

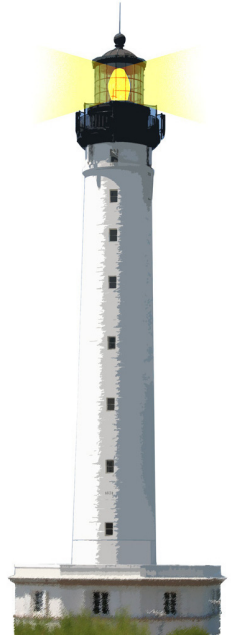
# Characters, Strings and Symbols

Damien Cassou, Stéphane Ducasse and Luc Fabresse

W6S07



<http://www.pharo.org>



# What You Will Learn

- Characters
- Strings: a collection of characters
- Symbols: unique strings



# Characters

- Characters:
  - \$F, \$Q \$U \$E \$N \$T \$i \$N
- Unprintable characters:
  - Character space, Character tab, Character cr



# 'Strings'

Delimited by '

```
'eclair au chocolat'
```

```
'eclair au chocolat'.size  
> 18
```

```
Character space split: 'eclair au chocolat'  
> an OrderedCollection('eclair' 'au' 'chocolat')
```

# A String: a Collection of Characters

```
'eclair au chocolat' at: 1  
> $e
```

```
'eclair au chocolat' do: [:each | Transcript show: each ; cr ]
```

# Quote in Strings

- To add a quote in a string with just type it twice

```
'L'eclair au chocolat'
```

- Pay attention there is only one element

```
'L'eclair au chocolat' at: 2  
> '$'
```

```
'L'eclair au chocolat' at: 3  
> '$e'
```



# Getting the Last Char

```
| str |  
str := 'Tiramisu'.  
str at: str size  
> $u  
  
str last  
> $u
```

# Ways to Obtain a String

```
#mac asString  
> 'mac'
```

```
12 printString  
> '12'
```

```
String with: $A
```





# For Concatenation Use ,

```
'Calvin' , ' & ' , 'Hobbes'  
> 'Calvin & Hobbes'
```

# Take Care With ,

Message comma #, copies strings so multiple concatenations can generate useless intermediate versions

```
'Calvin' , ' & ' , 'Hobbes'  
> 'Calvin & Hobbes'
```

- Benchmark it
- If this is worth, use a stream to avoid creating multiple intermediary strings

```
String streamContents: [ :s |  
  s  
  nextPutAll: 'Calvin';  
  nextPutAll: ' & ';  
  nextPutAll: 'Hobbes' ]
```

# Symbols

- #calvin is a symbol
- A kind of string
- Unique in the system
- Starts with #
  - #class #mac #at;put: #+ #accept:



# Symbols are Unique

Two symbols with the same representation points to the same object

```
#calvin == #calvin  
> true
```

Two strings with the same representation may be different objects depending on compiler optimisations



# Symbols vs. Strings

- A symbol is a read-only and unique object
- A string is a mutable object (for now)
- Symbols are used as method selectors
- Symbols are good candidates for identity based dictionaries (`IdentityDictionary`)



# What You Should Know

- Strings are collections of characters
- Symbols are unique immutable strings



A course by



and



in collaboration with



Inria 2020

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/>