

Cours de Programmation Python

Enseignant:

Ndombasi Diakusala Joao André, M.Eng. AI

<njadnissi@gmail.com>

Syllabus de Fondamentaux de la programmation en Python en 30 heures:

1. **Heure 1-2 : Introduction à Python et Environnement de Développement (2h)**
2. Qu'est-ce que Python ? Histoire et applications
3. Pourquoi apprendre Python ? Avantages
4. Installation de Python (Windows, macOS, Linux)
5. Installation et configuration d'un IDE (VS Code recommandé)
6. Premier programme Python : "Hello World!"
7. Commentaires en Python
8. **Heure 3-4 : Les Bases du Langage : Variables et Types de Données (2h)**
9. Déclaration et affectation de variables
10. Types de données numériques : entiers (int), nombres à virgule flottante (float)
11. Opérations arithmétiques de base
12. Les chaînes de caractères (str) : définition, concaténation
13. Conversion de types (casting)
14. Opérateurs d'assignation
15. **Heure 5-6 : Structures de Données Simples : Listes et Tuples (2h)**
16. Les listes (list) : définition, accès aux éléments, modification
17. Méthodes de liste courantes (ajouter, supprimer, trier, etc.)
18. Les tuples (tuple) : définition, immutabilité, accès aux éléments
19. Différences et cas d'utilisation entre listes et tuples
20. Boucles for avec listes et tuples (introduction)
21. **Heure 7-8 : Structures de Données Avancées : Dictionnaires et Ensembles (2h)**
22. Les dictionnaires (dict) : définition, paires clé-valeur, accès et modification
23. Méthodes de dictionnaire (ajouter, supprimer, parcourir)
24. Les ensembles (set) : définition, opérations d'ensemble (union, intersection)
25. Cas d'utilisation des dictionnaires et des ensembles
26. Quand utiliser chaque structure de données
27. **Heure 9-10 : Opérateurs et Expressions (2h)**
28. Opérateurs de comparaison (égal à, différent de, plus grand que, etc.)
29. Opérateurs logiques (and, or, not)
30. Priorité des opérateurs
31. Expressions booléennes
32. Raccourcis d'assignation (+=, -=, etc.)
33. **Heure 11-13 : Structures de Contrôle : Conditions (3h)**
34. L'instruction if
35. L'instruction if-else
36. L'instruction if-elif-else
37. Conditions imbriquées (nested conditionals)
38. L'opérateur ternaire (expressions conditionnelles)
39. Gestion des cas multiples : match-case (Python 3.10+)
40. **Heure 14-16 : Structures de Contrôle : Boucles (3h)**
41. La boucle for : itérer sur des séquences (listes, chaînes, range())
42. La boucle while : conditions d'arrêt
43. break et continue : contrôler le flux des boucles
44. Boucles imbriquées (nested loops)
45. Compréhensions de listes (introduction)
46. **Heure 17-19 : Fonctions (3h)**
47. Définir une fonction (def)
48. Appeler une fonction
49. Paramètres et arguments de fonction
50. Valeurs de retour (return)
51. Paramètres par défaut
52. Arguments nommés (keyword arguments)
53. Portée des variables (locale et globale)
54. **Heure 20-21 : Modules et Packages (2h)**
55. Qu'est-ce qu'un module ?
56. Importer des modules (import)
57. Modules standards (ex: math, random, datetime)
58. Créer ses propres modules
59. Qu'est-ce qu'un package ? (introduction)
60. Installation de packages avec pip
61. **Heure 22-23 : Gestion des Erreurs et Exceptions (2h)**
62. Types d'erreurs en Python (SyntaxError, TypeError, NameError, etc.)
63. Le bloc try-except
64. Capturer des exceptions spécifiques
65. Le bloc else dans try-except
66. Le bloc finally
67. Lever des exceptions (raise)
68. **Heure 24-26 : Manipulation de Fichiers (3h)**
69. Ouvrir et fermer des fichiers (open(), close())
70. Modes d'ouverture de fichiers (lecture, écriture, ajout)
71. Lire des fichiers texte (read(), readline(), readlines())
72. Écrire dans des fichiers texte (write(), writelines())
73. Le gestionnaire de contexte with open(...)
74. Traitement de fichiers CSV (introduction)
75. **Heure 27-28 : Programmation Orientée Objet (POO) - Concepts de Base (2h)**
76. Introduction à la POO : Classes et Objets
77. Définir une classe
78. Le constructeur \_\_init\_\_
79. Attributs d'instance
80. Méthodes d'instance
81. Héritage (introduction)
82. **Heure 29-30 : Projets Pratiques et Révision (2h)**
83. Mise en pratique des concepts appris via des mini-projets :
84. Calculatrice simple
85. Jeu de devinettes
86. Gestionnaire de contacts simple (avec fichiers)
87. Bonnes pratiques de codage (PEP 8)
88. Ressources pour aller plus loin (documentation, communautés)
89. Séance de questions/réponses et révision des points clés