

Les éléments à télécharger sont disponibles à l'adresse suivante :
<http://www.editions-eni.fr>
Saisissez la référence ENI de l'ouvrage **RI33PYT** dans la zone de recherche
et validez. Cliquez sur le titre du livre puis sur le bouton de téléchargement.

Avant-propos

1. Introduction	23
2. Contenu de l'ouvrage	23
3. Progressivité de l'ouvrage	24
4. À destination des enseignants et élèves	25
5. À destination des chercheurs ou doctorants	27
6. À destination de ceux qui viennent d'un autre langage	27

Partie 1 : Les atouts de Python

Chapitre 1.1

Clés théoriques

1. Petite histoire des langages informatiques	29
1.1 Informatique théorique	29
1.2 Chronologie de l'informatique	30
1.2.1 Évolutions des problématiques liées à l'informatique	30
1.2.2 Chronologie des langages informatiques	31
2. Typologie des langages de programmation	35
2.1 Paradigmes	35
2.1.1 Définition	35
2.1.2 Paradigme impératif et dérivés	36
2.1.3 Paradigme objet et dérivés	37
2.1.4 Programmation orientée aspect	37
2.1.5 Paradigme fonctionnel	37
2.1.6 Paradigme logique	38
2.1.7 Programmation concurrente	38
2.1.8 Synthèse	38
2.2 Interopérabilité	39

2.3	Niveau de programmation	41
2.3.1	Machine	41
2.3.2	Bas niveau	41
2.3.3	Haut niveau	42
2.4	Typage	43
2.4.1	Faible vs fort	43
2.4.2	Statique vs dynamique	43
2.5	Grammaire	43
2.5.1	Langages formels	43
2.5.2	Syntaxe	44
3.	Python et le reste du monde	45
3.1	Positionnement stratégique du langage Python	45
3.1.1	Segments de marchés	45
3.1.2	Niveau de complexité	45
3.1.3	Forces du langage	45
3.1.4	Points faibles	46
3.2	Intégration avec d'autres langages	46
3.2.1	Extensions C	46
3.2.2	Intégration de programmes écrits en C	47
3.2.3	Intégration de programmes Python dans du C	47
3.2.4	Intégration de programmes écrits en Java	47
3.2.5	Intégration de programmes Python dans Java	47
3.2.6	Autres intégrations	47

Chapitre 1.2

Présentation de Python

1.	Philosophie	49
1.1	Python en quelques lignes	49
1.1.1	D'où vient le nom « Python » ?	49
1.1.2	Présentation technique	49
1.1.3	Présentation conceptuelle	50
1.2	Comparaison avec d'autres langages	50
1.2.1	Shell	50
1.2.2	Perl	51
1.2.3	C, C++	51
1.2.4	Java	52
1.2.5	PHP	54

1.3	Grands principes	55
1.3.1	Le zen de Python	55
1.3.2	Le développeur n'est pas stupide	55
1.3.3	Documentation	56
1.3.4	Python est livré piles incluses	56
1.3.5	Duck Typing	56
1.3.6	Notion de code pythonique	57
2.	Histoire de Python.	57
2.1	La genèse	57
2.2	Extension du périmètre fonctionnel	58
2.3	Évolution de la licence	62
2.4	Avenir	62
3.	Gouvernance	63
3.1	Développement	63
3.1.1	Branches	63
3.1.2	Communauté.	64
3.2	Mode de gouvernance.	65
3.2.1	Créateur du langage.	65
3.2.2	PEP	65
3.2.3	Prise de décisions	65
3.2.4	Contribuer à Python	66
4.	Que contient Python ?	66
4.1	Une grammaire et une syntaxe	66
4.2	Plusieurs implémentations.	66
4.3	Une bibliothèque standard.	67
4.4	Des bibliothèques tierces	67
4.5	Des frameworks	67
5.	Phases d'exécution d'un programme Python.	67
5.1	Chargement de la machine virtuelle	67
5.2	Compilation	68
5.3	Interprétation	69

Chapitre 1.3

Pourquoi choisir Python

1. Qualités du langage	71
1.1 Ticket d'entrée	71
1.2 Qualités intrinsèques	73
1.3 Couverture fonctionnelle	73
1.4 Domaines d'excellence	74
1.5 Garanties	75
2. Diffusion	76
2.1 Entreprises	76
2.2 Le monde de la recherche	77
2.3 Le monde de l'éducation	78
2.4 Communauté	79
3. Références	80
3.1 Poids lourds de l'industrie informatique	80
3.1.1 Google	80
3.1.2 Mozilla	81
3.1.3 Microsoft	81
3.1.4 Canonical	81
3.1.5 Cisco	82
3.2 Entreprises innovantes	82
3.2.1 Services de stockage en ligne	82
3.2.2 Informatique dématérialisée	82
3.2.3 Forge	82
3.2.4 Réseaux sociaux	83
3.3 Éditeurs de contenus	83
3.3.1 Disney Animation Studio	83
3.3.2 YouTube	83
3.3.3 Box ADSL	83
3.3.4 Spotify	83
3.4 Éditeurs de logiciels	83
4. Retours d'expérience	84
4.1 Internet des objets	84
4.2 Système et développement web	85
4.3 Enseignement	85
4.4 Embarqué	86

4.5 Développement web..... 87
4.6 ERP..... 87

Chapitre 1.4
Installer son environnement de travail

1. Introduction..... 89
2. Installer Python..... 89
 2.1 Pour Windows..... 89
 2.2 Pour Mac..... 92
 2.3 Pour GNU/Linux et BSD..... 92
 2.4 Par la compilation..... 93
 2.5 Pour un smartphone..... 94
3. Installer une bibliothèque tierce..... 94
 3.1 À partir de Python 3.4..... 94
 3.2 Pour une version inférieure à Python 3.4..... 96
 3.3 Pour Linux..... 96
4. Créer un environnement virtuel..... 97
 4.1 À quoi sert un environnement virtuel ?..... 97
 4.2 Pour Python 3.3 ou version supérieure..... 97
 4.3 Pour toute version de Python..... 98
 4.4 Pour Linux..... 99
5. Installer Anaconda..... 100
 5.1 Pour Windows..... 100
 5.2 Pour Linux..... 103
 5.3 Pour Mac..... 103
 5.4 Mettre à jour Anaconda..... 104
 5.5 Installer une bibliothèque externe..... 104
 5.6 Environnements virtuels..... 104
6. Docker..... 104
7. La console Python..... 105
 7.1 Démarrer la console Python..... 105
 7.2 BPython..... 105
 7.3 IPython..... 106
 7.4 IPython Notebook..... 106

8.	Installer un IDE	107
8.1	Liste d'IDE	107
8.2	Présentation de PyCharm	107
8.3	Configuration de PyCharm	108

Partie 2 : Guide Python

Chapitre 2.1

Les premiers pas

1.	Avant de commencer	113
1.1	Quelques notions importantes	113
1.1.1	Comment fonctionne un ordinateur ?	113
1.1.2	Qu'est-ce qu'un programme informatique ?	114
1.1.3	Qu'est-ce qu'un code source ?	114
1.2	Quelques conventions utilisées dans ce livre	114
1.2.1	Code Python	114
1.2.2	Terminal	115
1.2.3	Mise en forme	115
1.3	Quelle est la meilleure méthode pour apprendre ?	116
2.	Premier programme	116
2.1	Hello world !	116
2.2	Affectation	118
2.3	Valeur booléenne	119
2.4	Type	120
2.5	Exceptions	121
2.6	Bloc conditionnel	124
2.7	Conditions avancées	125
2.8	Bloc itératif	126
3.	Premier jeu : Devine le nombre	128
3.1	Description du jeu	128
3.2	Aides	128
3.2.1	Gestion du hasard	128
3.2.2	Étapes de développement	128
3.3	Pour aller plus loin	129

Chapitre 2.2
Fonctions et modules

- 1. Les fonctions 131
 - 1.1 Pourquoi utiliser des fonctions ? 131
 - 1.2 Introduction aux fonctions 133
 - 1.2.1 Comment déclarer une fonction. 133
 - 1.2.2 Gestion d'un paramètre. 134
 - 1.2.3 Comment rendre une fonction plus générique 136
 - 1.2.4 Paramètres par défaut 138
 - 1.3 Problématiques de couplage et duplication de code 139
 - 1.3.1 Niveau de ses fonctions. 139
 - 1.3.2 Notion de complexité 141
 - 1.3.3 Bonnes pratiques 143
- 2. Les modules 144
 - 2.1 Introduction 144
 - 2.1.1 Qu'est-ce qu'un module ? 144
 - 2.1.2 Comment crée-t-on un module Python ? 145
 - 2.1.3 Organiser son code 145
 - 2.2 Gérer le code de ses modules 145
 - 2.2.1 Exécuter un module, importer un module. 145
 - 2.2.2 Gérer une arborescence de modules 146
- 3. Terminer le jeu. 147
 - 3.1 Créer des niveaux 148
 - 3.2 Déterminer un nombre de coups maximal 148
 - 3.3 Enregistrer les meilleurs scores. 148
 - 3.4 Intelligence artificielle 148

Chapitre 2.3
Les principaux types

- 1. Chaînes de caractères 149
 - 1.1 Syntaxe 149
 - 1.2 Formatage d'une chaîne 150
 - 1.3 Notion de casse. 151
 - 1.4 Notion de longueur. 152
 - 1.5 Appartenance 152
 - 1.6 Notion d'occurrence. 153

1.7	Remplacement	154
1.8	Notion de caractère	154
1.9	Typologie des caractères	155
1.10	Séquencer une chaîne de caractères	156
2.	Listes	156
2.1	Syntaxe	156
2.2	Indices	157
2.3	Valeurs	158
2.4	Hasard	159
2.5	Techniques d'itération	160
2.6	Tri	162
3.	Dictionnaires	164
3.1	Présentation des dictionnaires	164
3.2	Parcourir un dictionnaire	165
3.3	Exemple	165

Chapitre 2.4

Les classes

1.	Syntaxe	167
2.	Notion d'instance courante	168
3.	Opérateurs	170
4.	Héritage	172
4.1	Spécialisation	172
4.2	Programmation par composants	173

Partie 3 : Les fondamentaux du langage

Chapitre 3.1

Algorithmique de base

1.	Délimiteurs	175
1.1	Instruction	175
1.2	Une ligne de code = une instruction	175
1.3	Commentaire	176
1.4	Une instruction sur plusieurs lignes	176
1.5	Mots-clés	176

1.6	Mots réservés	177
1.7	Indentation	178
1.8	Symboles	179
1.9	Opérateurs	182
1.9.1	Opérateur expression d'affectation :=	185
1.10	Utilisation du caractère souligné	186
1.11	PEP-8	187
1.12	PEP-7	187
1.13	PEP-257	187
2.	Instructions	187
2.1	Définitions	187
2.1.1	Variable	187
2.1.2	Fonction	189
2.1.3	Fonctions lambda	190
2.1.4	Classe	191
2.1.5	Instruction vide	192
2.1.6	Suppression	192
2.1.7	Renvoyer le résultat de la fonction	193
2.2	Instructions conditionnelles	194
2.2.1	Définition	194
2.2.2	Condition	194
2.2.3	Instruction if	194
2.2.4	Instruction elif	195
2.2.5	Instruction else	195
2.3	Utilisation d'une expression d'affectation	197
2.3.1	Instruction switch	197
2.3.2	Interruptions	197
2.3.3	Approfondissement des conditions	198
2.3.4	Performances	199
2.4	Itérations	200
2.4.1	Instruction for	200
2.4.2	Instruction while	200
2.4.3	Quelle différence entre for et while ?	201
2.4.4	Instruction break	201
2.4.5	Instruction return	203
2.4.6	Instruction continue	203
2.4.7	Instruction else	203

2.4.8	Générateurs	204
2.5	Constructions fonctionnelles	207
2.5.1	Construction conditionnelle	207
2.5.2	Générateurs	207
2.5.3	Compréhensions de listes	207
2.5.4	Compréhensions d'ensembles	208
2.5.5	Compréhensions de dictionnaires	208
2.6	Compréhensions et expressions d'affectation	208
2.7	Gestion des exceptions	208
2.7.1	Présentation rapide des exceptions	208
2.7.2	Lever une exception	209
2.7.3	Pourquoi lever une exception ?	209
2.7.4	Assertions	210
2.7.5	Capturer une exception	211
2.7.6	Effectuer un traitement de l'exception	212
2.7.7	Gérer la sortie du bloc de capture	214
2.7.8	Gérer le non-déclenchement d'exceptions	214
2.7.9	Prise et libération de ressources	216
2.7.10	Programmation asynchrone	217
2.8	Divers	218
2.8.1	Gérer des imports	218
2.8.2	Traverser les espaces de nommage	219
2.8.3	Fonctions print, help, eval et exec	221

Chapitre 3.2

Déclarations

1.	Variable	223
1.1	Qu'est-ce qu'une variable ?	223
1.1.1	Contenu	223
1.1.2	Contenant	223
1.1.3	Modes de modification d'une variable	225
1.2	Typage dynamique	228
1.2.1	Affectation : rappels	228
1.2.2	Primitive type et nature du type	228
1.2.3	Caractéristiques du typage Python	229

1.3	Visibilité	231
1.3.1	Espace global	231
1.3.2	Notion de bloc	232
2.	Fonction	235
2.1	Déclaration	235
2.2	Paramètres	236
2.2.1	Signature d'une fonction	236
2.2.2	Notion d'argument ou de paramètre	237
2.2.3	Valeur par défaut	237
2.2.4	Valeur par défaut mutable	239
2.2.5	Paramètres nommés	240
2.2.6	Déclaration de paramètres extensibles	240
2.2.7	Passage de paramètres étoilés	242
2.2.8	Signature universelle	242
2.2.9	Obliger un paramètre à être nommé (keyword-only)	243
2.3	Obliger un paramètre à être positionnel (Positional-only)	245
2.3.1	Annotations	245
2.3.2	Types hint	249
3.	Classe	251
3.1	Déclaration	251
3.1.1	Signature	251
3.1.2	Attribut	251
3.1.3	Méthode	252
3.1.4	Bloc local	252
3.2	Instanciation	253
3.2.1	Syntaxe	253
3.2.2	Relation entre l'instance et la classe	253
4.	Module	254
4.1	À quoi sert un module ?	254
4.2	Déclaration	254
4.3	Instructions spécifiques	254
4.4	Comment appréhender le contenu d'un module ?	255
4.5	Compilation des modules	256

Chapitre 3.3

Modèle objet

1.	Tout est objet	259
1.1	Principes	259
1.1.1	Quel sens donner à « objet » ?	259
1.1.2	Adaptation de la théorie objet dans Python	260
1.1.3	Généralités	261
1.2	Classes	261
1.2.1	Introduction	261
1.2.2	Déclaration impérative d'une classe	262
1.2.3	Instance	262
1.2.4	Objet courant	264
1.2.5	Déclaration par prototype d'une classe	264
1.2.6	Tuples nommés	267
1.3	Méthodes	267
1.3.1	Déclaration	267
1.3.2	Appel de méthode	269
1.3.3	Méthodes et attributs spéciaux	271
1.3.4	Constructeur et initialisateur	275
1.3.5	Gestion automatisée des attributs	276
1.3.6	Intérêt du paradigme objet	276
1.3.7	Relation entre objets	277
1.4	Héritage	277
1.4.1	Polymorphisme par sous-typage	277
1.4.2	Surcharge de méthode	278
1.4.3	Surcharge des opérateurs	280
1.4.4	Polymorphisme paramétrique	281
1.4.5	Héritage multiple	283
2.	Autres outils de la programmation objet	285
2.1	Principes	285
2.2	Interfaces	285
2.3	Attributs	288
2.4	Propriétés	290
2.5	Emplacements	292
2.6	Métaclasses	294
2.7	Classes abstraites	296

- 2.8 La Zope Component Architecture. 299
 - 2.8.1 Présentation 299
 - 2.8.2 Installation 299
 - 2.8.3 Définir une interface et un composant 300
 - 2.8.4 Autres fonctionnalités. 301
 - 2.8.5 Avantages de la ZCA 301
- 3. Fonctions spéciales et primitives associées 301
 - 3.1 Personnalisation 301
 - 3.1.1 Classes 301
 - 3.1.2 Instances. 303
 - 3.1.3 Comparaison 304
 - 3.1.4 Évaluation booléenne 304
 - 3.1.5 Relations d’héritage ou de classe à instance. 305
 - 3.2 Classes particulières 305
 - 3.2.1 Itérateurs 305
 - 3.2.2 Conteneurs. 308
 - 3.2.3 Instances assimilables à des fonctions 308
 - 3.2.4 Ressources à protéger 309
 - 3.2.5 Types 310
 - 3.2.6 Classes de données. 310

Chapitre 3.4

Types de données et algorithmes appliqués

- 1. Nombres. 311
 - 1.1 Types 311
 - 1.1.1 Entiers. 311
 - 1.1.2 Réels 312
 - 1.1.3 Socle commun aux nombres entiers et réels 313
 - 1.1.4 Méthodes dédiées aux nombres entiers 314
 - 1.1.5 Méthodes dédiées aux nombres réels 315
 - 1.1.6 Complexes 315
 - 1.2 La console Python, la calculatrice par excellence 316
 - 1.2.1 Opérateurs mathématiques binaires 316
 - 1.2.2 Opérateurs binaires particuliers 317
 - 1.2.3 Opérateurs mathématiques unaires 318
 - 1.2.4 Arrondis 319
 - 1.2.5 Opérateurs de comparaison 321

1.2.6	Opérations mathématiques n-aires	322
1.2.7	Fonctions mathématiques usuelles	323
1.3	Représentations d'un nombre	329
1.3.1	Représentation décimale	329
1.3.2	Représentation par un exposant	329
1.3.3	Représentation par une fraction	329
1.3.4	Représentation hexadécimale	330
1.3.5	Représentation octale	331
1.3.6	Représentation binaire	332
1.3.7	Opérations binaires	332
1.3.8	Longueur de la représentation mémoire d'un entier	334
1.4	Conversions	335
1.4.1	Conversion entre entiers et réels	335
1.4.2	Conversion entre réels et complexes	336
1.4.3	Conversion vers un booléen	336
1.5	Travailler avec des variables	337
1.5.1	Un nombre est non mutable	337
1.5.2	Modifier la valeur d'une variable	338
1.5.3	Opérateurs d'incrément	338
1.6	Statistiques	339
2.	Séquences	340
2.1	Présentation des différents types de séquences	340
2.1.1	Généralités	340
2.1.2	Les listes	341
2.1.3	Les n-uplets	342
2.1.4	Conversion entre listes et n-uplets	344
2.1.5	Socle commun entre liste et n-uplet	344
2.1.6	Notion d'itérateur	345
2.2	Utilisation des indices et des tranches	347
2.2.1	Définition de l'indice d'un objet et des occurrences	347
2.2.2	Utiliser l'indice pour adresser la séquence	349
2.2.3	Retrouver les occurrences d'un objet et leurs indices	350
2.2.4	Taille d'une liste, comptage d'occurrences	351
2.2.5	Utiliser l'indice pour modifier ou supprimer	352
2.2.6	Itération simple	354
2.2.7	Présentation de la notion de tranches (slices)	357
2.2.8	Cas particulier de la branche 2.x de Python	366

2.2.9	Utilisation basique des tranches	367
2.2.10	Utilisation avancée des tranches	368
2.3	Utilisation des opérateurs	370
2.3.1	Opérateur +	370
2.3.2	Opérateur *	371
2.3.3	Opérateur +=	373
2.3.4	Opérateur *=	375
2.3.5	Opérateur in	376
2.3.6	Opérateurs de comparaison	377
2.4	Méthodes de modifications	378
2.4.1	Ajouter des éléments dans une liste et un n-uplet	378
2.4.2	Supprimer un objet d'une liste et d'un n-uplet	380
2.4.3	Solutions de contournement pour la modification de n-uplets	384
2.4.4	Renverser une liste ou un tuple	385
2.4.5	Trier une liste	386
2.5	Utilisation avancée des listes	388
2.5.1	Opérations d'ensemble	388
2.5.2	Pivoter une séquence	389
2.5.3	Itérer correctement	390
2.5.4	Programmation fonctionnelle	391
2.5.5	Compréhensions de listes	394
2.5.6	Itérations avancées	395
2.5.7	Combinatoire	400
2.6	Adapter les listes à des besoins spécifiques	402
2.6.1	Liste d'entiers	402
2.6.2	Présentation du type array	403
2.6.3	Utiliser une liste comme pile	405
2.6.4	Utiliser une liste comme file d'attente	406
2.6.5	Conteneur plus performant	406
2.6.6	Utiliser des listes pour représenter des matrices	407
2.6.7	Liste sans doublons	408
2.7	Autres types de données	411
3.	Ensembles	413
3.1	Présentation	413
3.1.1	Définition d'un ensemble	413
3.1.2	Différences entre set et frozenset	414

3.1.3	Utilisation pour dédoublonner des listes	415
3.1.4	Rajouter une relation d'ordre	415
3.2	Opérations ensemblistes	416
3.2.1	Opérateurs pour un ensemble à partir de deux autres	416
3.2.2	Opérateurs pour modifier un ensemble à partir d'un autre	417
3.2.3	Méthodes équivalentes à la création ou modification ensembliste	418
3.2.4	Méthodes de comparaison des ensembles	418
3.2.5	Exemples non classiques d'utilisation	419
3.3	Méthodes de modification d'un ensemble	423
3.3.1	Ajouter un élément	423
3.3.2	Supprimer un élément	423
3.3.3	Vider un ensemble	424
3.3.4	Dupliquer un élément	424
3.3.5	Sortir une valeur d'un ensemble	425
3.3.6	Utiliser un ensemble comme un recycleur d'objets	426
3.3.7	Algorithmique avancée : résolution du problème des n-dames	428
4.	Chaînes de caractères	430
4.1	Présentation	430
4.1.1	Définition	430
4.1.2	Vocabulaire	431
4.1.3	Spécificités de la branche 2.x	432
4.1.4	Changements apportés par la branche 3.x	433
4.1.5	Chaîne de caractères en tant que séquence de caractères	435
4.1.6	Caractères	437
4.1.7	Opérateurs de comparaison	438
4.2	Formatage de chaînes de caractères	441
4.2.1	Opérateur modulo	441
4.2.2	Méthodes de formatage sur l'ensemble de la chaîne	446
4.2.3	Nouvelle méthode de formatage des variables dans une chaîne	448
4.2.4	Littéraux formatés	451
4.3	Opérations d'ensemble	452
4.3.1	Séquençage de chaînes	452
4.3.2	Opérations sur la casse	454
4.3.3	Recherche sur une chaîne de caractères	455

4.3.4	Informations sur les caractères	456
4.4	Problématiques relatives à l'encodage	457
4.4.1	Encodage par défaut	457
4.4.2	Encodage du système.	458
4.4.3	L'unicode, référence absolue	458
4.4.4	Autres encodages	459
4.4.5	Ponts entre l'unicode et le reste du monde	460
4.4.6	Revenir vers l'Unicode.	461
4.5	Manipulations de bas niveau avancées	461
4.5.1	Opérations de comptage	461
4.5.2	Une chaîne de caractères vue comme une liste	462
4.5.3	Une chaîne de caractères vue comme un ensemble de caractères	463
4.6	Représentation mémoire	463
4.6.1	Présentation du type bytes	463
4.6.2	Lien avec les chaînes de caractères	464
4.6.3	Présentation du type bytearray	466
4.6.4	Gestion d'un jeu de caractères	468
5.	Dictionnaires	473
5.1	Présentation	473
5.1.1	Définition.	473
5.1.2	Évolutions et différences entre les branches 2.x et 3.x	474
5.1.3	Vues de dictionnaires.	475
5.1.4	Instanciation	477
5.1.5	Compréhension de dictionnaire	477
5.2	Manipuler un dictionnaire	478
5.2.1	Récupérer une valeur d'un dictionnaire	478
5.2.2	Modifier les valeurs d'un dictionnaire	479
5.2.3	Supprimer une entrée d'un dictionnaire	480
5.2.4	Dupliquer un dictionnaire.	480
5.2.5	Utiliser le dictionnaire comme agrégateur de données	481
5.2.6	Méthodes d'itération.	482
5.3	Utilisation avancée des dictionnaires	482
5.3.1	Rajouter une relation d'ordre	482
5.3.2	Algorithmiques classiques.	486
5.3.3	Adapter les dictionnaires à des besoins spécifiques	488
5.3.4	Représentation universelle de données	490

6.	Booléens	491
6.1	Le type booléen	491
6.1.1	Classe bool	491
6.1.2	Les deux objets True et False	492
6.1.3	Différence entre l'opérateur d'égalité et d'identité	492
6.2	Évaluation booléenne	492
6.2.1	Méthode générique	492
6.2.2	Objets classiques	492
7.	Données temporelles	493
7.1	Gérer une date calendaire	493
7.1.1	Notion de date calendaire	493
7.1.2	Travailler sur une date	494
7.1.3	Considérations astronomiques	495
7.1.4	Considérations historiques	495
7.1.5	Considérations techniques	495
7.1.6	Représentation textuelle	496
7.2	Gérer un horaire ou un moment d'une journée	498
7.2.1	Notion d'instant	498
7.2.2	Notion de fuseau horaire	499
7.2.3	Représentation textuelle	499
7.3	Gérer un instant absolu	500
7.3.1	Notion d'instant absolu	500
7.3.2	Rapport avec les notions précédentes	501
7.3.3	Représentation textuelle	502
7.3.4	Gestion des fuseaux horaires	503
7.3.5	Créer une date à partir d'une représentation textuelle	503
7.4	Gérer une différence entre deux dates ou instants	503
7.4.1	Notion de différence et de résolution	503
7.4.2	Considérations techniques	505
7.4.3	Utilisation avec des dates calendaires	506
7.4.4	Utilisation avec des horaires	506
7.4.5	Utilisation avec des dates absolues	506
7.4.6	La seconde comme unité de base	506
7.4.7	Précision à la nanoseconde	507
7.5	Spécificités des fuseaux horaires	507

- 7.6 Problématiques de bas niveau 508
 - 7.6.1 Timestamp et struct_time 508
 - 7.6.2 Mesures de performances 509
- 7.7 Utilisation du calendrier. 511
 - 7.7.1 Présentation du module calendar 511
 - 7.7.2 Fonctions essentielles du calendrier 516

Partie 4 : Les fonctionnalités

Chapitre 4.1 Manipulation de données

- 1. Manipuler des fichiers 519
 - 1.1 Ouvrir un fichier 519
 - 1.2 Lire un fichier 520
 - 1.3 Écrire un fichier. 521
 - 1.4 Comparer deux fichiers 522
- 2. Utilitaire de sauvegarde. 524
- 3. Lire un fichier de configuration 524
- 4. Format d'export/Import 525
 - 4.1 CSV 525
 - 4.1.1 Exploiter un fichier CSV 526
 - 4.1.2 Génération d'un fichier CSV 529
 - 4.2 JSON 531
 - 4.3 Base64 534
 - 4.4 Pickle 534
- 5. Compresser et décompresser un fichier 537
 - 5.1 Tarfile 537
 - 5.2 Gzip 539
 - 5.3 Bz2 539
 - 5.4 Zipfile 540
 - 5.5 Interface de haut niveau. 542
- 6. Outils de manipulation de données 543
 - 6.1 Générer des nombres aléatoires 543
 - 6.2 Expressions régulières. 544

7. Cryptographie légère	548
7.1 Nombre aléatoire sécurisé	548
7.2 Fonctions de chiffrement	548
7.3 Code d'authentification de message	550
7.4 Empreinte de fichier	551
7.5 Stéganographie	552
7.6 Communication inter-applicative sécurisée	555

Chapitre 4.2

Bases de données

1. Introduction	559
2. Accès à une base de données relationnelle	559
2.1 Point d'entrée	559
2.2 MySQL	560
2.3 PostgreSQL	565
2.4 SQLite	567
2.5 Oracle	567
3. Utilisation d'un ORM	568
3.1 Qu'est-ce qu'un ORM ?	568
3.2 ORM proposés par Python	568
3.3 SQLAlchemy	569
4. Autres bases de données	575
4.1 CSV	575
4.2 NoSQL	581
4.3 Base de données orientée objet : ZODB	581
4.4 Base de données orientée graphe : Neo4j	586
4.5 Base de données de type clé-valeur : Redis	587
4.6 Bases de données orientées documents : CouchDB et MongoDB	589
4.7 Bases de données natives XML : BaseX, eXist	590
4.8 Cassandra	591
4.9 Bases de données orientées colonnes : HBase	591
4.10 Big Data : l'écosystème Hadoop	593
5. LDAP	595
5.1 Protocole	595
5.2 Serveurs	595
5.3 Terminologie	596

- 5.4 Installation 596
- 5.5 Ouvrir une connexion à un serveur 596
- 5.6 Effectuer une recherche 598
- 5.7 Synchrone vs asynchrone 599
- 5.8 Connexions sécurisées 599

Partie 5 : Mise en pratique

Chapitre 5.1

Créer une application web en 30 minutes

- 1. Description de l'application à construire 601
- 2. Mise en place 602
 - 2.1 Isolation de l'environnement 602
 - 2.2 Création du projet 603
 - 2.3 Paramétrage 603
 - 2.4 Premiers essais 604
- 3. Réalisation de l'application 605
 - 3.1 Modèles 605
 - 3.2 Vues 608
 - 3.3 Contrôleurs 609
- 4. Pour aller plus loin 613

Chapitre 5.2

Créer une application console en 10 minutes

- 1. Objectif 615
- 2. Enregistrer le script 616
- 3. Création des données 616
- 4. Parseur d'arguments 617

Chapitre 5.3

Créer une application graphique en 20 minutes

- 1. Objectif 619
 - 1.1 Fonctionnel 619
 - 1.2 Technique 619

2. Présentation rapide de Gtk et d'astuces	620
2.1 Présentation	620
2.2 Astuces	620
3. Démarrer le programme	621
4. Interface graphique avec Glade	623
5. Créer le composant graphique	626
6. Contrôleur	628
7. Autres bibliothèques graphiques	629
7.1 TkInter	629
7.2 wxPython	629
7.3 PyQt	629
7.4 PySide	630
7.5 Autres	630

Chapitre 5.4

Créer un jeu en 30 minutes avec PyGame

1. Présentation de PyGame	631
2. Réalisation d'un jeu Tetris	632
2.1 Présentation du jeu	632
2.2 Présentation des problématiques	633
2.3 Création des constantes	633

Annexes

1. Objets mutables et non mutables	645
2. Table Unicode	647
2.1 Script	647
3. Bytes	647
3.1 Script	647
3.2 Résultat	647
4. Guide de portage vers Python 3	650

Index	653
-----------------	-----

Les éléments à télécharger sont disponibles à l'adresse suivante :

<http://www.editions-eni.fr>

Saisissez la référence ENI de l'ouvrage **LF4RASP** dans la zone de recherche et validez. Cliquez sur le titre du livre puis sur le bouton de téléchargement.

Avant-propos

Chapitre 1

Raspberry Pi 4 model B

1. Introduction	17
2. Les composants du Raspberry Pi 4	18
2.1 Vue de dessus	18
2.2 Vue de dessous	19
3. Le SoC du Raspberry Pi 4	19
3.1 Le CPU	19
3.2 Le GPU	20
3.3 La mémoire	21
4. Les ports USB et Ethernet du Raspberry Pi 4	22
4.1 USB 3 et USB 2	22
4.2 Ethernet Gigabit	22
4.3 Ports USB et Ethernet	23
5. Le Wi-Fi et le Bluetooth du Raspberry Pi 4	24
6. Les sorties vidéo du Raspberry Pi 4	25
6.1 Vidéo numérique	25
6.2 Vidéo analogique	26
7. Les sorties audio	26
7.1 Audio numérique	26
7.2 Audio analogique	27
8. Le bus CSI du Raspberry Pi 4	27
9. Le bus DSI du Raspberry Pi 4	28
10. Le GPIO du Raspberry Pi 4	28

2 Raspberry Pi 4 - Exploitez tout le potentiel de votre nano-ordinateur

11. L'alimentation du Raspberry Pi 4	29
11.1 Par la prise d'alimentation	29
11.2 Par la PoE	31
11.3 Par le GPIO	31
11.4 Par les ports USB	32
11.4.1 Alimentation des périphériques	32
11.4.2 Alimentation du Raspberry Pi	32
11.5 Les bornes RUN et GLOBAL_EN	32
12. Le connecteur de carte micro SD du Raspberry Pi 4	33
13. Problème de boot	34
14. Les LED d'état du Raspberry Pi 4	35
15. Les dimensions physiques du Raspberry Pi 4	36
16. Conclusion	36

Chapitre 2

Raspberry Pi 3 B+

1. Introduction	37
2. Les composants du Raspberry Pi 3 B+	38
2.1 Vue de dessus	38
2.2 Vue de dessous	39
3. Le SoC du Raspberry Pi 3 B+	39
3.1 Le CPU	39
3.2 Le GPU	40
3.3 La mémoire	42
4. Les ports USB et Ethernet du Raspberry Pi 3 B+	43
5. Le Wi-Fi et le Bluetooth du Raspberry Pi 3 B+	45
6. Les sorties vidéo du Raspberry Pi 3 B+	46
6.1 Vidéo numérique	46
6.2 Vidéo analogique	48
6.3 Les sorties audio	51
6.3.1 Audio numérique	51
6.3.2 Audio analogique	51
7. Le bus CSI du Raspberry Pi 3 B+	52

8. Le bus DSI du Raspberry Pi 3 B+	55
9. Le GPIO du Raspberry Pi 3 B+	56
10. L'alimentation du Raspberry Pi 3 B+	57
10.1 Par la prise d'alimentation	57
10.1.1 Différence entre chargeur et alimentation	58
10.1.2 Importance du câble USB	60
10.1.3 Alimentation officielle	61
10.1.4 Indicateur de sous-tension	61
10.2 Par la PoE	62
10.3 Par le GPIO	64
10.4 Par les ports USB	66
10.4.1 Alimentation des périphériques	66
10.4.2 Alimentation du Raspberry Pi	66
10.5 Les bornes RUN et PEN	67
11. Le connecteur de carte micro SD du Raspberry Pi 3 B+	68
12. Les LED d'état du Raspberry Pi 3 B+	69
13. Les dimensions physiques du Raspberry Pi 3 B+	70
14. L'horloge temps réel	71
15. Conclusion	72

Chapitre 3

Raspberry Pi 3 A+

1. Présentation	73
2. Les composants du Raspberry Pi 3 A+	74
2.1 Les composants identiques au B+	74
2.2 Vue de dessus	75
2.3 Vue de dessous	76
2.4 Vue latérale	76
2.5 Le port USB du Raspberry Pi 3 A+	77
2.6 Wi-Fi - Bluetooth	77
2.7 LED ACT et PWR	78
3. Les dimensions physiques	79
4. Tableau comparatif modèles 3 et 4	80

4 Raspberry Pi 4 - Exploitez tout le potentiel de votre nano-ordinateur

Chapitre 4

Préparer la carte micro SD

1. Introduction	81
2. Préparation de la carte micro SD	82
2.1 Accès à la carte micro SD	82
2.2 Acquisition d'une carte SD prête à l'emploi	83
3. Installation du système d'exploitation	83
3.1 Le site de téléchargement de la Fondation	84
3.2 Installation d'un OS avec NOOBS	85
3.2.1 Téléchargement de NOOBS	85
3.2.2 Vérification du téléchargement	87
3.2.3 Formatage de la carte micro SD	89
3.2.4 Copie de NOOBS sur la carte micro SD	93
3.2.5 Démarrage de NOOBS	95
3.2.6 Conclusion	100
3.3 Installation d'une image de Raspbian	101
3.3.1 Téléchargement de Raspbian	101
3.3.2 Transfert de Raspbian sur la carte micro SD	102
3.3.3 Sous Windows 10	103
3.3.4 Sous Linux Debian	105
4. Conclusion	108

Chapitre 5

Démarrer Raspbian

1. Préparation du Raspberry Pi	109
2. Mise en place de la carte SD	109
3. Connexion des câbles	110
3.1 Raspberry Pi 4	111
3.2 Raspberry Pi 3 B+	111
3.3 Raspberry Pi 3 A+	112

4. Mise sous tension	112
4.1 Démarrage de Raspbian Desktop	112
4.1.1 Configuration initiale	112
4.1.2 Redémarrage de Raspbian Desktop	116
4.2 Démarrage de Raspbian Lite	117
4.2.1 Création de la carte micro SD	117
4.2.2 Démarrage du système	117
4.3 Configuration avec raspi-config	118
4.3.1 Démarrage de raspi-config	119
4.3.2 Utilisation de raspi-config	119
4.3.3 Raspbian en mode headless	121
5. Conclusion	124

Chapitre 6

Utiliser le mode graphique

1. Introduction	125
2. Découverte de l'écran de Raspbian Buster	126
3. Le tableau de bord	126
3.1 Icônes à gauche	126
3.1.1 Menu principal	127
3.1.2 Navigateur web	128
3.1.3 Gestionnaire de fichiers	128
3.1.4 Terminal	128
3.1.5 Corbeille	129
3.2 Icônes à droite	129
3.2.1 Bluetooth	129
3.2.2 Réseau Ethernet et Wi-Fi	130
3.2.3 Son	130
3.2.4 Charge CPU	131
3.2.5 Horloge	131
3.2.6 Ejection	132
3.3 Ajouter/retirer des icônes	132
3.3.1 Sur le bureau	132
3.3.2 Dans le tableau de bord	133

6 Raspberry Pi 4 - Exploitez tout le potentiel de votre nano-ordinateur

4. Configurer le Raspberry Pi 4	136
4.1 Onglet Système	137
4.2 Onglet Interfaces	139
4.3 Onglet Performance	139
4.4 Onglet Localisation	140
4.5 Redémarrage du système	140
4.6 Gestion du double écran	141
4.7 Aspect du bureau	143
4.8 Configuration de la vidéo	144
4.9 Arrêt du système	145
5. Conclusion	146

Chapitre 7

Utiliser la ligne de commande

1. Introduction	147
2. L'arborescence de Linux	149
3. La ligne de commande	151
3.1 Connexion à Raspbian	151
3.1.1 Connexion en utilisateur normal	152
3.1.2 Connexion en root	155
3.2 Se déplacer dans l'arborescence	158
3.2.1 Identifier le répertoire courant	158
3.2.2 Lister le contenu d'un répertoire	158
3.2.3 Changer de répertoire	160
3.2.4 Chemin relatif et chemin absolu	161
3.2.5 Modifier l'arborescence	163
3.3 Gérer les fichiers	165
3.3.1 Copier les fichiers	165
3.3.2 Déplacer et renommer les fichiers	167
3.3.3 Supprimer les fichiers	167
3.3.4 Afficher le contenu d'un fichier	168
3.3.5 Modifier le contenu d'un fichier	169
3.3.6 Compresser et décompresser un fichier	170

3.4	Accélérer la frappe des commandes.....	172
3.4.1	Rappel des commandes précédentes.....	172
3.4.2	Autocomplétion.....	174
3.5	Administrer le système.....	176
3.5.1	Gérer les utilisateurs et les groupes.....	176
3.5.2	Gérer les dépôts.....	181
3.5.3	Tenir le système à jour.....	182
3.5.4	Installer/supprimer un programme.....	184
3.5.5	Gérer les droits.....	186
3.5.6	Connaître l'occupation de la carte micro SD.....	192
3.5.7	Écrire un script shell.....	193
3.5.8	Planifier des tâches.....	194
3.5.9	Gérer les processus.....	196
3.5.10	Configurer la date du système.....	200
4.	Sauvegarder votre configuration.....	201
4.1	Sauvegarde de la carte micro SD sous Windows 10.....	201
4.2	Sauvegarde de la carte micro SD sous Debian.....	202
5.	Conclusion.....	203

Chapitre 8

Le GPIO du Raspberry Pi

1.	Description de l'interface GPIO.....	205
1.1	Présentation du GPIO.....	205
1.2	Connecteur GPIO J8.....	206
1.3	Caractéristiques techniques des E/S.....	209
1.3.1	Numérotation des E/S.....	209
1.3.2	Schéma de l'interface GPIO.....	210
1.3.3	Hystérésis.....	211
1.3.4	Seuil du signal d'entrée.....	211
1.3.5	Limitation du courant de sortie.....	211
1.3.6	Résistance de tirage.....	212
1.3.7	Configurations dangereuses.....	212
1.4	Liaison GPIO - breadboard.....	214
1.4.1	Fils Dupont.....	214
1.4.2	Carte T-Cobbler.....	215

8 Raspberry Pi 4 - Exploitez tout le potentiel de votre nano-ordinateur

1.4.3 Carte RasPiO PiBridge	216
2. Conclusion	217

Chapitre 9

Matériel utilisé

1. Introduction	219
2. Carte de prototypage	219
3. Câbles Dupont	221
4. Principe de la résistance	221
5. Diode LED	222
5.1 Principe de la LED	222
5.2 Résistance de protection	223
5.3 Calcul de la résistance de protection	223
5.4 Connexion d'une diode LED à un GPIO	224
6. Bouton poussoir/interrupteur	225
6.1 Utilisation du GPIO en entrée	226
6.2 Inconvénients du bouton poussoir	227
7. LED RVB	227
8. LED adressable	229
9. Anneau et ruban de LED adressables	230
10. Matrice de LED	231
10.1 Principe	231
10.2 Module 7219 + matrice	232
10.3 Chaînage des modules	232
10.4 Activation du bus SPI	233
11. Servomoteur	233
11.1 Principe	233
11.2 Commande PWM	234
11.3 Connexion du servomoteur	234
11.4 Connexion au Raspberry Pi	235
12. Codeur incrémental	235
13. Matériel complémentaire	237

Chapitre 10

Choix du langage

1. Quel langage pour développer ?	239
1.1 Une nébuleuse de langages	239
1.1.1 Langages interprétés	239
1.1.2 Langages compilés	240
1.2 Le choix de l'Éducation nationale	240
1.3 Le langage Python	241
2. Les bibliothèques pour le Raspberry Pi	242
2.1 RPi.GPIO	242
2.2 RPIO.GPIO	242
2.3 WiringPi	242
2.4 pigpio	243
2.5 gpiozero	243
3. Écrire un script en Python	244
3.1 Avec un IDE	244
3.1.1 Démarrer l'IDE Thonny	244
3.1.2 Présentation de l'écran de Thonny IDE	245
3.1.3 Les possibilités de Thonny IDE	246
3.2 Thonny en mode sudo	246
3.2.1 Modifier le fichier de lancement de Thonny	247
3.2.2 Menu Programmation	248
3.3 Avec nano	248

Chapitre 11

Gérer le GPIO avec Python

1. Introduction	249
2. Allumer une LED	249
2.1 Connecter la LED au GPIO	250
2.2 Piloter la LED en Python	250
2.2.1 En Shell	250
2.2.2 Chargement partiel de gpiozero	251
2.2.3 Par script Python	252
2.3 Conclusion	253

10 Raspberry Pi 4 - Exploitez tout le potentiel de votre nano-ordinateur

3. Exécuter un script Python en shell	254
3.1 Exécuter directement le script	254
3.1.1 Le shebang #!	254
3.1.2 Rendre le script exécutable	254
3.2 Différence IDE/ligne de commande	255
4. Utiliser un bouton poussoir	256
4.1 Connecter le bouton poussoir au GPIO	256
4.2 Lire la position du bouton poussoir	258
4.2.1 Afficher la position du bouton poussoir	258
4.2.2 Attendre l'appui sur le bouton	259
4.3 Exécuter une fonction	261
4.3.1 Fonction en Python	261
4.3.2 Fonction Python et bouton poussoir	262
4.4 Conclusion	264
5. Allumer une LED avec le bouton poussoir	264
5.1 Méthode "classique"	264
5.2 Méthode "rapide"	265
6. Projet 1 : Réaliser un feu tricolore	266
6.1 Cahier des charges	266
6.2 Câblage du feu tricolore	266
6.3 Script du feu tricolore	267
6.3.1 Feu tricolore - version 1	267
6.3.2 Feu tricolore - version 2	268
6.3.3 Feu tricolore pour passage piéton	269
7. Variation de luminosité de la LED	271
7.1 Clignotement	271
7.1.1 Par défaut	271
7.1.2 Modification du clignotement	272
7.2 Variation continue de luminosité	272
7.2.1 MLI - PWM	272
7.2.2 Pulsation	273
7.2.3 Variation programmée	274
8. Projet 2 : Variation de Luminosité	275
8.1 Cahier des charges	275
8.2 Câblage du variateur de luminosité	275
8.3 Analyse du déroulement du script	276

8.4	Écriture du script	277
8.5	Les particularités	278
8.5.1	Virgule flottante	278
8.5.2	Inégalité	278
9.	Projet 3 : Jeu de réflexes	279
9.1	Cahier des charges	279
9.2	Câblage du jeu de réflexes	279
9.3	Écriture du script	279
9.4	Amélioration du script	280
9.5	Une solution possible	280
10.	LED tricolore RVB	281
10.1	Brochage de la LED RVB	282
10.2	Câblage de la LED RVB	282
10.3	Script Python pour tester la LED RVB	283
10.4	Changer la couleur de la LED	284
10.4.1	Cahier des charges	284
10.4.2	Câblage	285
10.4.3	Script du changement de couleur	285
11.	LED RVB adressable	286
11.1	Bibliothèque néopixel	287
11.2	Connexion des LED RGB	287
11.3	Test des LED WS2812	288
11.4	Allumer les LED d'un ruban/anneau	288
11.4.1	Cahier des charges	288
11.4.2	Amélioration du programme	290
11.4.3	Proposition de solution	290
11.4.4	Allumer une LED à la fois	291
11.4.5	Effet "trainée lumineuse"	292
12.	Gestion d'une matrice de LED	295
12.1	Installation de la bibliothèque	295
12.2	Connexion des matrices	296
12.3	Test avec un exemple existant	297
12.4	Police de caractères 437	298
12.5	Insérer un caractère dans une chaîne	300
12.5.1	Insertion d'un caractère spécial chr()	300
12.5.2	Insertion au format hexadécimal	300

12 Raspberry Pi 4 - Exploitez tout le potentiel de votre nano-ordinateur

13. Projet 4 : Journal lumineux	300
13.1 Cahier des charges	300
13.2 Câblage du journal lumineux	301
13.3 Script du journal lumineux	301
14. Piloter un servomoteur	302
14.1 Câblage	303
14.2 Script de commande du servomoteur	303
14.2.1 Positions prédéterminées	303
14.2.2 Positions variables	304
15. Codeur rotatif incrémental	305
15.1 Cahier des charges	305
15.2 Branchement au GPIO	305
15.3 Programme	305
15.4 Amélioration du programme	306
16. Projet 5 : Défilement de couleurs	308
16.1 Cahier des charges	308
16.2 Câblage	308
16.3 Script de défilement de couleurs	308
17. Carte d'acquisition analogique	310
17.1 Choix de la carte	310
17.2 Montage	311
17.3 Test	312
17.3.1 Composants utilisés	312
17.3.2 Schéma de test	312
17.3.3 Câblage	313
17.4 Script de mesure	313
18. Conclusion	314

Chapitre 12

Station Météo

1. Présentation	315
2. Cahier des charges	316
3. Abri Stevenson	316

4. Température, humidité, pression.....	318
4.1 Principe.....	318
4.1.1 Température, humidité, pression.....	318
4.1.2 Composés organiques volatils.....	319
4.2 Bus I2C et bus SPI.....	321
4.3 Connexion du BME680.....	322
4.4 Test du bus I2C.....	322
4.5 Installation des dépendances.....	323
4.6 Script de mesure TPH.....	324
4.6.1 Affichage des mesures T, P, H.....	324
4.6.2 Réglage des mesures T, P, H.....	325
4.6.3 Valeurs permises.....	326
4.7 Script de mesure des VOC.....	326
4.7.1 Activation du capteur VOC.....	327
4.7.2 Mesure de la résistance.....	327
4.7.3 Affichage des mesures.....	328
4.8 Projet 6 : Mesures climatiques.....	329
4.8.1 Cahier des charges.....	329
4.8.2 Connexion du BME680.....	329
4.8.3 Script de lecture/affichage.....	329
5. Anémomètre.....	331
5.1 Principe.....	331
5.2 ILS.....	331
5.3 Fonctionnement.....	332
5.4 Connectique de l'anémomètre.....	333
5.5 Carte de connexion Dev-Lex.....	334
5.6 Test de l'anémomètre.....	335
5.7 Mesure de la vitesse du vent.....	336
5.8 Étalonnage de l'anémomètre.....	340
6. Pluviomètre.....	341
6.1 Principe.....	341
6.2 Fonctionnement.....	342
6.3 Connectique du pluviomètre.....	343
6.4 Test du pluviomètre.....	344
6.5 Mesure d'un niveau de pluie.....	344

14 Raspberry Pi 4 - Exploitez tout le potentiel de votre nano-ordinateur

7. Girouette	348
7.1 Présentation	348
7.2 Principe	349
7.3 Connectique de la girouette	352
7.4 Test de la girouette	352
7.5 Mesure de la direction du vent	354
7.5.1 Conventions	354
7.5.2 Moyenne	355
7.5.3 Script de la girouette	356
8. Projet 6 : Station météo	359
8.1 Démarche progressive	359
8.2 Analyse des mesures	359
8.3 Connexion des éléments	360
8.4 Logigramme du script	361
8.5 Script de gestion de la station météo	362

Chapitre 13

Exporter les données

1. Par SMS	369
1.1 Présentation	369
1.2 Carte NadHAT	370
2. Avec Sigfox	371
3. Avec LoRa - LoRaWAN	372
4. Sur une page web	373
5. Sur un réseau de partage	373

Chapitre 14

Interface graphique

1. Choix de l'interface	375
2. Désactiver l'écran de veille	376
3. InfluxDB	376
3.1 Présentation	376
3.2 Installation d'InfluxDB	377
3.3 Configuration d'InfluxDB	378
3.3.1 Adresse IP et port	378
3.3.2 Base de données et utilisateur	379
4. Chronograf	380
4.1 Présentation	380
4.2 Installation	380
4.3 Configuration de la BDD	381
4.3.1 Suppression d'une BDD dans Chronograf	381
4.3.2 Création d'une BDD dans Chronograf	382
4.4 Connexion à la base de données	383
4.5 Installation de la bibliothèque InfluxDB	384
4.6 Script de test de la BDD	384
4.7 Création d'un tableau de bord	385
4.8 Ajout de cellules au tableau de bord Chronograf	387
5. Liaison avec le script météo	388
5.1 Script final	388
5.2 Exécution du script	394
5.3 Création d'un tableau de bord	395
5.4 Retour sur l'utilisation de Python	395
5.5 Démarrage automatique du script	396
6. Conclusion	397
Index	399