

Programmation fonctionnelle en java

Si, si

Date: 14 juin 2013

Auteur: Damien GOUYETTE

Auteur



Développeur java/web > 10 ans
Indépendant depuis 2011 :-)



@CestPasDur



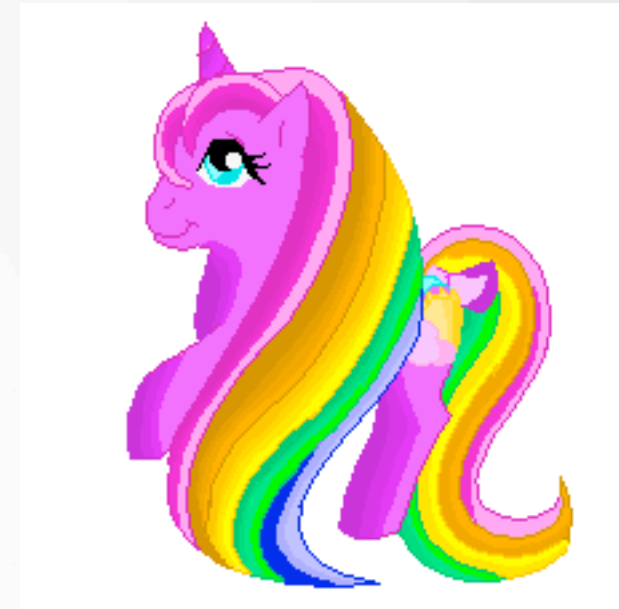
<http://www.cestpasdur.com>



Spoiler :

Il n'y aura pas :

- Suite de fibonacci,
- Calcul de nombre premiers,



Mais des exemples mis en place qui en prod

Après une première approche dans un langage dit fonctionnel Scala

Comment mettre en place dans son projet Java 6 (le java d'aujourd'hui ?

Librairies utilisées

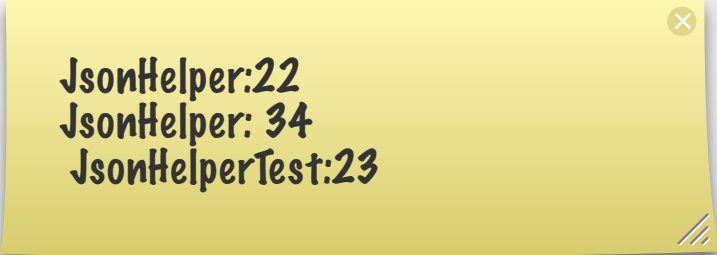
- Guava,
- FonctionnalJava
- LambdaJ

Avantages :

1. Découpage en fonctions facilement testables,
2. Code plus expressif,
3. Fonctions composables,
4. Adapté au multicoeur,

Fonction de premier ordre

- Répétable
- Ne modifie pas l'extérieur,
- Facilement testable
- Composable.



```
JsonHelper:22  
JsonHelper: 34  
JsonHelperTest:23
```

Prédicats

Retourne vrai ou faux, selon que l'expression contenue est respectée, ou non.

Ex : `RexgexPredicate` retourne vrai s'il y a correspondance à la regexp.



```
HomeMadeTalkService  
TalkHelper  
TalkHelperTest  
FunctionalTalkService
```


Type Option

Option : type polymorphique représentant l'encapsulation d'une valeur optionnelle. Il est utilisé en retour de méthode pour signifier que celle-ci retourne ou non une **valeur significative**.

Deux valeurs sont possibles :

- **None**, absence de valeur,
- **Some(valeur)**, présence de valeur



TalkDao:24
TalkResource:32

Type Either

Le type **Either** est l'encapsulation de l'appel à une méthode pouvant échouer.

Deux valeurs sont possibles :

- **Left(valeur)**, indique que le traitement a échoué,
- **Right(valeur)**, indique que le traitement a réussi.

TalkDao:24
TalkResource:32

En savoir +

[Code source sur github](#)

[Functional Programming Principles in Scala](#)

[Programmation fonctionnelle : utiliser le type Option](#)

[Programmation fonctionnelle : utiliser le type Either](#)



Questions ?

That's all for today

Date: 14 juin 2012

Auteur: Damien GOUYETTE