

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 **EIPYTDAT** dans la zone de recherche et validez. Cliquez sur le titre du livre puis sur le bouton de téléchargement.

Chapitre 1

Avant-propos et introduction

1. Avant-propos	11
2. Python et l'analyse de données	12
2.1 L'explosion des données	12
2.2 L'analyse de données	13
2.3 R et Python pour l'analyse de données	14
3. Connaître les sources de données libres	15
3.1 Kaggle	15
3.2 Les données gouvernementales	17
4. Déroulement du livre	18

Chapitre 2

Mise en place de l'environnement de travail

1. Introduction : pourquoi utiliser Python pour la Data Science ?	19
2. Introduction à IPython et Jupyter	20
2.1 Introduction à IPython	20
2.2 Le projet Jupyter	22
3. Qu'est-ce qu'Anaconda ?	23
4. Installation d'Anaconda	24
4.1 Installation sur Windows	24
4.2 Installation sur MacOS	26
4.3 Installation sur Linux	26

2 Python pour la Data Science

Analysez vos données par la pratique

5. Découverte d'Anaconda Navigator	28
5.1 Applications disponibles dans Anaconda Navigator	28
5.2 Gestion des packages et environnements	29
6. Prise en main de Jupyter Notebook	32
6.1 Tableau de bord de Jupyter Notebook	32
6.2 Premiers pas avec les notebooks	34
6.3 Comprendre l'interface des notebooks	38
6.3.1 La barre de menus	38
6.3.2 La barre d'outils	40
6.3.3 Les cellules	41
6.3.4 Les modes Commande et Edition et les raccourcis-clavier	42
6.3.5 Les bases du langage Markdown pour écrire dans un notebook	45
6.3.6 Partager son notebook	51
7. Les packages Python essentiels pour la Data Science	52
7.1 NumPy	52
7.2 Pandas	53
7.3 Matplotlib	53
7.4 Seaborn	53

Chapitre 3

Rappels sur le langage Python

1. Introduction sur le langage de programmation Python	55
2. Les variables	56
3. Les différents types de données (int, float, bool, str)	58
3.1 Les nombres réels et entiers	58
3.2 Les booléens	59
3.3 Les chaînes de caractères	60

- 4. Les structures de données basiques (listes, tuples et dictionnaires) . 62
 - 4.1 Les listes 62
 - 4.1.1 Créer une liste 62
 - 4.1.2 Accéder aux éléments d'une liste 63
 - 4.1.3 Ajouter et supprimer des éléments à une liste. 66
 - 4.2 Les tuples. 70
 - 4.3 Les dictionnaires 72
 - 4.3.1 Introduction aux dictionnaires. 72
 - 4.3.2 Ajouter, modifier et supprimer des éléments d'un dictionnaire 74
 - 4.3.3 Parcourir un dictionnaire 76
- 5. Les opérateurs arithmétiques, relationnels et logiques 78
 - 5.1 Les opérateurs arithmétiques. 78
 - 5.2 Les opérateurs relationnels et logiques 80
 - 5.2.1 Les opérateurs relationnels 80
 - 5.2.2 Les opérateurs logiques 81
- 6. Vocabulaire en Python : fonctions, méthodes, attributs, modules et librairies (packages) 82
 - 6.1 Fonctions. 82
 - 6.2 Méthodes. 84
 - 6.3 Attributs 85
 - 6.4 Modules. 85
 - 6.5 Librairies (packages) 87
- 7. Instructions de condition if et boucles for 88
 - 7.1 Instruction de condition if. 88
 - 7.2 Boucle for 90

4 Python pour la Data Science

Analysez vos données par la pratique

Chapitre 4

Maîtriser la librairie NumPy

1. Introduction à NumPy	93
2. Les tableaux NumPy	94
2.1 Créer un ndarray	94
2.1.1 Créer un ndarray à partir de listes	94
2.1.2 Créer un ndarray grâce à des fonctions NumPy	97
2.1.3 Créer un ndarray à partir d'un fichier	100
2.2 Indexation	103
2.2.1 Indexation simple	103
2.2.2 Indexation booléenne	105
2.2.3 Fancy indexing	108
2.3 Accéder aux éléments par tranche (slicing)	110
2.3.1 Slicing sur un tableau NumPy à 1 dimension	110
2.3.2 Slicing sur un tableau NumPy à 2 dimensions	112
2.4 Notion de vue et copie	114
3. Les opérations mathématiques avec NumPy	116
3.1 Les opérations arithmétiques	116
3.2 Les fonctions d'agrégations	119
4. Inspecter un tableau grâce aux attributs de NumPy	122
5. Manipuler des tableaux NumPy	124
5.1 Ajouter et supprimer des éléments dans un tableau	124
5.1.1 Ajouter des éléments dans un tableau	124
5.1.2 Supprimer des éléments d'un tableau	127
5.2 Diviser un tableau NumPy (split, hsplit et vsplit)	128
5.2.1 Sur un tableau à une dimension	129
5.2.2 Sur un tableau à deux dimensions	130
5.3 Concaténer/combiner des tableaux	131
5.3.1 La fonction concatenate()	131
5.3.2 Les fonctions vstack() et hstack()	133
6. Introduction aux matrices avec NumPy	135

Chapitre 5
Maîtriser la librairie Pandas

- 1. Introduction 137
 - 1.1 Introduction à la librairie Pandas. 137
 - 1.2 Introduction au jeu de données utilisé pour les exemples ... 140
- 2. Lire et écrire des fichiers avec Pandas 143
 - 2.1 Lecture de fichiers texte (CSV ou TXT) 143
 - 2.1.1 Lecture basique d'un fichier 143
 - 2.1.2 Gestion de l'en-tête 146
 - 2.1.3 Gestion des index. 148
 - 2.1.4 Création d'un tableau à une dimension
à partir du fichier. 150
 - 2.1.5 Filtrage des colonnes lors de la lecture du fichier 150
 - 2.1.6 Les types des différentes colonnes 152
 - 2.1.7 Gestion des dates lors de la lecture du fichier 152
 - 2.2 Lecture de fichiers Excel. 153
 - 2.3 Importation des données à partir d'une base de données. 156
 - 2.4 Lecture de fichiers au format JSON. 158
 - 2.5 Écriture de fichiers ou exportation de données. 160
- 3. Structure de données Pandas : les Series (Séries) 162
 - 3.1 Introduction 162
 - 3.2 Créer des séries 163
 - 3.2.1 À partir de valeurs aléatoires. 163
 - 3.2.2 À partir d'une liste Python 165
 - 3.2.3 À partir d'un tableau NumPy (ndarray) 168
 - 3.2.4 À partir d'un fichier texte 168
 - 3.3 Choisir l'index d'une série. 169
 - 3.4 Accéder aux valeurs d'une série 171
 - 3.4.1 Indexing via la position des valeurs 171
 - 3.4.2 Indexing via l'étiquette des valeurs 172
 - 3.4.3 Les indexeurs loc et iloc. 173
 - 3.4.4 Indexing via une expression booléenne 175

6 — Python pour la Data Science

Analysez vos données par la pratique

3.4.5	Slicing : découpage de valeurs successives	178
3.5	Les attributs et les méthodes des objets de classe Series	184
3.5.1	Les attributs des objets de classe Series	184
3.5.2	Les méthodes des objets de classe Series	185
3.6	Ajouter, supprimer et modifier les valeurs d'une série	187
3.6.1	Ajouter des valeurs à une série	187
3.6.2	Supprimer une valeur d'une série	188
3.6.3	Modifier les valeurs d'une série	189
4.	Structure de données Pandas : les objets de type DataFrame	191
4.1	Introduction	191
4.2	Indexing : sélectionner des valeurs d'un dataframe	193
4.2.1	Indexing et slicing avec l'attribut loc	194
4.2.2	Indexing et slicing avec l'attribut iloc	198
4.2.3	Indexing avec une expression booléenne	199
4.3	Ajout, suppression et modification sur un dataframe	201
4.3.1	Ajouter une ou plusieurs colonnes à un dataframe	201
4.3.2	Ajouter une ligne à un dataframe	202
4.3.3	Supprimer des lignes ou colonnes d'un dataframe	205
4.3.4	Modifier des valeurs dans un dataframe	207
4.4	Nettoyage et préparation des données avec Pandas	209
4.4.1	Gestion des données manquantes	210
4.4.2	Gestion des données dupliquées	215
4.5	Exploration préliminaire d'un dataframe	219
4.5.1	Principaux attributs	219
4.5.2	Définition des termes variable, variable quantitative et variable qualitative et découverte de la méthode describe()	222
4.5.3	Méthodes de tri d'un dataframe	225
5.	Structure de données Pandas : les panels	227
6.	Manipulation avancée des données avec Pandas	228
6.1	Les opérations groupby	228
6.1.1	groupby sur une colonne	228
6.1.2	groupby sur plusieurs colonnes	231

6.1.3 Appliquer plusieurs fonctions avec la méthode groupby et la méthode aggregate	232
6.2 Appliquer une fonction à un dataframe avec la méthode apply	234
6.3 Remodeler/réorganiser des dataframes	236
6.3.1 Pivotage : la méthode pivot_table	236
6.3.2 Les méthodes stack (empiler) et unstack (désempiler) .	238

Chapitre 6

Maîtriser la librairie Matplotlib

1. Introduction	241
2. Le fonctionnement de Matplotlib	242
2.1 Architecture de Matplotlib	242
2.2 Organisation des figures avec Matplotlib	244
3. La création d'un premier graphique simple	246
3.1 Préparer son jeu de données	246
3.2 Créer un nuage de points	250
3.3 Ajouter un titre principal et des labels aux axes du nuage de points	254
3.4 Enregistrer son graphique	256
3.5 Changer la taille de la fenêtre graphique et la résolution de son graphique	258
3.6 Tracer plusieurs courbes sur un même graphique (sur un même objet axes)	260
3.7 Ajouter une légende à son graphique	263
3.8 Annoter son graphique avec du texte	265
3.9 Combiner plusieurs graphiques grâce à subplot et subplots .	268
3.9.1 Tracer des sous-graphiques (subplot) sur une ligne ou une colonne	268
3.9.2 Tracer des sous-graphiques sur plusieurs lignes et plusieurs colonnes	271
3.9.3 Incruster un objet axes dans un autre	274

8 Python pour la Data Science

Analysez vos données par la pratique

4. Les différents types de graphes	278
4.1 Types de graphiques selon les types de variables (quantitatives et qualitatives)	278
4.2 Scatterplot.	279
4.3 Graphique à barres (bargraph).	282
4.3.1 Graphique à barres simple.	282
4.3.2 Graphique à barres groupées.	287
4.3.3 Graphique à barres empilées	294
4.4 Boxplots	296

Chapitre 7

Maîtriser la librairie Seaborn

1. Introduction	301
2. L'esthétique des figures avec Seaborn (Aesthetic)	303
2.1 Paramétrer les styles Seaborn (thèmes).	304
2.2 Supprimer les axes	306
2.3 Paramétrer les contextes avec Seaborn	309
2.4 Les palettes de couleur avec Seaborn.	314
2.4.1 Choisir une palette de couleurs existante	314
2.4.2 Créer sa propre palette de couleurs	317
3. Les différents types de graphiques.	318
3.1 Préparation du jeu de données.	318
3.2 Nuage de points (scatterplot)	318
3.3 Graphiques de régression	323
3.4 Pointplot	327
3.5 Nuage de points avec une variable qualitative : stripplot	331
3.6 Boxplots	334
3.7 Graphique à barres : countplot	336
3.8 Histogrammes.	339
3.9 Jointplot	344
3.10 Pairplot	346
3.11 Heatmap	349

- 4. Les graphiques multi-grilles 352
 - 4.1 FacetGrid 352
 - 4.2 PairGrid 354
 - 4.3 JointGrid 359
- 5. Conclusion 361

Chapitre 8

Exercice complet sur jeu de données réel

- 1. Introduction 363
- 2. Présentation du jeu de données 365
- 3. Énoncé de l'exercice 366
 - 3.1 Lire le fichier 366
 - 3.2 Afficher les dimensions du dataframe 367
 - 3.3 Compter les films et les séries 367
 - 3.4 Générer le résumé statistique du dataframe 367
 - 3.5 Compter les valeurs manquantes 368
 - 3.6 Explorer les valeurs manquantes 368
 - 3.6.1 Sur la colonne des directeurs de production 368
 - 3.6.2 Sur la colonne des acteurs 368
 - 3.7 Supprimer les lignes dupliquées 369
 - 3.8 Compter les films/séries produits
par les États-Unis et par la France 369
 - 3.9 Afficher le contenu le plus vieux disponible sur Netflix 370
 - 3.10 Afficher le film avec la durée la plus longue sur Netflix 370
 - 3.10.1 Nouvelle notion : les méthodes str. 370
 - 3.10.2 Énoncé 371
 - 3.11 Étudier les catégories avec le plus de contenu 372
 - 3.12 Afficher les directeurs qui ont produit
le plus de films/séries disponibles sur Netflix. 375
 - 3.13 Voir si Jan Suter travaille souvent avec les mêmes acteurs ... 375

10 _____ Python pour la Data Science

Analysez vos données par la pratique

3.14 Représenter les dix pays qui ont produit le plus de contenus disponibles sur Netflix, avec le nombre de contenus par pays	376
3.15 Tracer un graphe à barres du nombre de films/séries par classement de contenu (rating)	377
3.16 Afficher l'évolution du nombre de films/séries disponibles sur Netflix au cours du temps	377
3.16.1 Notions supplémentaires sur les dates	377
3.16.2 Énoncé	377
3.17 Afficher la distribution de la durée des films disponibles sur Netflix	378
3.18 Tracer un graphique représentant le nombre de séries par modalité de nombre de saisons	379
Index	381