Variables

Stéphane Ducasse and Damien Cassou

http://stephane.ducasse.free.fr/ stephane.ducasse@inria.fr

Lowercase for local variables

- temporary variables

```
| str |
str := 'Tiramisu'
```

- instance variables

```
Object subclass: #Point
  instanceVarNames: 'x y'
```

- method or block arguments

```
raisedTo: power

[ :x | ]
```

Special Variables

`self`, `super`, `thisContext`, `true`, `false`, `nil`

- `self`: the receiver of the message (this in Java).
- `super`: refers to the receiver but look method in the super class of the class defining the method.
- `thisContext` refers to the current stack.
- `true`, `false` are the Booleans
  - `true` is instance of `True`
  - `false` is instance of the class `False`
- `nil` is `UndefinedObject` instance

Shared or Global starts with Uppercase

- Transcript is an object that is globally accessible a kind of stdout

```
Transcript cr .
Transcript show: 'hello world'.
Transcript cr .
```

- Class `Object` is global

```
Object subclass: #Point
```

ClassVariables are Shared Variables

- How to share information between all the instances of a class and subclasses
- Use a classVariable

```
Object subclass: #CombinedChar
    instanceVariableNames: ’codes combined’
    classVariableNames: ’Compositions Decompositions Diacriticals’
    category: ’Kernel−BasicObjects’
```

- Here `Compositions` is shared between all the CombinedChar instances and instances of subclasses

Summary

- Lowercase are used for local/private/temporary variables
- Uppercase are used for shared or global variables