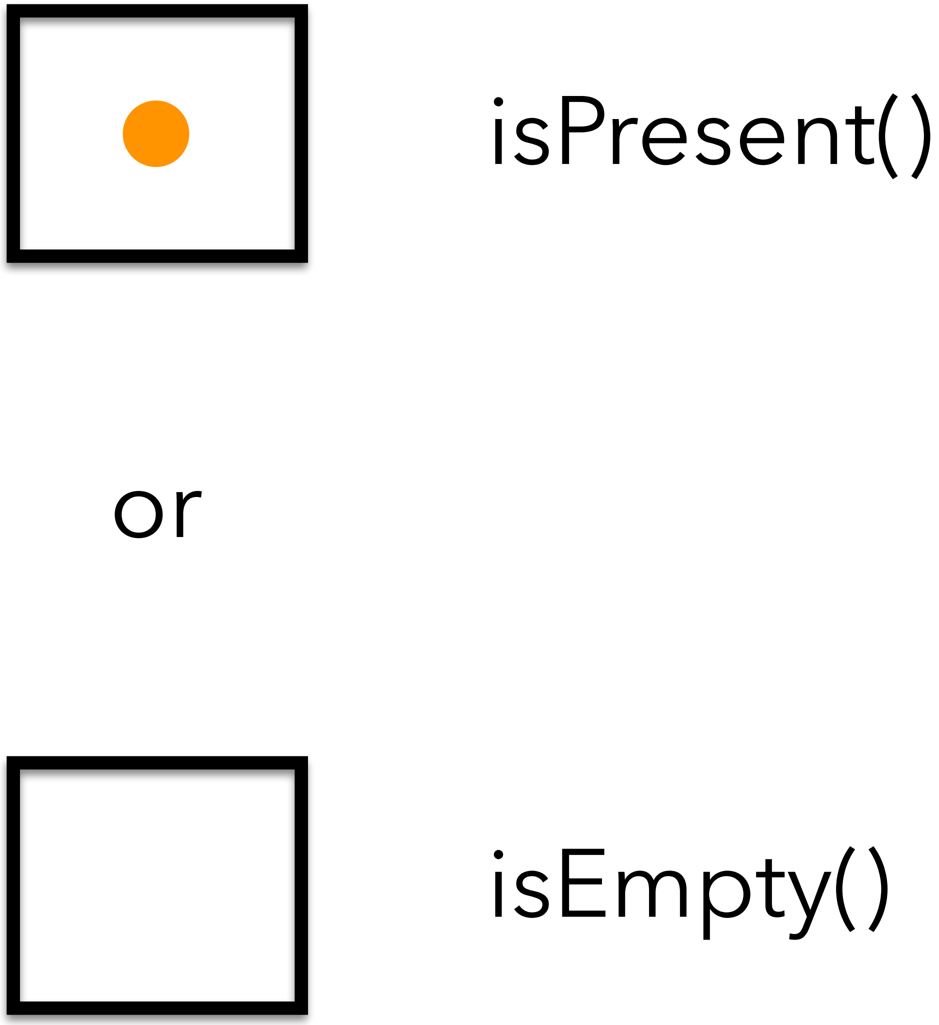
to

Optional



or





isPresent()

or

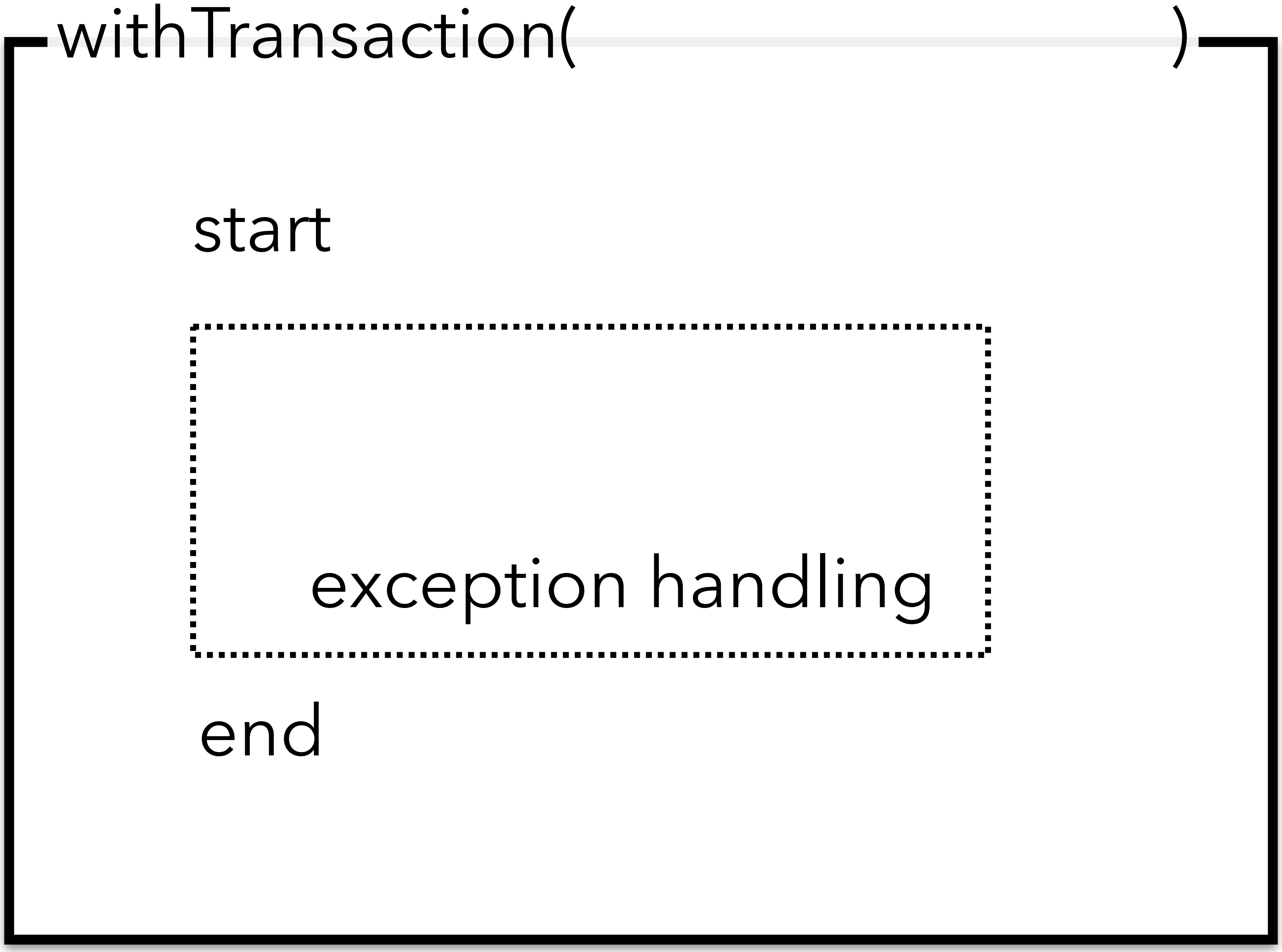
isEmpty()

OW!
null

Stream

Optional

Function ?



withTransaction( )

start

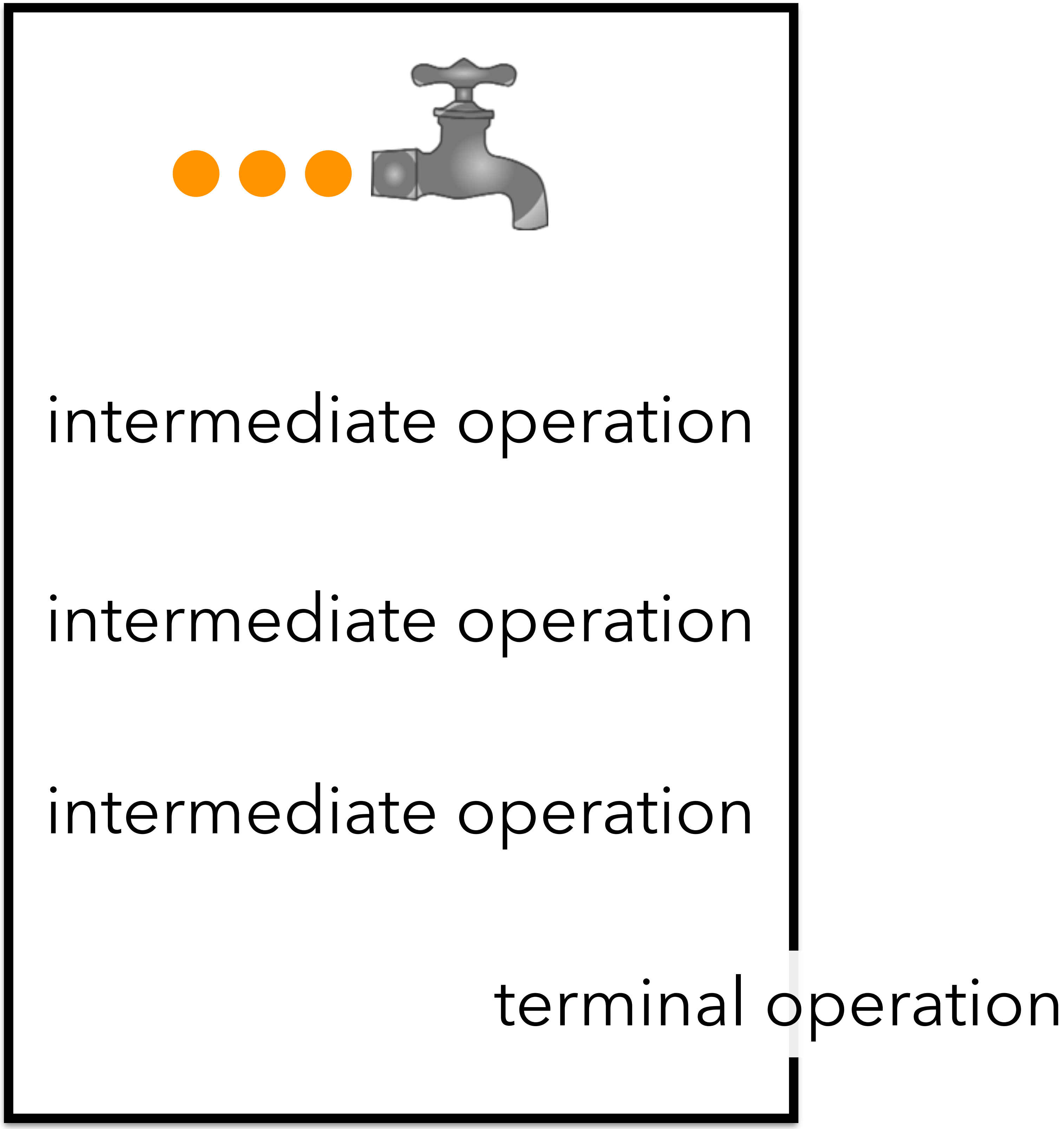
exception handling

end

Review Stream

Java 6

Secret Functional Programmer Terms

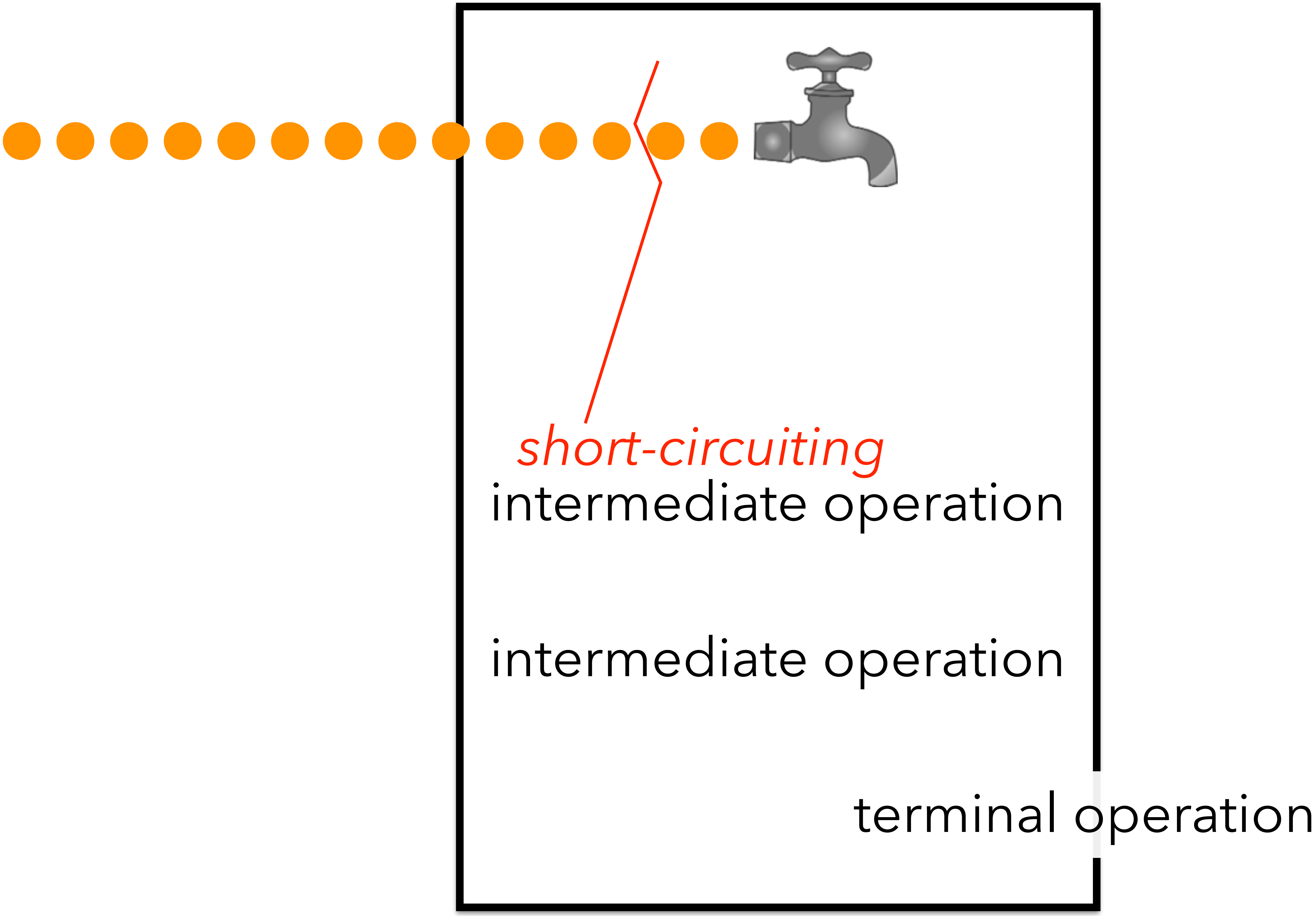


intermediate operation

intermediate operation

intermediate operation

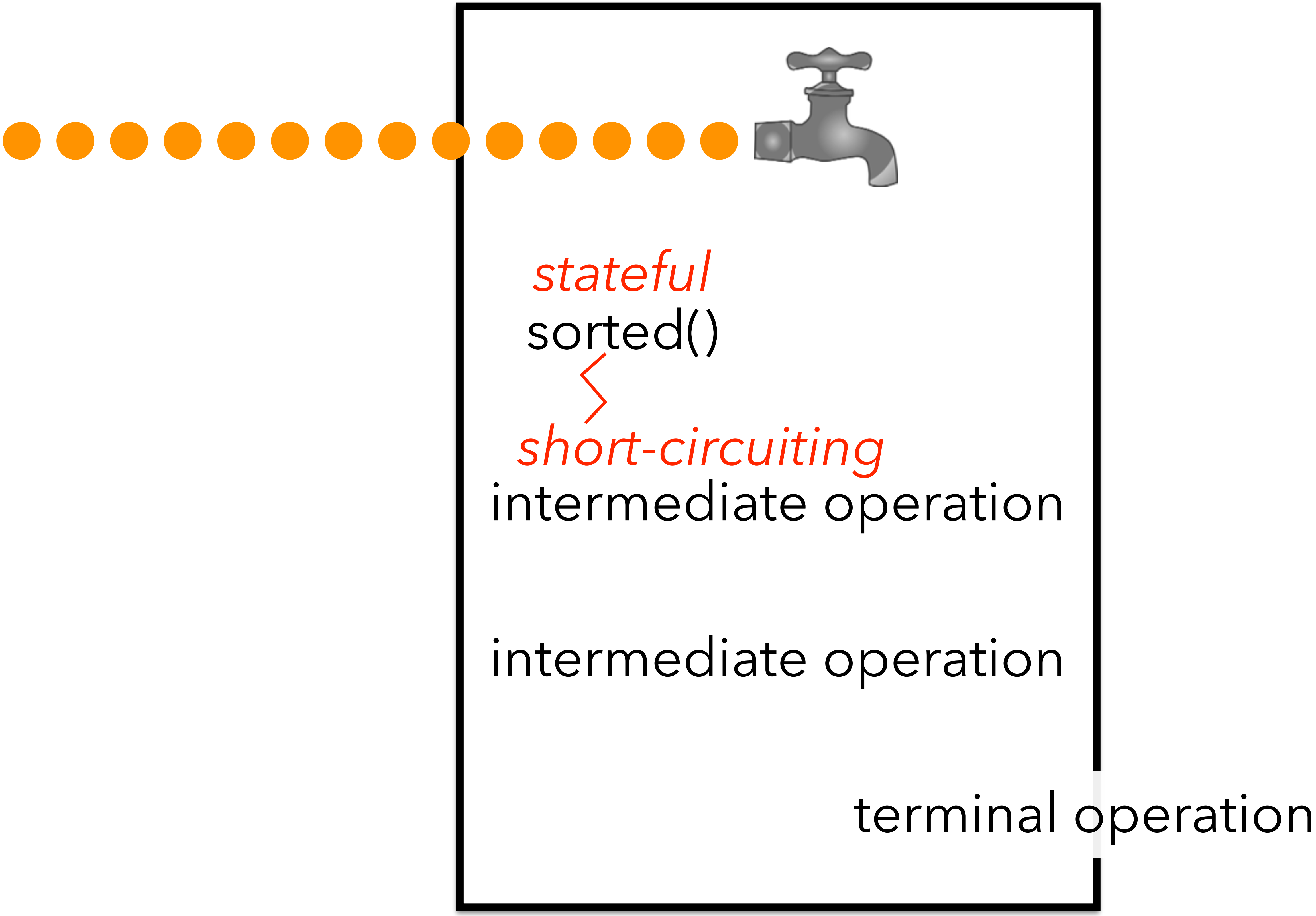
terminal operation



short-circuiting
intermediate operation

intermediate operation

terminal operation



stateful

sorted()

short-circuiting

intermediate operation

intermediate operation

terminal operation

see also:

Stream<T>

DoubleStream

LAZY

Stream

ONE TIME

LAZY

Guava
FluentIterable

MANY TIMES

Guava
Optional

~~Optional.map()~~

~~Optional.flatMap()~~

<table>
<tr><th>Stream</th><th>Guava
FluentIterable</th></tr>
<tr><td colspan="2">anyMatch</td></tr>
<tr><td colspan="2">limit</td></tr>
<tr><td>count</td><td>size</td></tr>
<tr><td>findFirst</td><td>first</td></tr>
<tr><td>flatMap</td><td>transformAndConcat</td></tr>
<tr><td>distinct</td><td>toSet</td></tr>
<tr><td>sorted</td><td>toSortedList</td></tr>
<tr><td>Collectors.groupingBy</td><td>index</td></tr>
</table>

Guava
ListenableFuture

map

flatMap

wrapping
constructor

map

flatMap

Stream.of(value)
Optional.of(value)

map

flatMap

wrapping constructor

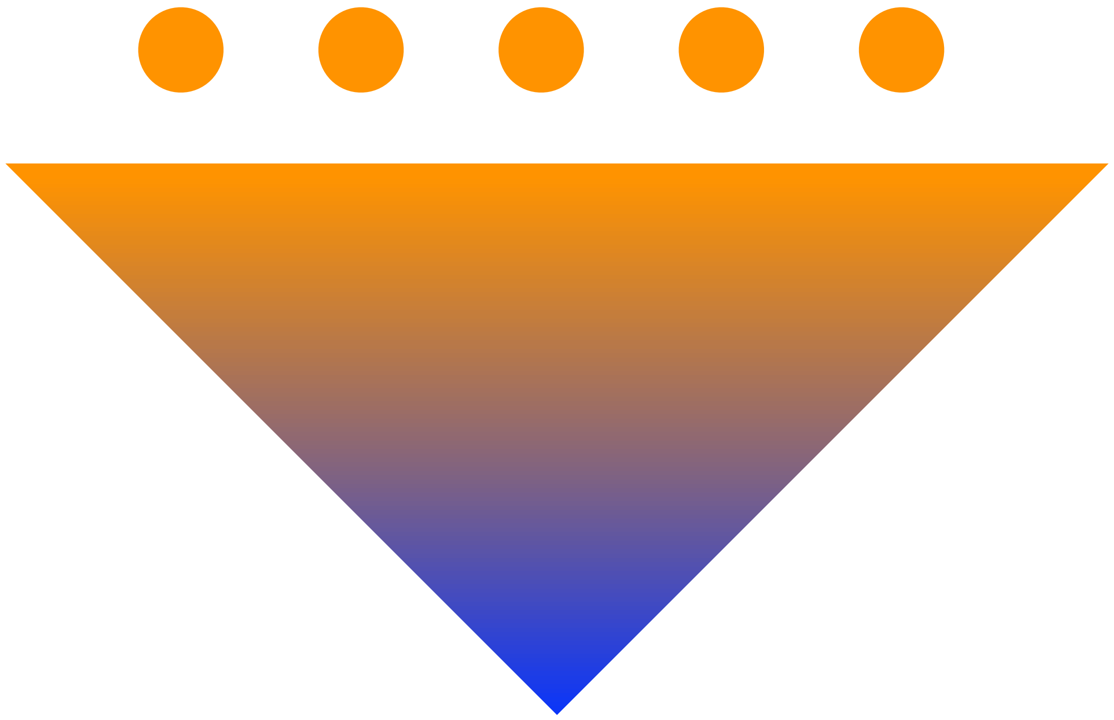
MONAD

Stream

MONAD

Optional

collect



also known as fold

Functions as Values

STORE functions in variables

PASS functions in parameters

RETURN functions from other functions

Remove duplication

PASS functions in parameters

Stream processing

Contexts