



Programme de Hackathon

Présentation.

Ce programme que nous voulons simple et riche doit définir les contenus nécessaires pour amener les participants à s'intéresser à Haskell. À la fin d'une session de Hackathon les apprenants devront être en mesure de comprendre et d'appliquer (de façon pratique) les notions introduites dans ladite session. Globalement il est question de préparer les participants à apprendre à implémenter les solutions aux problèmes concrets de la vie en utilisant le paradigme fonctionnel.

Profile des participants

Ce programme s'adresse à toute personne intéressée dans la programmation utilisant le paradigme fonctionnel. Il n'est exigé aucun prérequis pour participer à ce programme. Toutefois il est fortement conseillé à toute personne possédant au moins des connaissances en programmation.

Préparation des candidats

Pour être plus productif au cours de ce Hackathon, il est fortement conseillé de se préparer en apprêtant un certain nombre de choses:

- S'informer sur Haskell et la programmation fonctionnelle
- L'environnement de travail: (Installer Haskell et ses dépendances en se referant à l'adresse <https://www.haskell.org/downloads/>, et un éditeur de texte au choix)
- Familiarisation avec la plateforme replit que vous trouverez au lien <https://replit.com/>

Déroulement du programme

Le Hackathon se déroulera sur trois jours. Les deux premiers jours seront réservés aux cours théoriques et pratiques. Le troisième jour sera consacré à une séance de challenge. Le challenge consiste à traiter une épreuve de type QCM (Question À Choix Multiples) pour des challenges individuels et à implémenter des solutions à des problèmes qui seront posés



en utilisant le paradigme fonctionnel (principalement les notions acquises les deux premiers jours) pour des challenges par équipe.

Attribution des prix

Chaque candidat bénéficiera d'une prime de participation. Pour le challenge individuel et de groupe, les primes seront attribuées proportionnellement aux notes obtenues.

NB: Les primes seront en ADA (Crypto Monnaie) et seront transférées dans les portefeuilles électroniques des différents candidats

Le contenu de l'apprentissage

L'objectif étant de susciter l'intérêt à l'utilisation de Haskell, nous avons retenu à cet effet le contenu suivant: (Nous tenons en compte que le participant à déjà un environnement de travail bien installé)

1. Types
 - a. basics types
 - i. Bool
 - ii. Char
 - iii. ASCII Table
 - iv. Numbers
 - b. Constants and polymorphism
 - c. Some Operations on numbers
 - d. Comparison operators
 - e. Inference
 - f. List
 - i. String
 - ii. Generalization
 - iii. Some functions on list
 - iv. Texas range
 - v. List comprehension
 - vi. Tuples
 - g. Types variable
2. Functions
 - a. Function declaration and definition
 - b. Creating à single Function
 - c. Specifying the Function's Type
 - d. Conditional Expression
 - e. Pattern matching
 - f. Guards
 - g. Where
 - h. Let and In
 - i. Case expressions



wada



- j. Lambda expressions
 - k. Overview on recursion function
- 3. Higher order function
 - a. Curried functions
 - b. Some higher-orderism is in order
 - c. Advanced function on List
 - i. Map
 - ii. Filter
 - iii. Folder
 - d. Function application with \$
 - e. Function composition
- 4. Custom type
- 5. Typeclasses