**Python**

Règle de nommage

* Utilisez les conventions de nommage précisées précédemment.
* Pour différencier les classes des fonctions, commencez vos fonctions par une minuscule puis séparez les espaces par « \_ » et vos classes par une majuscule suivie de majuscules. (Exemple : class BonjourLesSid ; def bonjour\_les\_sid)
* Précédez les fonctions/classes protected par « | » et celles private par « \_ ». (Exemple : def |bonjour\_les\_sid (protected) ; def \_bonjour\_les\_sid (private))

Ecriture du code

* Séparez les méthodes par une ligne blanche. Séparez les classes par deux lignes blanches.
* Expliquez les méthodes/classes par une ou plusieurs lignes de commentaire.
* Préférez l'utilisation de la programmation fonctionnelle dans vos scripts. Il s'agit d'utiliser le plus possible des fonctions pour les actions qui sont répétées plusieurs fois
* Assurez-vous qu’une fonction retourne quelque chose dans tous les cas. Attention que le seul «return» ne se situe pas dans un «if» dont la condition n’est pas toujours vérifiée.
* Si une méthode parait longue, n’hésitez pas à sauter des lignes entre gros blocs de code pour rendre la méthode plus lisible. Attention à ne pas en abuser et rendre le code illisible.
* Mettez un espace avant et après un « = ». Exemple : bonjour = 3

Importation des modules

* Préférez des chemins relatifs. Exemple : import bonjour (A faire) ; from bonjour import \* (A ne pas faire)
* Mettez un espace entre les différents imports.

Autre

* Avec Spyder, vous pouvez afficher des alertes lorsque votre code viole une directive PEP8. Pour les activer, allez dans Outils -> Preferences -> Editeur -> Introspection et analyse de code et cochez la case à côté de Real-time code sytle analysis (PEP8).