Bonnes pratiques générales :

* Commenter régulièrement
* Sauvegarder régulièrement
* Ecrire de façon claire et lisible
* Eviter les accents et les espaces dans les noms de fichiers.
* Utiliser des noms de variables logiques.

JAVA :

* Regrouper les classes par packages selon des critères.
* Faire des fichier assez courts (moins de 2000 lignes par fichier source).
* Mettre un commentaire dans chaque début de fichier
* La première ligne de code du fichier devrait être une clause package qui indique à quel paquetage appartient la classe.
* Chaque code devrait être documenté pour aider à la compréhension ou à s’y retrouver.
* Bien déclarer les variables.

Javascript

* Utiliser une notation simplifiée
* Balise <script> doit être déclarée en fin de page html
* Utiliser l’opérateur de comparaison
* Utiliser « For in » au lieu de « eval » ou « with »
* Utiliser l’opérateur ternaire
* Tester son code et sa qualité avec un testeur (comme JSLint).

Python :

* Utiliser une bonne syntaxe
* Nommer ses variables uniquement avec des lettres (minuscules) et des sous tirets.
* Pour gérer les espaces, il est conseillé d’entourer ses opérateurs d’un espace avant et après l’opérateur.
* Une ligne de code ne doit pas dépasser 79 caractères pour améliorer la lisibilité.
* Les lignes vides sont utiles pour séparer les différentes parties du code.
* Commenter régulièrement son code et modifier les commentaires si le code change.
* Contrôler la qualité de son code.

C :

* Il est recommandé d’utiliser un éditeur de texte pouvant remplacer les tabulations par des espaces.
* Placer ses commentaires au-dessus de la portion de code à commenter.
* Ne pas utiliser trop de commentaires : utiliser l’auto-documentation.
* Il est recommandé d’utiliser des minuscules et des suffixes comme \_f pour fonction par exemple.
* Protéger le fichier d’en-tête des inclusions multiples dans la même unité de compilation.
* Le fichier d’en-tête ne doit contenir que le strict nécessaire.

PHP :

* Les accolades sont à la ligne suivant la déclaration d’une classe, d’une fonction ou d’une méthode.
* Une fonction doit toujours retourner le même type de résultat : si une fonction retourne plusieurs choses, il faut favoriser l’utilisation d’un tableau.
* Un tableau ne doit stocker qu’un seul type d’informations : il faut utiliser un tableau associatif pour stocker plusieurs types d’informations.
* Les noms des variables, des fonctions, des méthodes et des classes doivent être le plus explicite possible.
* Favoriser les quotes simples aux doubles et favoriser la concaténation.
* Limiter le nombre de paramètres d’une fonction.

HTML/CSS :

* Insérer le reset CSS pour réduire considérablement les erreurs d’incompatibilité (comme normalize.css).
* Organiser les éléments de sa feuille de style.
* Respecter les conventions d’écritures pour les IDs et les classes.