

# Langages de scripts et Frameworks de développement Web

## 1. Programmation PPRP

Dans la NFIP (Nouvelle Identification Fictive des Protéines), une protéine est repérée par un identifiant qui se compose d'un bloc de 4 lettres en majuscules, d'un tiret puis de 4 à 8 chiffres. Ainsi ABCE-1234 et DDKZ-33722 sont des identifiants valides alors que U592 ne l'est pas.

- a) Ecrire une fonction PHP nommée `valideNfip` qui prend comme unique paramètre un identifiant possible et qui renvoie VRAI (au sens de PHP) si l'identifiant correspond aux règles présentées et FAUX sinon.
- b) Implémenter une fonction PERL équivalente.
- c) Implémenter une fonction RUBY équivalente.
- d) Implémenter une fonction PYTHON équivalente.

**Remarque :** dans tous ces langages, on utilisera une expression régulière pour tester la validité de l'identifiant.

Vous fournirez, si vous en êtes capable, le ou les tests unitaires susceptibles de prouver l'adéquation de votre code aux contraintes fournies.

Les réponses b), c) et d) sont indépendantes. Vous pouvez par exemple ne fournir que les réponses a), c) et d) ou seulement a) et c). Mais bien sûr pour avoir tous les points de l'exercice, il faut donner toutes les solutions.

## 2. Utilisation de Ruby et Rails pour la NFIPS

Votre chef n'est toujours pas convaincu de l'intérêt d'utiliser Ruby et Rails pour développer des applications Web pour l'entreprise. La passage obligé à la NFIPS (Nouvelle Identification Fictive des Patrons et Salariés) va vous en fournir l'occasion. Voici de quoi il s'agit. Actuellement, dans l'entreprise, tout est géré par des pages accessibles via la page de départ `menu.php`. Avec la nouvelle réglementation, tout poste informatique qui utilise un navigateur doit participer à la démarche qualité de traçabilité. Chaque utilisateur doit donc s'identifier avant d'accéder à la page `menu.php`.

Pour cela, il doit utiliser son nom et son code-utilisateur composé d'un bloc de 4 lettres en majuscules, d'un tiret puis de 4 à 8 chiffres. On admettra que la base de données `ENTRE` contient une table `UTILS` qui a entre autres pour champs `NOM` et `NFIPS` qui permettent de stocker respectivement le nom et une version cryptée du code-utilisateur. Pour cet examen, on supposera qu'il s'agit seulement du code-utilisateur.

Vous devez faire une démonstration de Ruby et Rails à votre chef. Donnez les instructions en ligne de commande et les explications nécessaires pour réaliser la démonstration suivante :

- créer une nouvelle application nommée `nfips` dont la partie interface se compose d'un panneau de saisie d'un champ `nom` et d'un champ `nfips` avec une table de données `ACCES` avec les champs `nom`, `nfips`, `poste`, `date`, `heure` qui enregistre chaque connexion ;
- insérer trois ou quatre utilisateurs dans `ACCES` à partir de `ENTRE`, si possible en `ORM` ;
- écrire un ou plusieurs tests avec `rspec` pour vérifier que lorsqu'un utilisateur arrive bien à s'identifier, il arrive à la page `menu.php` ;
- ajouter un champ `ALERTE` dans `ACCES` et le remplir si trois tentatives successives dans le temps sur un poste ont échoué en moins de 10 minutes, le remplissage se faisant dans le troisième enregistrement.
- créer un rapport hebdomadaire des alertes déclenchées et le proposer à la suite de la page de connexion si le code-utilisateur commence par `SECU` (premières lettres du mot sécurité) comme choix possible, l'autre choix possible étant bien sûr `menu.php`.

Au passage, vous expliquerez comment on pourrait détecter automatiquement le poste informatique utilisé.

Le code informatique et les explications que vous fournissez doivent permettre à votre chef de refaire la démonstration tout seul sur son poste où vous avez déjà installé tous les outils logiciels comme `rvm`, `ruby`, `rails`, `rspec`...

Il vaut mieux en écrire plus et bien détailler une étape que de vouloir traiter toutes les étapes de la démonstration, l'énoncé étant volontairement concis et succinct de façon à vous obliger à inventer de vous-même ce qui manque, à choisir une solution et à la documenter. Vous passerez sous silence tout ce qui concerne `git` et `heroku`.

### 3. Discussion sur l'ORM

Les frameworks récents bâtis sur `Python` et `Ruby` implémentent la technique de programmation nommée `ORM` et le concept de `MVC` comme méthodologie de développement. Imaginez que vous êtes un(e) développeur(e) confirmé(e) en `PHP` qui a déjà écrit ses propres fonctions pour gérer les bases de données `MySQL` dans votre entreprise. Vous devez convaincre votre chef qu'il ne sert à rien de réécrire tout votre code `PHP` pour implémenter l'`ORM` dans vos applications Web.

Rédigez pour cela une argumentation soutenue et motivée. Il est conseillé d'utiliser au moins trois mots de trois syllabes ou plus pour « transmettre un contenu rédactionnel fort ». Vous commencerez par citer les avantages principaux de l'`ORM` pour `Ruby` et `Python` avant de les réfuter point par point et au fur et à mesure. Une dizaine de lignes paraît être une rédaction minimale.