# Exonet : Gestion des inscriptions à un séminaire

## Description du thème

|  |  |
| --- | --- |
| Propriétés | Description |
| **Intitulé long** | Site de gestion des inscriptions à un séminaire en PHP |
| **Formation concernée** | Classes de terminale SIG |
| **Matière** | Systèmes d’information de gestion |
| **Thème** | L’information pour agir et décider |
| **Question de gestion** | La résolution de tous les problèmes de gestion est-elle automatisable ? |
| **Présentation** | A partir de l’observation de code fourni, l’élève est questionné sur différents points du programme associé au thème |
| **Notions** | Programmation, étude de code, variables, tests, boucles, fonctions |
| **Outils** | Serveur web, navigateur internet |
| **Mots-clés** | Site web, PHP, XML |
| **Durée** | 3 heures |
| **Auteur(es)** | Patrice Grand |
| **Version** | v 1.0 |
| **Date de publication** | 25 mars 2013 |

## Contexte

Le Centre de conférence de la ville de Poitiers accueille régulièrement des conférences ou des séminaires dont les thèmes sont variés. La proximité du Futuroscope permet aux participants de profiter de ce lieu de détente.

Un séminaire est prévu le 23 janvier autour des questions urbaines, il est ouvert aux acteurs des seules municipalités, qui peuvent venir de différents départements.

Ce séminaire, durant une seule journée, prévoit différentes conférences qui pour certaines ont lieu sur le même créneau horaire.

Un site est proposé aux représentants des municipalités qui pourront consulter le programme et s’inscrire aux conférences.

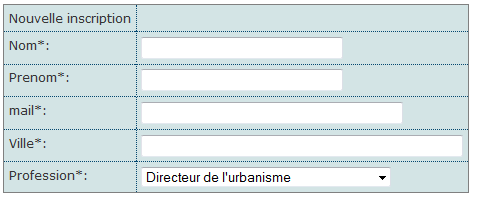
## Programme du séminaire

Lorsque le visiteur demande à voir le programme, le site propose le formulaire suivant :



## Inscription

Pour s’inscrire, le visiteur doit, dans un premier temps répondre à un questionnaire :

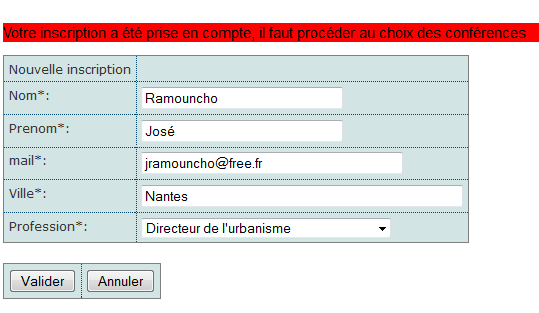


Un extrait du code PHP de ce formulaire d’inscription est fourni en **annexe 1**.

### Travail à faire

1. Donner la liste des variables utilisées et pour chacune, indiquer le type en justifiant votre réponse.

Un visiteur vient de saisir ses informations et valide son formulaire, il reçoit le formulaire suivant :

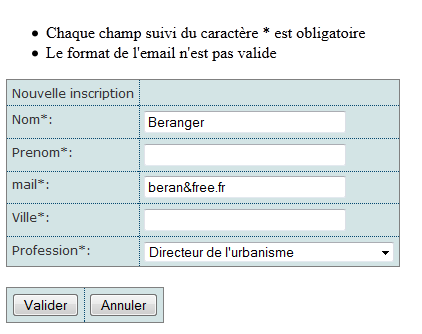


### Travail à faire

2.a A partir de **l’annexe 1**, énumérer les numéros des lignes d’instructions qui se sont exécutées pour produire ce formulaire.

2.b En déduire les conditions satisfaites (conditions if).

Un autre visiteur tente de s’inscrire :



### Travail à faire

3.a Combien d’erreurs ont été décelées ?

3.b A partir de **l’annexe 1**, énumérer les numéros des lignes d’ instruction qui se sont exécutées pour produire ce formulaire.

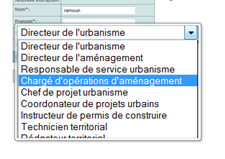
3.c En déduire les conditions satisfaites (conditions if).

L’**annexe 2** présente la suite du code qui permet de créer le formulaire avec ses éléments (zone de saisie, liste déroulante, etc.).

**Travail à faire**

4.a Réécrire la ligne 45 qui sera interprétée par le navigateur dans le cas de la saisie faite par le visiteur « Ramouncho ».

Le visiteur doit sélectionner sa profession dans une liste :



Cette liste contient 11 lignes (seules les 8 premières lignes sont visibles ici), le code de chargement de cette liste commence à la ligne 53 de **l’annexe 2**. La ligne 56 permet de récupérer le nombre de métiers à afficher en comptant le nombre de lignes du tableau retourné par la fonction *donnerLesProfessions*.

**Travail à faire**

5.a A quelle condition la ligne 61 s’exécute ?

5.b Réécrire cette ligne telle qu’elle sera interprétée par le navigateur.

## Sélection des conférences

Le visiteur inscrit peut ensuite procéder à la sélection de ses conférences :

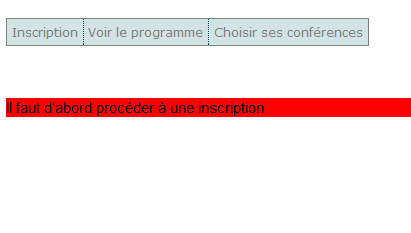


Le code principal de ce formulaire est présenté dans **l’annexe 3**.

**Travail à faire**

6.a Une variable *$inscrit* est utilisée à la ligne 13 ; quel est son type ? Justifier votre réponse.

Un visiteur sélectionne des conférences, sans être inscrit au préalable, il reçoit le message suivant :



**Travail à faire**

7.a Quels sont les numéros des lignes de code de **l’annexe 3** qui ont été exécutées ?

7.b Quelles sont, dans ce cas, les valeurs des variables *$action* et *$inscrit* ?

7.c Quelle est la valeur de la variable *$nbConferences* dans le cas ou $i vaut 1 ?

7.d Quelles sont les valeurs de $i et $j correspondant aux trois choix faits par le visiteur inscrit dans le formulaire de sélection des conférences présenté plus haut.

## Les fonctions utilisées

Plusieurs fonctions sont utilisées dans le code des formulaires. **L’annexe 4** présente les fonctions de gestion des erreurs de saisie.

**Travail à faire**

8.a Comment reconnait-on l’utilisation (l’appel) de fonctions dans le code ?

8.b Combien de fonctions sont utilisées dans l’**annexe 1** ?

8.c Combien d’arguments prend la fonction appelée à la ligne 19 de **l’annexe 1** ?

8.d Retourne-t-elle une valeur ? Justifier votre réponse.

8.e Pourquoi cette fonction ne prend pas la profession comme argument ?

8.f Combien d’arguments prend la fonction appelée à la ligne 20 de **l’annexe 1** ? Retourne-t-elle une valeur ? De quel type ? Justifier votre réponse.

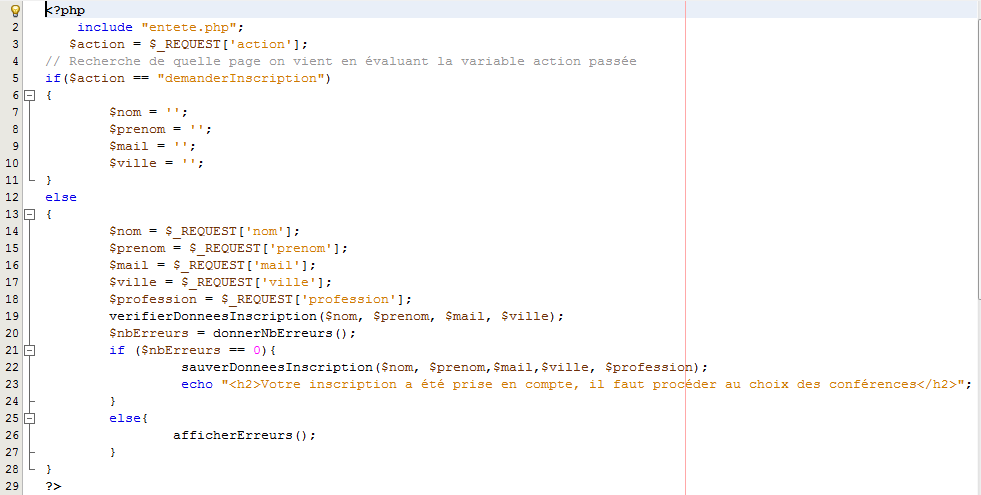
## Nouvelle règle de gestion

La version actuelle ne permet pas de voir le nom de l’intervenant au moment des sélections des conférence, ce qui peut gêner les inscrits qui déterminent parfois leur choix en fonction de l’intervenant. Il est décidé de procéder à une modification afin de répondre à cet objectif. L’**annexe 5** présente la version attendue de cette évolution.

**Travail à faire**

9. Modifier le code afin de répondre à cette nouvelle règle de gestion.

**Annexe 1 : Début du code du formulaire d’inscription**



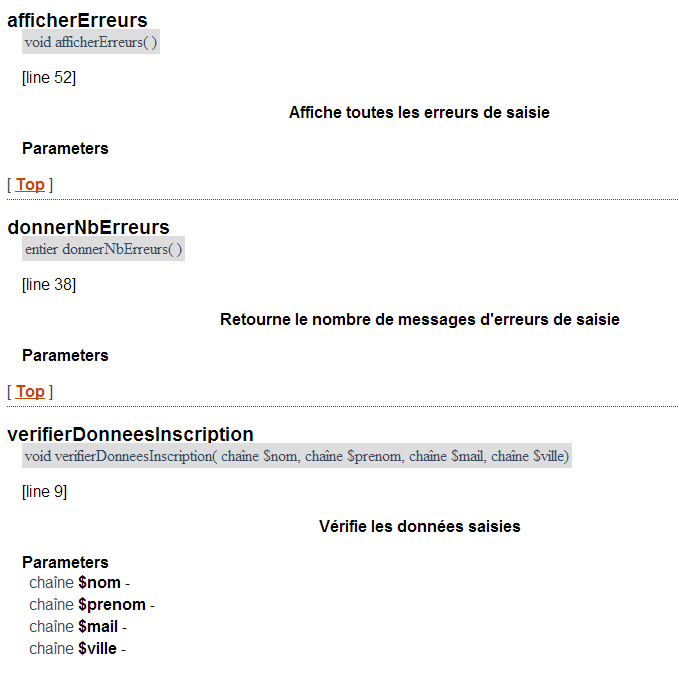
**Annexe 2 : Suite du code du formulaire d’inscription**



**Annexe 3 : Extrait du code de sélection des conférences**



**Annexe 4 : Fonctions de gestion des erreurs**



**Annexe 5 : Evolution du site**

