**Instructions Formation LI248 – Exercices & Travaux pratiques**

1. **Qu'est Python?** 
   1. Atelier : Mise en place d'un environnement de développement

Voir la slide correspondante

1. **Maîtriser la syntaxe de base** 
   1. Interprétation directe et en script
      1. Console

Ouvrez une console Python

Taper les expressions suivantes :

a = 10

b = a + 2

print(a)

print(b)

print(‘Calculs Faits’)

Que se passe-t-il dans votre console ?

* + 1. Ecrivez les instructions de la section précédente dans un fichier nommé **script.py** et dans une invite de commande (Windows) ou un shell, tapez python script.py

(Chemin à adapter selon l’emplacement du fichier)

Que se passe-t-il dans votre invite de commande / shell ?

* 1. Les variables, les opérateurs, les expressions
     1. Opérateurs sur les entiers (5-10 minutes de manipulation)

Définissez des variables de type Entier avec des valeurs quelconques et tester les différents opérateurs. Afficher le résultat.

Exemple :

a = 5

b = 10

print(a \* b)

* + 1. Opérateurs sur chaines de caractères

Définissez des chaines de caractères et explorer les différents possibilités des opérateurs présentés dans la slides.

Introduire la définition d’une chaine de caractères avec les simples quotes ou les doubles quotes.

* + 1. Opérateurs de conversion de types

Reprendre l’exemple de la slide et la faire fonctionner.

Tester la même solution avec un nombre flottant.

Testons la conversion d’un nombre exprimé sous forme de chaine en un nombre

* + 1. Les expressions

Faire recopier et exécuter les exemples de code des slides

* 1. Les tests et les boucles
     1. Les tests

Faire recopier et exécuter les exemples de code des slides

Proposer aux stagiaires d’implémenter leur propre algorithme ou bien un exemple de groupe (décision de l’algo et résolution en groupe)

* + 1. Les boucles

Faire recopier et exécuter les exemples de code des slides

Proposer aux stagiaires d’implémenter leur propre algorithme ou bien un exemple de groupe (décision de l’algo et résolution en groupe)

* 1. Les opérations avancées
     1. Faire réécrire et exécuter les exemples des slides + Discussion
  2. Atelier : Multiples algorithmes pour maîtriser la syntaxe de base
     1. Ecriture de l’algorithme FizzBuzz décrit dans la slide

1. Comment structurer son code
   1. Définir une fonction

Ecrire une fonction qui permet de calculer le prix soldé d’un article avec en paramètre le prix de l’article ainsi que le pourcentage de rabais.

Ecrire un script qui appelle cette fonction pour plusieurs types d’article et calculer le prix total sans rabais, le montant total des rabais ainsi que le montant total avec rabais et renvoyer les résultats.

Faire une version avec des arguments positionnels, puis nommés, puis variables nommés.

* 1. Atelier : Opérations sur les chaînes de caractères par des fonctions
     1. Voir les instructions dans la slide dédiée

1. Les algorithmes de bases
   1. Les représentations graphiques
      1. Faire écrire par chaque stagiaire un algorithme avec la signalétique standardisée et la faire présenter par chaque stagiaire.
   2. Les boucles prévisibles et imprévisibles
      1. Ecrire un script avec le code de la slide.
   3. La fonction range
      1. Manipuler la fonction range pour écrire :
         1. La liste des entiers de 0 à 10 inclus par pas de 1
         2. La liste des entiers pairs de 0 à 15
         3. La liste des entiers impairs de 0 à 15
   4. La récursivité
   5. Atelier : Écriture en Python d'algorithmes courants
2. La structuration de données
   1. Comment choisir sa structure
   2. Comment simplifier son algorithme
   3. Atelier : Écriture en Python d'algorithmes plus étendus
3. Les calculs Les données temporelles
   1. Manipulation des entrées et sorties Les calculs scientifiques L'essentiel du Python au quotidien
4. Les bases de la programmation orienté objet
   1. Les apports de l'objet

Chaque stagiaire devra présenter un exemple de modélisation objet en sous-lignant les 3 composantes de l’apport de l’objet.

* 1. Instanciation, destruction, encapsulation, agrégation
     1. Création Destruction

Faire la définition d’une classe avec un constructeur / Créer plusieurs instances et en détruire certaines. Que se passe-t-il quand on appelle une fonction membre d’un objet quand l’objet vient d’être crée / détruit ?

* + 1. Encapsulation agrégation

Chaque stagiaire écrit une classe sur un thème de son choix pour démontrer l’encapsulation et l’aggrégation.

* 1. Polymorphisme et introspection
     1. Faire créer une hiérarchie de classe sur un thème : véhicule, animaux, plantes, etc

Créer des objets à partir des différentes classes et appeler les différentes méthodes d’introspection pour comprendre ce qu’ils font.

* 1. Atelier : Conception d'une application et implémentation Objet
     1. Voir slide