

Raspberry Pi 3, premier contact

1. Introduction au Raspberry Pi	7
2. Démarrage de Raspbian	11
2.1 Historique	11
3. Comprendre l'écosystème Python : quelle version utiliser ?	16
4. Installer des bibliothèques Python	17
4.1 La méthode aptitude	18
4.2 La méthode pip	19
4.3 pip ou aptitude ?	21
5. IDLE : l'éditeur de code en Python pour Python	22
6. Conclusion	27

Python : bases et concepts avancés

1. Hello World	29
2. Les types de base : int, float, str et bool	30

2.1 L'entier : int	30
2.2 Le flottant : float	31
2.3 La chaîne de caractères : str	31
2.4 Le booléen	35
3. Les structures de données : list, dict, tuple	35
3.1 La liste : list	35
3.2 Les tuples	36
3.3 La table de hachage ou dictionnaire : dict	37
3.4 L'ensemble : set	38
4. Les instructions, conditions et boucles	38
4.1 La condition if	39
4.2 La condition else	39
4.3 La boucle for	40
4.4 L'instruction break	40
4.5 L'instruction continue	41
4.6 La boucle while	41
5. Les opérateurs	42
5.1 Opérateurs arithmétiques	42
5.2 Opérateurs logiques	

5.3 Opérateur d'appartenance	45
5.4 Opérateur d'affectation	45
	46
6. Les classes : définition avec le mot-clé class	47
6.1 Premiers pas	48
6.2 Exposer les attributs d'une classe	49
6.3 Composition de classes	50
6.4 Réutiliser du code	51
7. Les fonctions : les mots-clés def et lambda	53
7.1 Définir une fonction	53
7.2 La fonction anonyme	56
8. La syntaxe en compréhension	57
9. Itérateur et générateur : les mots-clés iter et yield	60
10. La gestion des exceptions avec les mots-clés try, except, raise et finally	64
11. L'import des modules avec le mot-clé import	68
12. La gestion de contexte avec les mots-clés with et as	70
13. Conclusion	72

Administration du Raspberry Pi en Python

1. Introduction	73
2. Naviguer dans le système de fichiers avec les modules os et pwd	74
2.1 Manipuler et interroger le système de fichiers	77
2.2 Explorer le système de fichiers du Raspberry Pi	78
3. Interagir avec l'interpréteur Python via le module sys	81
4. Lancer des commandes shell avec le module subprocess	83
5. Chercher des fichiers avec le module glob	85
6. Comparer des fichiers ou répertoires avec le module filecmp	87
7. Capturer des signaux UNIX avec le module signal	89
8. Écriture de scripts avec le module argparse	92
9. Conclusion	95

Le Raspberry Pi en console avec urwid

1. Introduction	97
2. urwid, les fondamentaux	97
3. Projet #1 : une horloge en console	100
4. Projet #2 : un navigateur de fichiers en console	102
5. Conclusion	106

Programmation d'interfaces graphiques avec tkinter

1. Les fondamentaux	107
2. Projet #1 : Hello world avec tkinter	110
3. Projet #2 : une visionneuse d'images	117
4. Projet #3 : un éditeur de texte	125
5. Conclusion	136

À l'assaut du Web avec le Raspberry Pi

1. Webscraping facile avec les modules urllib et HTMLParser
--

	137
2. Développer un serveur HTTP avec le module http.server	142
3. Exécuter des scripts avec le module cgi	148
4. Envoyer des e-mails avec le module smtplib	155
5. Écrire une API légère avec Flask	158
6. Conclusion	166

Multimédia et audio sur le Raspberry Pi

1. Dessiner avec Pillow	167
1.1 Créer et manipuler des images	167
1.2 Dessiner des figures géométriques	173
2. Contrôler les entrées et sorties audio avec pyalsaudio	175
3. Projet #1 : un enregistreur/lecteur audio	182
4. Conclusion	187

Persistance de données sur le Raspberry Pi

1. Introduction	189
2. Sérialisation et désérialisation avec les modules pickle et shelve	189
3. Traiter des fichiers CSV avec le module csv	193
3.1 Création et lecture d'un fichier CSV	193
3.2 Créer son propre dialecte CSV	195
4. Manipuler des données XML avec le module xml.etree.ElementTree	196
4.1 Créer et sérialiser un fichier XML	197
4.2 Interroger un fichier XML	198
4.3 Ajouter et supprimer des nœuds	199
5. Travailler avec le format d'échange de données JSON via le module json	201
6. Gestion d'une base de données SQL légère avec le module sqlite3	203
7. Conclusion	208

Documenter et tester ses scripts en Python

1. Introduction	209
------------------------	------------

2. Consulter de la documentation avec pydoc3	210
3. Documenter et tester son code en une seule fois avec le module doctest	217
4. Écriture de tests unitaires avec le module unittest	221
5. Benchmarker son code avec le module timeit	226
6. Déboguer ses programmes avec le module pdb	230
6.1 Déboguer pas à pas	231
6.2 Déboguer à un endroit précis du programme	234
6.3 Procéder à l'autopsie de son programme	234
7. Conclusion	235

Raspberry Pi et GPIO

1. Les GPIO, comment ça marche ?	237
2. Connecter un écran LCD 16x2 au Raspberry Pi	239
3. Projet #1 : communiquer avec l'écran LCD	245
4. Projet #2 : créer un tube FIFO dédié à l'écran LCD	248

5. Projet #3 : écrire des messages depuis une interface en ligne de commande	253
6. Projet #4 : piloter l'écran LCD depuis une interface graphique tkinter	256
7. Conclusion	259
Index	261