

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

## Chapitre 1

# Présentation et bases du langage Python

1. Introduction .....	15
2. Présentation de Python .....	17
2.1 Le langage .....	17
2.2 Installation de Python .....	18
2.2.1 Installation de Python sur Windows .....	18
2.2.2 Installation de Python sur Mac OS .....	18
2.2.3 Installation de Python sur Linux .....	18
3. Les bases de Python .....	20
3.1 L'interpréteur de commande .....	20
3.2 Les variables .....	20
3.3 De l'interpréteur de commande à l'usage d'un fichier .....	21
3.4 Installation des modules avec pip .....	21
3.5 Les chaînes de caractères .....	22
3.5.1 La fonction print .....	22
3.5.2 Une chaîne de caractères est comme une collection .....	23
3.5.3 La taille d'une chaîne de caractères .....	23
3.5.4 Premières manipulations de chaînes de caractères .....	23
3.6 Les listes .....	24
3.7 Les conditions .....	25
3.8 Les boucles .....	26
3.8.1 La boucle for .....	26
3.8.2 La boucle while .....	26
3.8.3 Le mot-clé break .....	27
3.9 L'environnement virtuel en bref .....	27
3.10 Développement d'un petit jeu en ligne de commande .....	27

## 2 Pygame - Initiez-vous au développement de jeux vidéo en Python

### Chapitre 2

## Premiers pas avec Pygame

1. La boucle de jeu .....	31
2. Présentation de Pygame .....	32
3. Installation de Pygame .....	32
4. Les modules composant Pygame .....	33
5. Réalisation d'un premier jeu graphique : Fusée et planètes .....	34
5.1 Les images utilisées .....	35
5.2 La fenêtre du jeu .....	35
5.3 La boucle du jeu .....	36
5.4 Le système de coordonnées Pygame .....	37
5.5 Les variables du jeu .....	38
5.5.1 Les variables liées à la fusée .....	38
5.5.2 Les variables liées aux deux planètes qui « tombent » .....	39
5.5.3 Les variables relatives au comptage des points .....	39
5.5.4 Les variables relatives aux images .....	39
5.6 Les déplacements de la fusée .....	41
5.7 Les déplacements des planètes .....	42
5.8 Les collisions .....	43
5.9 Le code complet .....	44

### Chapitre 3

## La structure d'un jeu Pygame

1. Introduction .....	47
2. Initialisation .....	47
3. Aide en ligne de commande .....	48
4. Affichage de la fenêtre .....	48
4.1 Le tuple size de set_mode .....	49
4.2 Le paramètre flags de set_mode .....	49
5. Rappels concernant la boucle de jeu .....	50



6. Les surfaces Pygame .....	50
6.1 Définition d'une surface .....	50
6.2 La fenêtre de jeu, une surface particulière .....	51
6.3 Exemple de manipulation d'une surface .....	51
6.4 Manipulation de la surface .....	52
6.5 Surface ou copie de surface ? .....	52
6.6 Coloration de la surface .....	53
7. Gestion des couleurs .....	53
8. Système de coordonnées .....	54
9. Gestion du temps et des évènements .....	57
9.1 Gestion du temps dans Pygame .....	57
9.2 Gestion des évènements dans Pygame .....	58
9.2.1 La fonction <code>pygame.event.get</code> .....	58
9.2.2 La fonction <code>pygame.event.wait</code> .....	59
9.2.3 La fonction <code>pygame.event.poll</code> .....	59
9.3 Un exemple : le carré qui rebondit .....	59

## Chapitre 4

# Le dessin et le graphisme dans tous ses états avec Pygame

1. Introduction .....	63
2. Dessiner des formes avec Pygame .....	64
2.1 Le module <code>pygame.draw</code> .....	64
2.2 Dessiner une ligne .....	64
2.3 Dessiner une ligne brisée .....	65
2.4 Dessiner un rectangle .....	66
2.5 Dessiner un polygone .....	66
2.6 Dessiner un cercle .....	67
2.7 Dessiner une ellipse .....	67
2.8 Dessiner un arc de cercle .....	68
2.9 L'anti-aliasing .....	71

## 4 Pygame - Initiez-vous au développement de jeux vidéo en Python

3. Afficher et sauvegarder des images avec Pygame.....	72
3.1 Le module pygame.image.....	72
3.2 Charger des images avec Pygame.....	73
3.2.1 La fonction pygame.draw.load.....	73
3.2.2 La bonne pratique de l'appel de convert().....	73
3.2.3 Exemple de chargement et d'affichage d'une image.....	74
3.3 Sauver des images avec Pygame.....	75
4. Utiliser et manipuler du texte dans Pygame.....	77
4.1 Le module pygame.font.....	77
4.2 Utiliser des polices de caractères système avec Pygame.....	77
4.3 Utiliser ses propres polices de caractères avec Pygame.....	79
5. Concevoir un logiciel de dessin avec Pygame.....	79
5.1 Gérer les événements de la souris ou du clavier en Pygame.....	79
5.2 Première version du logiciel : afficher le tracé réalisé avec la souris.....	81
5.3 Seconde version du logiciel : améliorations diverses.....	82
5.3.1 Amélioration de l'expérience du tracé.....	82
5.3.2 Ajout de fonctionnalités (couleur, épaisseur, etc.).....	83
6. Appliquer des transformations géométriques dans Pygame.....	85
6.1 Le module pygame.transform.....	85
6.2 Exemple d'utilisation des transformations Pygame.....	86

## Chapitre 5

### L'ajout de sons dans un jeu Pygame

1. Introduction.....	91
2. La gestion du son avec Pygame.....	91
2.1 Les modules pygame.mixer et pygame.mixer.music.....	92
2.1.1 Le module pygame.mixer.music (fond sonore).....	92
2.1.2 Le module pygame.mixer (effets sonores).....	92
2.2 Les fichiers son.....	92
2.3 La notion de channel (canal) dans Pygame.....	93
3. Exemple d'utilisation du son avec Pygame.....	94

## Chapitre 6

**Les sprites avec Pygame**

1. La notion de sprite dans Pygame.....	97
2. La notion de group dans Pygame.....	98
3. Une gestion des collisions simplifiée.....	98
4. Quelques explications sur la programmation orientée objet.....	99
4.1 Le paradigme objet, les grandes lignes.....	99
4.2 L'héritage.....	100
4.3 Des mots-clés fondamentaux en Python.....	101
4.3.1 Le mot-clé self.....	101
4.3.2 Le mot-clé class.....	101
4.3.3 Le mot-clé def.....	102
4.3.4 <code>__init__</code> .....	102
4.4 L'exemple de la classe Voiture en Python.....	102
4.5 Ce que l'on savait déjà... sans le savoir.....	104
5. Le module sprite et son utilisation.....	104
5.1 Le contenu du module sprite.....	104
5.2 Création d'un sprite.....	105
5.2.1 Premier exemple d'utilisation.....	105
5.2.2 Retour sur le premier exemple.....	109
5.3 Gros plan sur l'attribut <code>rect</code> de la classe <code>Sprite</code> .....	109
5.4 La liste de sprites ( <code>group</code> ).....	110
5.5 Gestion des collisions grâce aux sprites.....	111

## Chapitre 7

**Plus loin avec le module `sprite`, exemples appliqués**

1. Introduction.....	113
2. Le jeu du serpent ( <code>snake</code> ).....	114
2.1 Le contexte.....	114
2.2 Les images utilisées.....	114
2.3 Les effets sonores utilisés.....	115
2.4 Le programme global.....	115
2.5 Les listes de sprites ( <code>group</code> ).....	119

## 6 Pygame - Initiez-vous au développement de jeux vidéo en Python

2.6	Les variables globales .....	119
2.7	Les classes .....	120
2.7.1	La classe SERPENT .....	120
2.7.2	La classe CORPS .....	123
2.7.3	La classe NOURRITURE .....	124
2.8	Le programme lui-même .....	124
2.8.1	La fonction AFFICHER_SCORE .....	124
2.8.2	Le corps du programme .....	125
3.	Le jeu du labyrinthe .....	127
3.1	Le contexte .....	127
3.2	Conception du labyrinthe .....	127
3.3	Les images utilisées .....	128
3.4	Le programme global .....	129
3.5	Les listes de sprites (group) .....	133
3.6	Les classes .....	133
3.6.1	La classe MUR .....	133
3.6.2	La classe OBJET .....	133
3.6.3	La classe Chrono .....	134
3.6.4	La classe PERSONNAGE .....	134
3.7	Le programme lui-même .....	135
4.	Le jeu de casse-briques .....	136
4.1	Le contexte .....	136
4.2	Les images utilisées .....	137
4.3	Le programme global .....	137
4.4	Les listes de sprites (group) .....	140
4.5	Les constantes .....	141
4.6	Les classes .....	141
4.6.1	La classe OBJET .....	142
4.6.2	La classe RAQUETTE .....	143
4.6.3	La classe BRIQUE .....	144
4.6.4	La classe BALLE .....	144
4.7	Le programme lui-même .....	145
5.	Le jeu de défilement : Fusée et planètes (version 2) .....	148
5.1	Le contexte .....	148
5.2	Le programme global .....	149
5.3	Les listes de sprites (group) .....	151

5.4 Les classes .....	151
5.4.1 La classe FUSEE .....	151
5.4.2 La classe PLANETE .....	152
5.5 Le programme lui-même .....	153

## Chapitre 8

### Introduction à la 3D et à la notion de moteur de jeu

1. Travailler en 3D avec Pygame .....	155
1.1 La bibliothèque 3D OpenGL .....	155
1.2 OpenGL en Python/Pygame .....	156
1.2.1 PyOpenGL .....	156
1.2.2 Les notions fondamentales : sommet et arête .....	156
1.2.3 PyOpenGL et Pygame .....	157
1.3 PyOpenGL/Pygame : l'exemple du cube .....	157
1.3.1 Le code global .....	157
1.3.2 Explication détaillée du code .....	159
1.4 Plus loin avec PyOpenGL/Pygame : l'exemple du cube (suite) .....	162
1.4.1 Le code global .....	162
1.4.2 Explication détaillée du code .....	164
2. La notion de moteur de jeu vidéo .....	165
2.1 Définition .....	165
2.2 Créer son propre moteur de jeu ? .....	166
2.3 Une ébauche de moteur de jeu .....	166

## Chapitre 9

### Les principaux modules Pygame

1. Introduction .....	169
2. L'objet Color .....	170
2.1 La classe Color .....	170
2.2 Les constructeurs de Color .....	171
2.3 Les principales fonctions de la classe Color .....	171

## 8 Pygame - Initiez-vous au développement de jeux vidéo en Python

2.4	Les fonctions associées aux autres représentations de la couleur .....	172
2.4.1	La représentation CMY .....	172
2.4.2	La représentation HSV et HSL .....	172
2.4.3	La représentation lll2l3 .....	172
2.5	Les autres fonctions .....	172
2.5.1	La fonction normalize .....	172
2.5.2	La fonction correct_gamma .....	173
2.5.3	La fonction set_length .....	173
3.	Le module time .....	173
3.1	La fonction get_ticks .....	173
3.2	La fonction wait .....	174
3.3	La fonction delay .....	174
3.4	La fonction set_timer .....	174
3.5	L'objet Clock .....	175
3.5.1	Création d'une instance .....	175
3.5.2	La fonction tick .....	175
3.5.3	La fonction get_time .....	176
3.5.4	La fonction get_fps .....	176
4.	Le module event .....	177
4.1	La fonction pump .....	177
4.2	La fonction get .....	178
4.3	La fonction poll .....	178
4.4	La fonction wait .....	178
4.5	La fonction peek .....	179
4.6	La fonction clear .....	179
4.7	La fonction event_name .....	179
4.8	La fonction set_blocked .....	180
4.9	La fonction set_allowed .....	180
5.	Le module display .....	180
5.1	La fonction init .....	180
5.2	La fonction quit .....	181
5.3	La fonction get_init .....	181
5.4	La fonction set_mode .....	181
5.5	La fonction flip .....	182
5.6	La fonction update .....	182
5.7	La fonction set_icon .....	182

5.8	La fonction <code>set_caption</code> .....	182
6.	L'objet <code>Surface</code> .....	183
6.1	Le constructeur de <code>Surface</code> .....	183
6.2	La fonction <code>blit</code> .....	183
6.3	La fonction <code>blits</code> .....	184
6.4	Les fonctions <code>convert</code> et <code>convert_alpha</code> .....	185
6.5	La fonction <code>copy</code> .....	185
6.6	La fonction <code>fill</code> .....	186
6.7	La fonction <code>scroll</code> .....	186
6.8	La fonction <code>set_colorkey</code> .....	187
6.9	La fonction <code>get_colorkey</code> .....	187
7.	Le module <code>draw</code> .....	188
7.1	La fonction <code>rect</code> .....	188
7.2	La fonction <code>polygon</code> .....	188
7.3	La fonction <code>circle</code> .....	188
7.4	La fonction <code>ellipse</code> .....	189
7.5	La fonction <code>arc</code> .....	189
7.6	La fonction <code>line</code> .....	190
7.7	La fonction <code>lines</code> .....	190
7.8	Les fonctions <code>aaline</code> et <code>aalines</code> .....	190
8.	Le module <code>image</code> .....	191
8.1	La fonction <code>load</code> .....	191
8.2	La fonction <code>save</code> .....	191
8.3	Les fonctions <code>tostring</code> , <code>fromstring</code> , <code>frombuffer</code> .....	192
9.	Le module <code>font</code> .....	192
9.1	La fonction <code>init</code> .....	192
9.2	La fonction <code>quit</code> .....	192
9.3	La fonction <code>get_init</code> .....	192
9.4	La fonction <code>get_default_font</code> .....	193
9.5	La fonction <code>get_fonts</code> .....	193
9.6	La fonction <code>match_font</code> .....	193
9.7	La fonction <code>SysFont</code> .....	194
9.8	L'objet <code>Font</code> .....	194
9.8.1	La fonction <code>Font</code> .....	194
9.8.2	La fonction <code>render</code> .....	194
9.8.3	La fonction <code>size</code> .....	195

## 10 Pygame - Initiez-vous au développement de jeux vidéo en Python

9.8.4	La fonction set_underline.....	195
9.8.5	La fonction get_underline.....	195
9.8.6	La fonction set_bold.....	195
9.8.7	La fonction get_bold.....	195
9.8.8	La fonction set_italic.....	196
9.8.9	La fonction get_italic.....	196
9.8.10	La fonction metrics.....	196
10.	Le module mouse.....	196
10.1	La fonction get_pressed.....	196
10.2	La fonction get_pos.....	197
10.3	La fonction get_rel.....	197
10.4	La fonction set_pos.....	198
10.5	La fonction set_visible.....	198
10.6	La fonction get_visible.....	198
10.7	La fonction get_focused.....	198
10.8	La fonction set_cursor.....	198
10.9	La fonction get_cursor.....	198
11.	Le module key.....	199
11.1	Les constantes correspondant aux touches du clavier.....	199
11.2	La fonction get_focused.....	201
11.3	La fonction pressed.....	201
11.4	La fonction set_repeat.....	201
11.5	La fonction get_repeat.....	202
11.6	La fonction name.....	202
12.	Le module transform.....	202
12.1	La fonction flip.....	202
12.2	La fonction scale.....	202
12.3	La fonction rotate.....	203
12.4	La fonction rotozoom.....	203
12.5	La fonction scale2x.....	203
12.6	La fonction chop.....	203
12.7	La fonction laplacian.....	204
12.8	La fonction average_surfaces.....	205
12.9	La fonction average_color.....	205



13. Le module mixer .....	205
13.1 La fonction init .....	205
13.2 La fonction quit .....	206
13.3 La fonction get_init .....	206
13.4 L'objet Sound .....	206
13.4.1 Les constructeurs de Sound .....	206
13.4.2 La fonction play .....	206
13.4.3 La fonction stop .....	207
13.4.4 La fonction fadeout .....	207
13.4.5 La fonction set_volume .....	207
13.4.6 La fonction get_volume .....	207
13.4.7 La fonction get_num_channels .....	207
13.4.8 La fonction get_length .....	208
13.4.9 La fonction get_raw .....	208
13.5 L'objet Channel .....	209
13.5.1 Le constructeur de Channel .....	209
13.5.2 La fonction queue .....	209
13.5.3 La fonction set_volume .....	209
13.5.4 La fonction get_volume .....	209
13.5.5 Les fonctions play, stop, pause etc .....	210
13.6 Les fonctions get_num_channels, set_num_channels, find_channel .....	210
14. Le module music .....	211
14.1 La fonction load .....	211
14.2 La fonction unload .....	211
14.3 La fonction play .....	211
14.4 La fonction rewind .....	211
14.5 La fonction stop .....	212
14.6 La fonction pause .....	212
14.7 La fonction unpause .....	212
14.8 La fonction fadeout .....	212
14.9 La fonction set_volume .....	212
14.10 La fonction get_volume .....	212
14.11 La fonction set_pos .....	212
14.12 La fonction get_pos .....	213
14.13 La fonction queue .....	213

## 12 Pygame - Initiez-vous au développement de jeux vidéo en Python

15. Le module sprite.....	213
15.1 La classe Sprite.....	213
15.1.1 La fonction update.....	213
15.1.2 La fonction add.....	213
15.1.3 La fonction remove.....	213
15.1.4 La fonction kill.....	214
15.1.5 La fonction alive.....	214
15.1.6 La fonction groups.....	214
15.2 La classe Group.....	214
15.2.1 La fonction sprites.....	214
15.2.2 La fonction copy.....	214
15.2.3 La fonction add.....	214
15.2.4 La fonction remove.....	215
15.2.5 La fonction has.....	215
15.2.6 La fonction update.....	215
15.2.7 La fonction draw.....	215
15.2.8 La fonction clear.....	215
15.2.9 La fonction empty.....	215
15.3 Les principales fonctions du module.....	216
15.3.1 La fonction spritecollide.....	216
15.3.2 La fonction collide_rect.....	216
15.3.3 La fonction collide_circle.....	216
15.3.4 La fonction collide_mask.....	216
15.3.5 La fonction groupcollide.....	216
15.3.6 La fonction spritecollideany.....	217

### Chapitre 10

## Les modules secondaires Pygame

1. Introduction.....	219
2. Le module cursors.....	220
2.1 Les curseurs prédéfinis du module.....	220
2.2 La fonction compile.....	221
2.3 La fonction load_xbm.....	221

3. Le module joystick .....	222
3.1 La fonction init .....	222
3.2 La fonction quit.....	222
3.3 La fonction get_init .....	222
3.4 La fonction get_count .....	222
3.5 La classe Joystick .....	223
3.5.1 La fonction Joystick.....	223
3.5.2 La fonction init .....	223
3.5.3 La fonction quit .....	223
3.5.4 La fonction get_init .....	223
3.5.5 La fonction get_id .....	223
3.5.6 La fonction get_name .....	224
3.5.7 La fonction get_numaxes.....	224
3.5.8 La fonction get_axis .....	224
3.5.9 La fonction get_numballs.....	224
3.5.10 La fonction get_ball.....	224
3.5.11 La fonction get_numbuttons.....	225
3.5.12 La fonction get_button .....	225
3.5.13 La fonction get_numhats .....	225
3.5.14 La fonction get_hat .....	225
4. Le module touch.....	225
4.1 La fonction get_num_devices .....	225
4.2 La fonction get_device .....	226
4.3 La fonction get_num_fingers.....	226
4.4 La fonction get_finger .....	226
5. Le module math .....	226
5.1 La classe Vector2 - Création de vecteur .....	227
5.2 La classe Vector3 - Création de vecteur .....	227
5.3 Les principales fonctions de Vector2 et Vector3 .....	228
5.3.1 La fonction dot .....	228
5.3.2 La fonction length.....	228
5.3.3 La fonction normalize .....	228
5.3.4 La fonction reflect.....	228
5.3.5 La fonction distance_to .....	228
5.3.6 La fonction rotate .....	229
5.3.7 La fonction rotate_rad.....	229

## 14 Pygame - Initiez-vous au développement de jeux vidéo en Python

6. Le module surfarray .....	229
6.1 La fonction array2d .....	229
6.2 La fonction pixels_red .....	229
6.3 La fonction pixels_green .....	229
6.4 La fonction pixels_blue .....	230
6.5 La fonction make_surface .....	230
6.6 La fonction blit_array .....	230
7. Le module camera .....	230
7.1 La fonction list_cameras .....	231
7.2 L'objet Camera .....	231
7.2.1 Instanciation de Camera .....	231
7.2.2 La fonction start .....	232
7.2.3 La fonction stop .....	232
7.2.4 La fonction get_image .....	232
7.2.5 La fonction get_raw .....	232
7.2.6 La fonction query_image .....	232
7.2.7 La fonction get_size .....	232
7.2.8 La fonction get_controls .....	232
7.2.9 La fonction set_controls .....	233
Index .....	235