

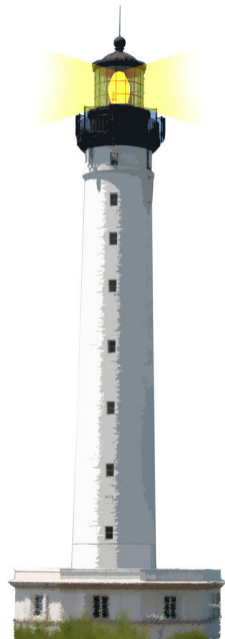
Files in Pharo

Damien Cassou, Stéphane Ducasse and Luc Fabresse

W5S07



<http://www.pharo.org>



What You Will Learn

- Navigating between directories
- Creating and removing directories
- Listing files
- Reading from and writing to files



Getting Started

Some standard directories

`FileLocator home`. "User's home directory (~)"

`FileLocator root`. "File system's root (/)"

`FileLocator C`. "Windows C:\"

Accessing the children of a directory

```
| home |
```

```
home := FileLocator home.
```

```
home pathString
```

```
> '/home/cassou/'
```

```
home children
```

```
> an Array({home}/.bashrc {home}/Music ...)
```

Navigating Up and Down

Use message / to build a path

```
(home / 'Music') directories  
> an Array({home}/Music/Anouar_Brahem ...)
```

Use parent to go up

```
(home / 'Music') parent pathString  
> '/home/cassou/'
```



Creating and Removing Directories

```
| music |  
music := FileLocator home / 'Music'.  
(music / 'Pink_Floyd') isDirectory.  
> false  
  
(music / 'Pink_Floyd') ensureCreateDirectory.  
(music / 'Pink_Floyd') isDirectory.  
> true  
  
(music / 'Pink_Floyd') delete.  
(music / 'Pink_Floyd') isDirectory  
> false
```

Finding Children

```
(pinkFloyd / 'Meddle') allChildrenMatching: '*.ogg'.
```

is equivalent to

```
(pinkFloyd / 'Meddle') allChildren  
  select: [ :each | each basename endsWith: 'ogg' ]
```

Getting Information

```
| pharo |  
pharo := 'pharo5.image' asFileReference.
```

```
pharo isFile.  
> true
```

```
pharo basename  
> 'pharo5.image'
```

```
pharo extension  
> 'image'
```

```
pharo size  
> 23744868
```

```
pharo parent pathString  
> '/home/cassou/Pharo/images/pharo5'
```



Writing to a File

```
| hello stream |  
hello := 'hello.txt' asFileReference.  
hello exists  
> false  
  
stream := hello writeStream.  
stream nextPutAll: 'Hello World'.  
stream close.
```


Reading from a File

```
| hello stream |  
hello := 'hello.txt' asFileReference.  
hello exists  
> true
```

```
stream := hello readStream.  
stream next.  
> $H
```

```
stream upToEnd.  
> 'ello World'
```

```
stream close
```



Automatically Closing File Streams

```
| hello |  
hello := 'hello.txt' asFileReference.
```

Writing

```
hello writeStreamDo: [ :stream |  
  stream nextPutAll: 'Hello World' ].
```

Reading

```
hello readStreamDo: [ :stream | stream contents ]  
> 'Hello World'
```



What You Should Know

- File and directory references are objects
- The simple API facilitates navigation and manipulation
- Reading from and writing to files is done through streams



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/>