

```
module JeepsTalk where

--      \|||/
--      (o o)
-- ----oo0-(_)-0oo-----

-- # Jean-Pierre

-- ## Il faut que nous ayons une

-- # petite conversation

-- ## ... à propos des langages de

-- # programmation

-- ### -----
-- ### by Frédéric Peschanski ♀^•^?
-- ###(Sorbonne Université/LIP6)

--      \|||/
--      (o o)
-- ----oo0-(_)-0oo-----

-- ## Pour ton jubilée ...

-- Mon objectif initial était d'expliquer un de tes
-- travaux en lien avec la programmation.

--      \|||/
--      (o o)
-- ----oo0-(_)-0oo-----

-- ## Pour ton jubilée ...

-- Mon objectif initial était d'expliquer un de tes
-- travaux en lien avec la programmation.

-- ## ... Naturellement j'ai choisi

--      \|||/
--      (o o)
-- ----oo0-(_)-0oo-----
```

```
-- ## Pour ton jubilée ...  
-- Mon objectif initial était d'expliquer un de tes  
-- travaux en lien avec la programmation.
```

```
-- ## ... Naturellement j'ai choisi
```



```
-- # A Framework for  
-- # Object-Oriented Concurrent Programming  
-- yeah ! `(-_-)`
```

```
--      \|||/  
--      (o o)  
-- ----oo0-(_)-0oo-----
```

```
-- Il y a cependant un petit
```

```
-- ### .      |_-| o   _  
-- ### .      | | | | (_
```

```
--      \|||/  
--      (o o)  
-- ----oo0-(_)-0oo-----
```

```
-- Il y a cependant un petit
```

```
-- ### .      |_-| o   _  
-- ### .      | | | | (_
```

```
-- Je ne pratique (quasiment) plus
```

```
-- # Les Objets !  
-- ## .          -\_(ツ)_/-
```

```
--      \|||/
```

-- (o o)
-- ----oo0-(_)-0oo-----

-- Il y a cependant un petit
-- ### . |_-| o _
-- ### : | | | (_

-- Je ne pratique (quasiment) plus
-- # Les Objets !
-- ## . -_(ツ)_/-

-- (et en plus j'ai jamais vraiment compris SmallTalk)

-- # Interlude : Smalltalk !
-- (je vous préviens ... ça va mal finir ئ(-_-")ئ)

-- Je vous rappelle qu'en Smalltalk :

-- ## 1. Tout est objet
-- (par exemple a et b)

-- [[[image :type png :file "~/Recherche/jubilee-jeeps/jeepstalk/doc/smalltalk01.png" :width 300]]]

-- # Interlude : Smalltalk !

-- Je vous rappelle qu'en Smalltalk :

-- ### 1. Tout est objet
-- ## 2. Tout objet est instance d'une classe

-- [[[image :type png :file "~/Recherche/jubilee-jeeps/jeepstalk/doc/smalltalk02.png" :width 400]]]

-- # Interlude : Smalltalk !

-- Je vous rappelle qu'en Smalltalk :

-- ### 1. Tout est objet
-- ### 2. Tout objet est instance d'une classe
-- ## 3. La superclasse par défaut est Object

-- [[[image :type png :file "~/Recherche/jubilee-jeeps/jeepstalk/

```
doc/smalltalk03.png" :width 500)]]

-- # Interlude : Smalltalk !
-- Je vous rappelle qu'en Smalltalk :
-- ## 1. Tout est objet
-- ## 2. Tout objet est instance d'une classe
-- ### 3. La superclasse par défaut est Object

-- [[[image :type png :file "~/Recherche/jubilee-jeeps/jeepstalk/
doc/smalltalk04.png" :width 600]]]

-- # Interlude : Smalltalk !
-- Je vous rappelle qu'en Smalltalk :
-- ### 1. Tout est objet
-- ### 2. Tout objet est instance d'une classe
-- ### 3. La superclasse par défaut est Object
-- ## 4. La superclasse-classe par défaut est Class

-- [[[image :type png :file "~/Recherche/jubilee-jeeps/jeepstalk/
doc/smalltalk05.png" :width 700]]]

-- # Interlude : Smalltalk !
-- Je vous rappelle qu'en Smalltalk :
-- ### 1. Tout est objet
-- ### 2. Tout objet est instance d'une classe
-- ### 3. La superclasse par défaut est Object
-- ### 4. La superclasse-classe par défaut est Class
-- ## 5. Toute classe est instance de MetaClass

-- [[[image :type png :file "~/Recherche/jubilee-jeeps/jeepstalk/
doc/smalltalk06.png" :width 900]]]

-- # Interlude : Smalltalk !
-- Je vous rappelle qu'en Smalltalk :
```

```
-- ### 1. Tout est objet
-- ### 2. Tout objet est instance d'une classe
-- ### 3. La superclasse par défaut est Object
-- ### 4. La superclasse-classe par défaut est Class
-- ### 5. Toute classe est instance de MetaClass
-- ## 6. La classe de MetaClass est MetaClass class

-- [[[image :type png :file "~/Recherche/jubilee-jeeps/jeepstalk/doc/smalltalk07.png" :width 900]]]

-- # Interlude : Smalltalk !
-- Je vous rappelle qu'en Smalltalk :
-- ### 1. Tout est objet
-- ### 2. Tout objet est instance d'une classe
-- ### 3. La superclasse par défaut est Object
-- ### 4. La superclasse-classe par défaut est Class
-- ### 5. Toute classe est instance de MetaClass
-- ### 6. La classe de MetaClass est MetaClass class

-- ## ... et puis aussi ClassDescription, Behavior ...
-- [[[image :type png :file "~/Recherche/jubilee-jeeps/jeepstalk/doc/smalltalk08.png" :width 900]]]

-- # (-_-)ゞ

-- (revenons à nos moutons ʕ^聿^?)
```

 -- ... Il me faut un langage plus simple ...

```
-- (revenons à nos moutons ʕ^聿^?)
```

 -- ... Il me faut un langage plus simple ...

```
-- ## (-_-)ゞ
```

```
-- (revenons à nos moutons ʕ^聿^?)
```

 -- ... Il me faut un langage plus simple ...

```
-- ## (-_-) ``

-- # Haskell !

-- # eh bien oui, c'est plus simple, parce que :

-- ## - il a des monades !
-- (et les monades c'est juste des monoïdes dans la catégories des
endo-foncteurs,
-- c'est quoi le problème ?)

-- ## - il a des kinds d'ordre supérieurs H0K H0K !
-- (et du coup * -> * -> * peut avaler * -> * pour produire * !)

-- ## - il est paresseux !
-- (c'est bien pour les étudiants)

-- ## - il faut un PhD pour y comprendre quelque chose !
-- (ça tombe bien ...)

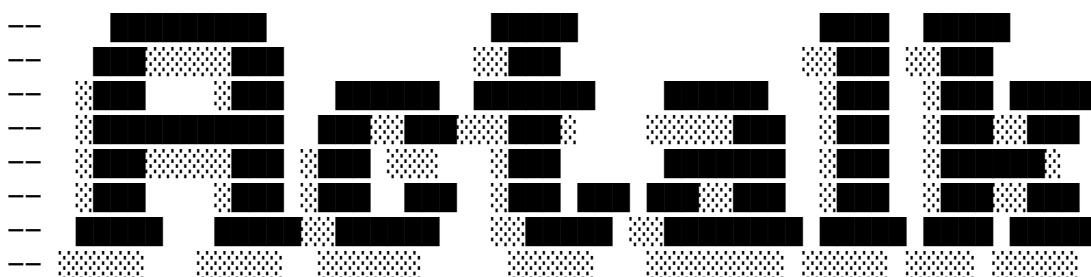
-- # c(○○○)८
```

```
--      \||/
--      (o o)
-- -----oo0-(_)-0oo-----
-- (donc)

-- ## Pour ton jubilée ...

-- Mon objectif initial était d'expliquer un de tes
-- travaux en lien avec la programmation.

-- ## ... Naturellement j'ai choisi
```



```
-- # A Framework for
-- # Object-Oriented Concurrent Programming
```

```
--          \|||/
--          (o o)
-- -----oo0-(_)-Ooo-----
-- # La check-list
-- ## Pour (ré-)implémenter Actalk (en version simpliste), il faut :
-- ### [X] Des objets, des classes et des méthodes
-- (enfin, y'a les typeclasses ...)
-- ### [✓] De l'«héritage»
-- (par ex: Monade «hérite» de Applicative qui «hérite» de Functor
```

ヽ(ーー)ゞ)

-- ### [X] Des envois de message (et leur réception)
-- (presque pareil : des appels de fonction)

-- ### Des variables [✓] et affectations [X]
-- (variables ≠ case mémoire)

-- ### [✓] Un runtime concurrent
-- (concurrent *et* parallèle !)

-- et puis j'ai aussi :

-- ### [✓] des fonctions
-- (en veux-tu en voilà !)

-- ### [✓] des types
-- (et pas n'importe lesquels!)

-- ### [✓] du peps !