Flot de développement

1. Différentes versions de Raspberry Pi **9**

1.1 Raspberry Pi 3 B+ 11

1.2 Raspberry Pi Zero W 12

1.3 Système Raspbian 14

2. Historique du langage Python et distributions **17**

3. Extensions : packages et bibliothèques **19**

4. Outils d'édition **20**

5. Outils d'exécution d'un script **21**

5.1 Console 21

5.2 Outils intégrés 21

5.3 Outil IPython 22

6. Méthode de débogage et outil ipdb **23**

7. Écrire un code cohérent et homogène **25**

8. Configuration retenue pour l'ouvrage **28**

Éléments de base du langage

1. Variables et types de données simples **29**

1.1 Entiers 30

1.2 Flottants 31

1.3 Chaînes de caractères 32

1.3.1 Modification d’une chaîne de caractères 33

1.3.2 Propriété itérable des chaînes de caractères 34

1.3.3 Longueur d’une chaîne de caractères 34

1.3.4 Concaténation de chaînes de caractères 35

1.3.5 Conversion en chaîne de caractères 35

1.4 Booléens 36

2. Types de données complexes **37**

2.1 Listes 37

2.1.1 Définition 37

2.1.2 Création d’une liste 37

2.1.3 Accès aux données 38

2.1.4 Fonctions de manipulation d'une liste 39

2.2 Tuples 40

2.3 Dictionnaires

41

3. Opérateurs **41**

3.1 Opérateur d’affectation 42

3.2 Opérateurs arithmétiques 42

3.2.1 Addition 42

3.2.2 Soustraction 43

3.2.3 Multiplication 43

3.2.4 Division 44

3.2.5 Division entière 44

3.2.6 Opération composée 44

3.3 Opérateurs de comparaison 44

3.4 Opérateurs d’appartenance 45

3.5 Opérateurs logiques 46

4. Structures conditionnelles **48**

5. Boucles **50**

5.1 Boucle while 50

5.2 Boucle for 51

6. Exercice **54**

6.1 Énoncé

6.2 Solution 54
54

Modularité

1. Fonctions **57**

1.1 Cas classique 58

1.2 Pas de valeurs retournées 59

1.3 Plusieurs valeurs retournées 59

1.4 Paramètres par défaut 60

1.5 Rendre le code modulaire 62

2. Programmation orientée objet **63**

3. Python, programmation objet et différences principales avec d'autres langages **64**

4. Constructeur et attributs **65**

5. Méthodes membres **69**

6. Python et l'encapsulation **73**

7. Méthodes spéciales **76**

8. Héritage **80**

9. Créer ses propres modules et packages **84**

9.1 Modules 84

9.2 Packages 85

10. Travaux pratiques **86**

10.1 Le problème 86

10.2 Solution 87

11. Conclusion **90**

GPIO : un pas vers l'extérieur

1. Définition et intérêt du GPIO **91**

2. Broches d'entrée/sortie numériques **93**

3. Port I2C **96**

4. Interfacer un capteur usuel : le BME 280 **98**

5. Acquérir des données analogiques **106**

5.1 Étalonner le capteur 107

5.2 En pratique

108

6. Utiliser le port USB **110**

6.1 Connexion et configuration du port 111

6.2 Lecture sur le port 112

6.3 Écriture sur le port 114

6.4 Exemple complet 115

7. Utiliser le Bluetooth **116**

7.1 Lecture et écriture 120

7.2 Fonctionnement par notifications 121

8. Conclusion **123**

Manipulation des données

1. Lire et écrire dans des fichiers **125**

1.1 Ouvrir et fermer un fichier 125

1.2 Écriture 127

1.3 Lecture 130

1.4 Cas des fichiers binaires 132

1.4.1 Écriture 132

1.4.2 Lecture 134

2. Utiliser des CSV 135

3. Utiliser Pandas 139

4. Stocker des données dans une base MySQL 142

4.1 Connexion à la base de données 144

4.2 Écriture dans la base 144

4.3 Recherche dans la base 145

4.4 Conclusion 146

5. Visualiser des données avec matplotlib 147

6. Conclusion 154

tkinter et les interfaces graphiques

1. Présentation de tkinter 155

2. Widgets 158

2.1 Label 158

2.2 Bouton 160

2.3 Ligne de saisie

2.4 Cases à cocher 162
2.5 Boutons radio 163
2.6 Liste déroulante 165
3. Visualiseur d'images 166
4. Créer un datalogger **168**
5. Conclusion **171**
174

Multimédia

1. Acquérir un signal audio **175**
2. Compresser un signal audio **181**
3. Image et vidéo **183**
3.1 Interfacer la caméra officielle 183
3.2 Acquérir des images à partir du flux vidéo 189
3.3 Manipuler des images et leur contenu avec NumPy et scikit-image 190
3.3.1 Ouvrir et visualiser une image 192
3.3.2 Sauvegarder une image 193
3.3.3 Filtrage passe-bas

3.3.4 Détecter des contours 194
3.3.5 Manipuler des images avec NumPy 196
3.4 Détecter une présence ou un mouvement 198
3.4.1 Détecter un mouvement 200
3.4.2 Envoyer un e-mail 200
3.4.3 Sauvegarder des images après la détection d'une présence 201
3.4.4 Code complet 202
4. Conclusion 202

203

Programmation système

1. Module OS **205**

2. Flux standards **208**

3. Signaux **209**

4. Interpréter les arguments de la ligne de commande **211**

5. Exécuter une commande système **213**

6. Réseau **215**

6.1 Côté serveur 216
6.2 Côté client 217
6.3 Exemple complet 217

7. Programmation multithreading et calcul parallèle 220

7.1 Utiliser les threads 221
7.2 Calcul multiprocessing 222
7.3 Joblib 224

8. Conclusion 226

Pour aller plus loin

1. Documenter le code 227

2. Tester le code 230

2.1 Unittest 232
2.2 Pytest 234

3. Profiler le code 235

4. Conclusion 238

Index **239**