Suivi des modifications

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Référence** | **Auteur** | **Date** | **Commentaires** |
| A | 1\_20231122\_phpCodeSniffer | Allan, Hugo B, Hugo P, Alexis, Najib | 24/11/2023 | Création |
|  |  |  |  |  |
|  |  |  |  |  |

Objet :

Ce document est une documentation expliquant ce qu’est phpCodeSniffer, à quoi il sert et comment s’en servir.

Diffusion :

BTS SIO – Hovhannessian S.

BTS SIO – Elèves BTS SIO option SLAM

Développement :

Table des matières

[Introduction 1](#_Toc151671696)

[PhpCodeSniffer, qu’est-ce que c’est ? 2](#_Toc151671697)

[Ok, mais à quoi ça sert ? 2](#_Toc151671698)

[Fonctionnement 2](#_Toc151671699)

[Installation de PhpCodeSniffer 3](#_Toc151671700)

[Auto-Évaluation 10](#_Toc151671701)

[Évaluation de la formation 11](#_Toc151671702)

# Introduction

Le but de cette documentation est t’expliquer ce qu’est phpCodeSniffer et de t’apprendre à l’utiliser. Nous te proposons donc ce tutoriel afin que tu te familiarise avec ce nouvel outil et que tu sois toi-même capable de l’utiliser.

Après avoir suivis l’ensemble de cette documentation tu seras capable :

* Comprendre ce qu’est phpCodeSniffer
* Installer l’outil
* L’intégrer à l’environnement PhpStorm
* Utiliser cet outil
* Former tes collègues sur l’ensemble des connaissances que tu auras maîtrisés

**C’est partis !**

# PhpCodeSniffer, qu’est-ce que c’est ?

PHP\_CodeSniffer, également connu sous le nom de PHPCS, est un outil de vérification de la qualité du code source PHP. Il analyse le code pour détecter les violations des normes de codage prédéfinies et produit des rapports détaillés sur les problèmes trouvés.

# Ok, mais à quoi ça sert ?

* **Respect des normes de codage :** PHP\_CodeSniffer permet de définir des normes de codage spécifiques (telles que PSR-2, PSR-12, etc.) que les développeurs doivent suivre. Cela contribue à maintenir une cohérence dans le style du code au sein d'un projet ou d'une communauté de développement.
* **Détection d'erreurs potentielles :** Il peut identifier des erreurs de programmation potentielles, telles que des variables non utilisées, des espaces superflus, des indentations incorrectes, etc.
* **Amélioration de la lisibilité :** L'outil encourage l'écriture de code plus lisible en signalant les portions de code difficiles à comprendre ou à suivre.
* **Facilitation de la maintenance :** En utilisant PHP\_CodeSniffer, les équipes de développement peuvent s'assurer que le code est conforme aux normes établies, facilitant ainsi la maintenance du code à long terme.

# Fonctionnement

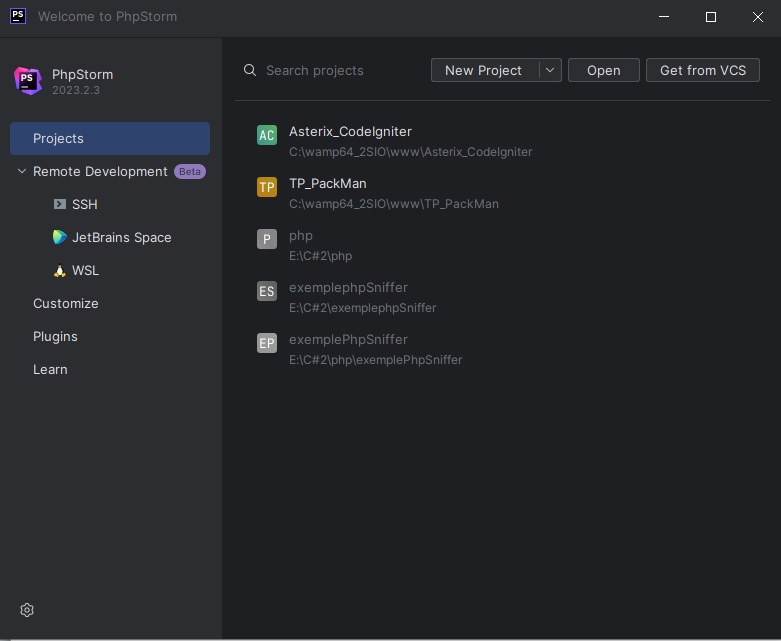
* **Fichiers de règles :** PHP\_CodeSniffer utilise des fichiers de règles pour définir les normes de codage. Ces règles spécifient la manière dont le code doit être formaté et organisé.
* **Analyse statique :** L'outil effectue une analyse statique du code source, ce qui signifie qu'il examine le code sans l'exécuter. Il ne détecte que des problèmes liés à la structure et à la syntaxe du code.
* **Rapports de sortie :** Après l'analyse, PHP\_CodeSniffer génère des rapports détaillés indiquant les violations des normes détectées. Ces rapports aident les développeurs à identifier et à corriger les problèmes.
* **Intégration dans le processus de développement :** PHP\_CodeSniffer peut être intégré dans des outils tels que des éditeurs de texte, des systèmes de gestion de versions ou des plates-formes d'intégration continue, facilitant ainsi l'intégration de la vérification des normes dans le processus de développement.

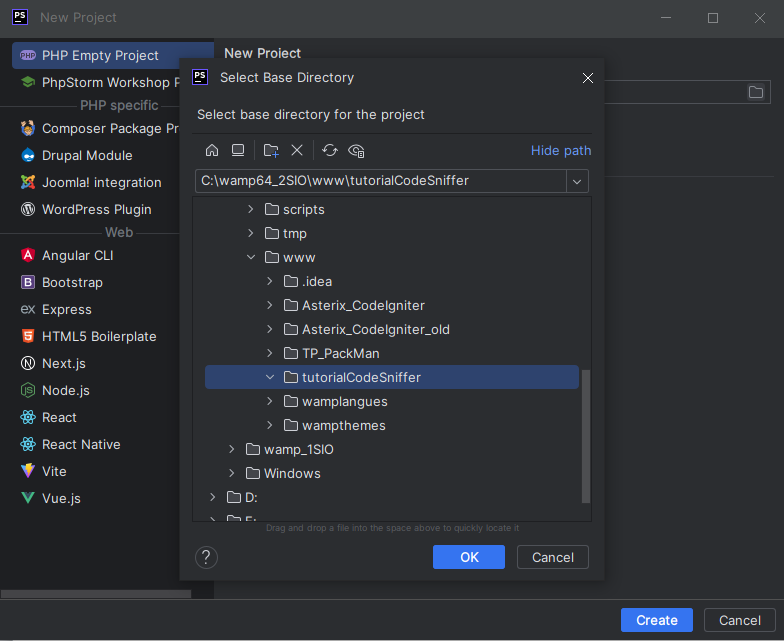
# Installation de PhpCodeSniffer

Voici un tutoriel étape par étape permettant d’installer PhpCodeSniffer sur l’environnement de travail PhpStorm.

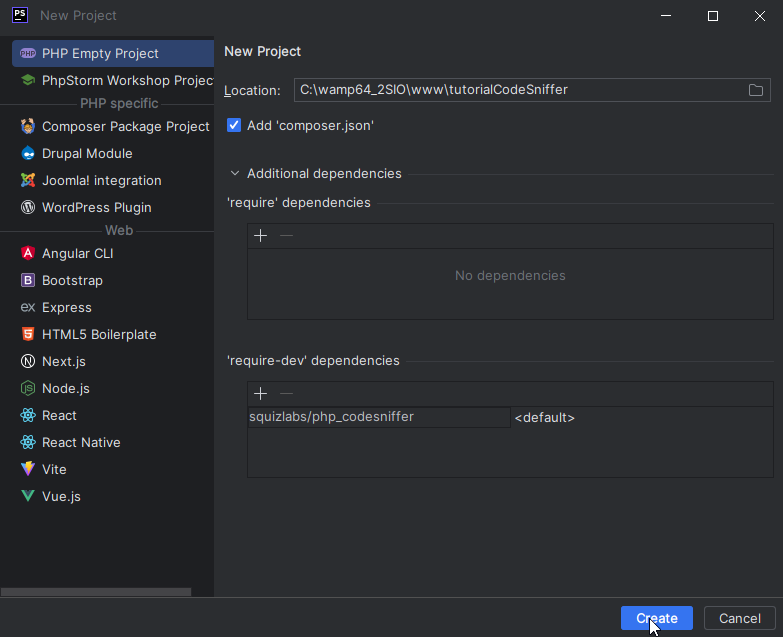
**Etape 1 :**

Créer un nouveau projet php sur phpstorm à la racine du site web. Donner lui le nom que vous voulez. Ici, dans notre exemple nous l’avons appelé « tutorielCodeSniffer ».



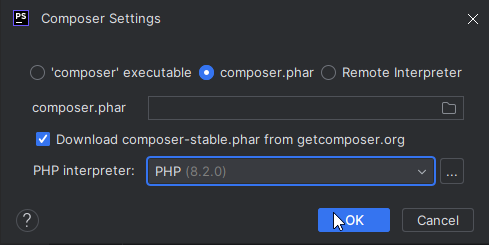


Pensez à bien cocher la case « Add ‘composer.json’ » avant de cliquer sur le bouton « Create ».



**Etape 2 :**

Dans le fichier « composer.json », cliquez sur « install » afin d’installer composer sur phpstorm. Vérifiez que le bouton composer.phar est cocher et que la dernière version de PHP (ici 8.2.0) est bien sélectionner dans la liste déroulante associé à la section PHP interpreter.



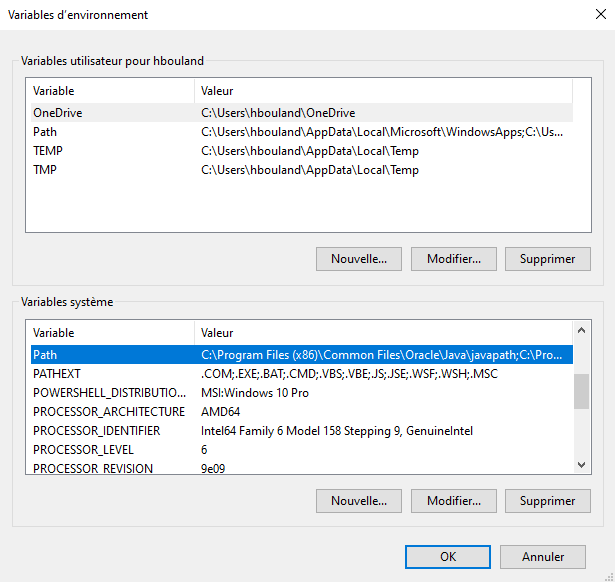
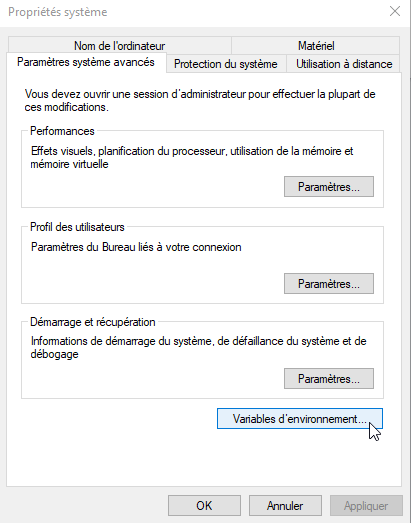
**Etape 3 :**

Une fois cette étape faite, PhpCodeSniffer devrait être installer. Dans ce cas passez directement à l’étape 7.

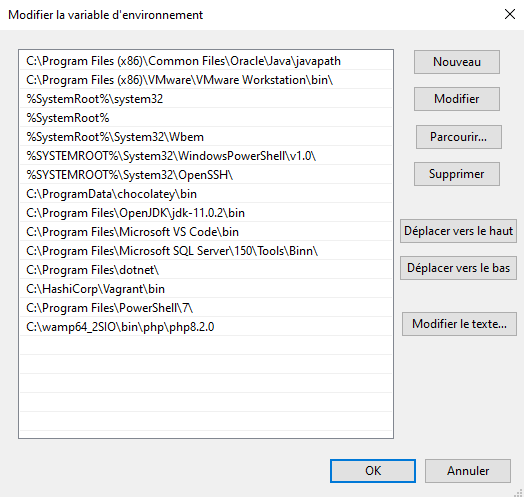
Cependant il se peut qu’un message d’erreur apparaisse, vous indiquant que le dossier contenant la version de php n’a pas étais trouver. Il s’agit d’une erreur propre à Windows lier à une variable d’environnement manquante dans le dossier Path de Windows. Dans ce qu’à suivez les étapes suivantes.

Depuis le menu windows, aller sur « modifier les variables environnement systèmes » et cliquez sur « Variables d’environnement… »

Puis aller sur les variables d’environnement, double cliquez sur « Path ».



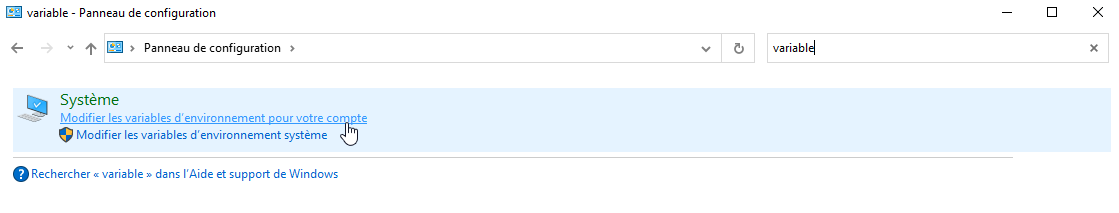
Enfin, ajouter le chemin de votre fichier php se trouvant à la racine du site web.



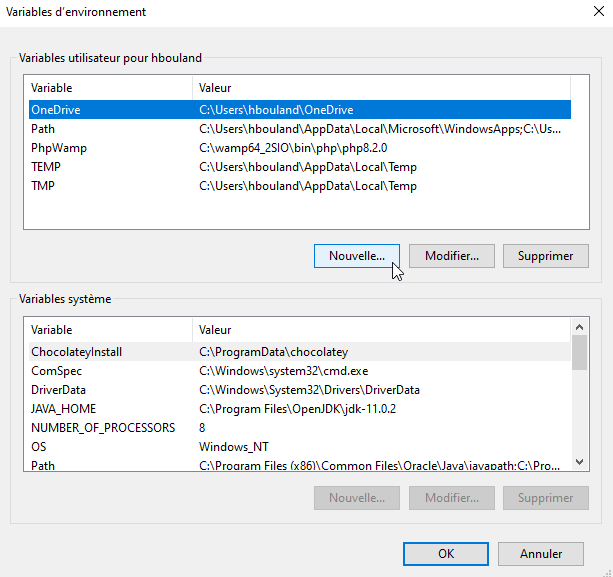
**Etape 4 :**

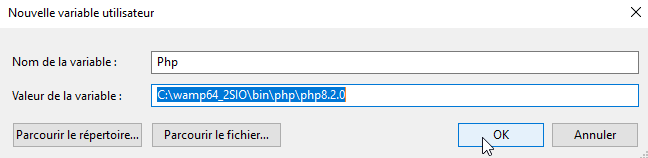
Aller sur panneaux de configuration, puis aller sur « modifier les variables d’environnement systèmes ».

Après cela, aller sur « Modifier les variables d’environnement pour votre compte ».



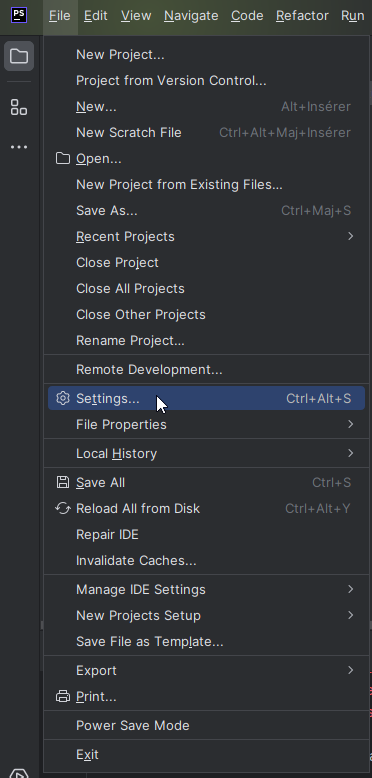
Puis, dans les variables utilisateur pour votre compte, selectionnez « nouvelle… » Et ajouter le chemin de votre fichier php se trouvant à la racine du site web.



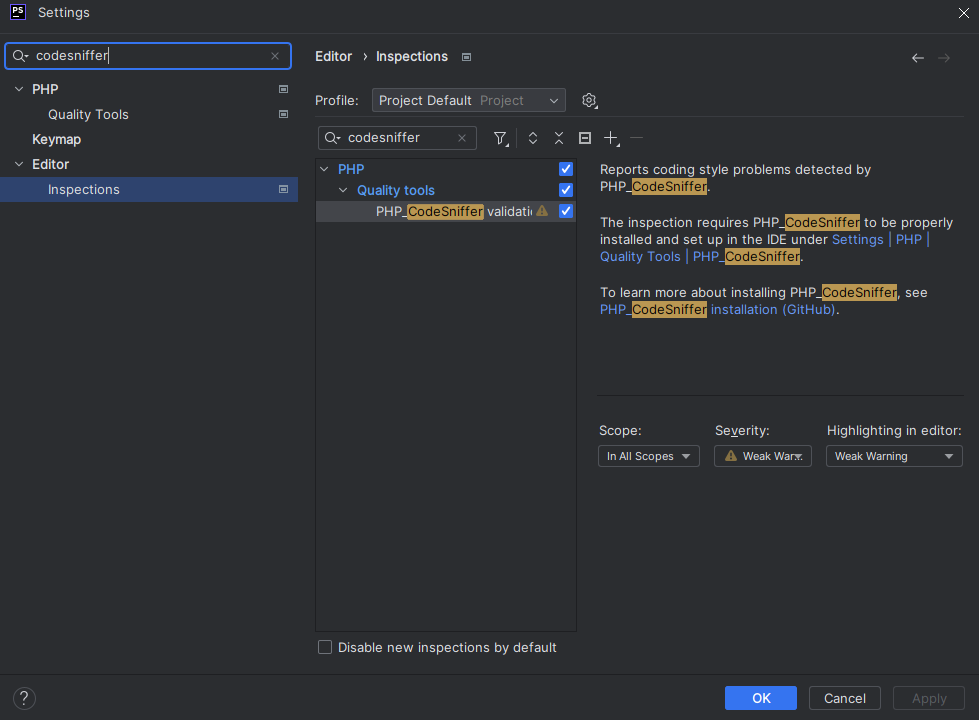


**Etape 5 :**

Aller dans les options.



Aller sur « Editor » puis « Inspections » et cherchez « Php\_codeSniffer » afin de cocher « PHP\_CodeSniffer validation ».

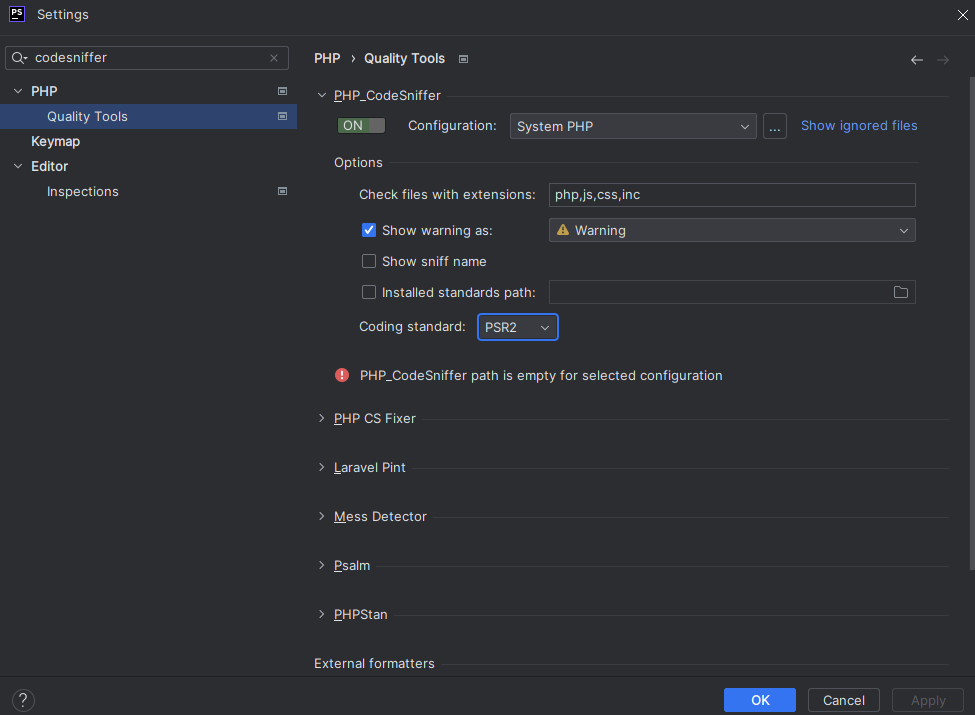


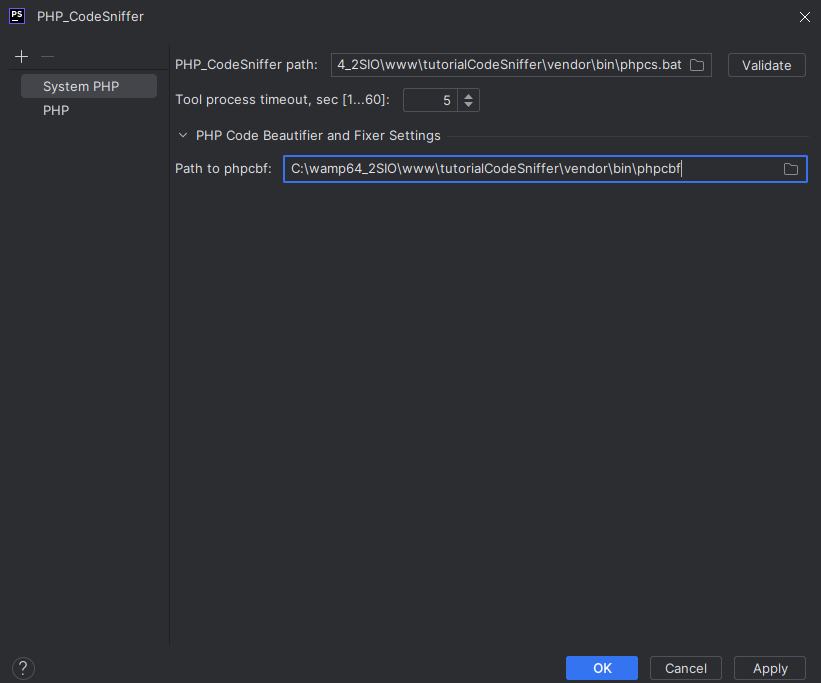
**Etape 6 :**

Toujours dans les options, aller sur « Php » puis « quality tools ».

Appuyez sur « off » afin de lancer PhpCodeSniffer, puis appuyez sur « … » à côté de « System PHP ».

Ensuite, ajoutez le chemin de PHP codeSniffer ainsi que le chemin du phpcbf ce trouvant dans le fichier php créer au début

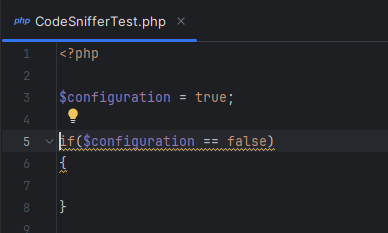




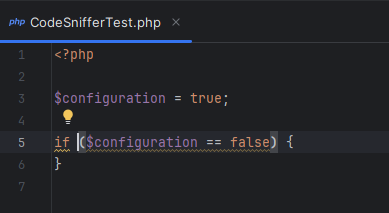
**Etape 7 : Vérification**

Après avoir appliquez les modification

Exemple test : AVANT



Exemple test : APRES



# Auto-Évaluation

Voici un tableau permettant de t’évaluer. Coche les cases qui corrspondent à ton niveau de compréhension actuelle. Cela te permettra d’avoir une vue d’ensemble sur tes connaissances et de savoir quelles étapes tu à moins compris pour pouvoir les retravailler. N’hésite pas à demander à ton tuteur en cas de besoin, il est là pour ça ;)

A toi met une croix ‘X’ dans les cases correspondante !

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| Comprendre le fonctionnement de phpCodeSniffer | X |  |  |  |  |
| Configuration de l'interpréteur PHP dans PHPStorm |  |  |  |  |  |
| Ajouter une variable d’environnement Windows (si besoin) |  |  |  |  |  |
| Installation de PhpCodeSniffer |  |  |  |  |  |
| Utiliser phpCodeSniffer |  |  |  |  |  |

# Évaluation de la formation

Voici un autre tableau similaire ! Celui-là te permet d’évaluer le niveau global de la formation. Cela permettra à ton tuteur de savoir les points à améliorer pour les prochaines formations. Alors n’hésite pas à noter toutes tes impressions qu’elles soient positives ou non, cela nous aidera grandement.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| Clarté des explications founis dans la documentation | X |  |  |  |  |
| Pertinence des informations apportés |  |  |  |  |  |
| Accessibilité de la formation (étais elle difficile à suivre ou non) |  |  |  |  |  |
| Durée de la formation (la durée était-elle correct, ou trop longue ?) |  |  |  |  |  |
| Qualité des explications données par le formateur |  |  |  |  |  |
| Utilité de la formation |  |  |  |  |  |
| Qualité générale de la formation |  |  |  |  |  |

Enfin si tu as des remarques ou des pistes d’améliorations n’hésite pas à les noter ici.

Remarques :

Merci d’avoir suivis notre formation jusqu’au bout, nous espérons qu’elle t’a plus. Nous espérons te revoir pour une prochaine formation ! ☺