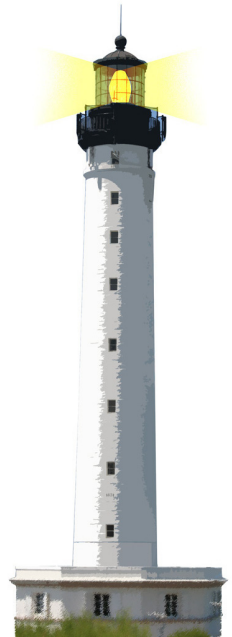
Stream Overview

Damien Cassou, Stéphane Ducasse and Luc Fabresse

W3S10

Pharo

http://www.pharo.org

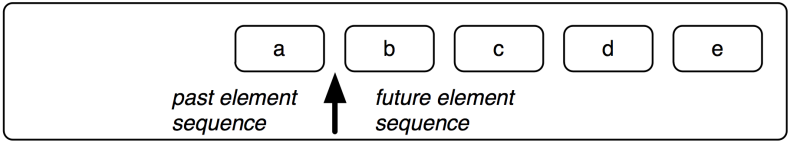


What You Will Learn

- What streams are?
- How to use them

Streams

- Iterate over a sequence of elements
  - e.g., collection, file, network
- Current position separates past from future



API Overview

- Creating
  - anObject readStream
  - anObject writeStream
  - Collection streamContents: [:stream | ...]
  - (Read/Write)Stream on: aCollection
- Reading elements
  - next
  - upTo: anObject
  - upToEnd
- Writing elements
  - nextPut: anElement
  - nextPutAll: aCollection

Reading Elements

- next, upTo: anObject, upToEnd

```
| strean |
stream := #($a $b $c $d $e $f) readStream.

stream next.
> $a

stream upTo: $d.
>  #($b $c)

stream position.
> 4

stream upToEnd
> #($e $f)
```

Writing Elements

- nexPut:, nextPutAll:

```
stream := (Array new: 6) writeStream.

stream nextPut: 1.
stream contents.
> #(1)

stream nextPutAll: #(4 8 2 6 7).
stream contents.
> #(1 4 8 2 6 7)
```

Writing to a File

```
stream := 'hello.txt' asFileReference writeStream.
stream nextPutAll: 'Hello Pharo!'.
stream close.
```

Reading from a File

```
stream := 'hello.txt' asFileReference readStream.

stream next.
> $H

stream upToEnd.
> 'ello Pharo!'

stream close
```

Creating Collections from a Stream

When you want to create a collection by writing to a stream:

```
stream := OrderedCollection new writeStream.
stream nextPut: 1.
stream contents
```

is equivalent to

```
OrderedCollection
  streamContents: [:stream | stream nextPut: 1]
```

Both scripts return a collection with number 1 inside.

What You Should Know

- A stream can read from and write to
  - collections, files and network
- A stream maintains a current position
- A stream can help create collections

A course by

and

in collaboration with

Inria 2016

Except where otherwise noted, this work is licensed under CC BY-NC-ND 3.0 France
https://creativecommons.org/licenses/by-nc-nd/3.0/fr/