

Table des matières

Avant-propos	XXXI
---------------------------	-------------

PREMIÈRE PARTIE

Découverte de Python	1
-----------------------------------	----------

CHAPITRE 1

Introduction	3
---------------------------	----------

Python ?	3
Du code de qualité	4
Orienté objet	4
Portable	4
Facile à intégrer	5
Hautement productif	5
Dynamique	6
Python et les autres langages	6
Python et Perl	7
Ruby, PHP, Java...	7

CHAPITRE 2

Python pour quels usages ?.....	9
--	----------

Administration système	10
Des API simples et efficaces	10
<i>Manipuler des fichiers et des dossiers</i>	10
<i>Manipuler des programmes</i>	11
<i>Envoyer et recevoir des courriers électroniques</i>	13
<i>Échanger des informations avec d'autres systèmes</i>	15
Le match Perl-Python	17
<i>Syntaxe</i>	17
<i>Structures de données</i>	18
<i>Manipulation de texte</i>	19
<i>Conclusion</i>	20

Prototypage rapide d'applications	20
Objectif d'une maquette	20
Maquette d'interfaces	21
Maquette de bibliothèque ou Fake	22
<i>Exemple de prototype de bibliothèque</i>	22
Recherche et calcul scientifique	24
Pas de paradigme imposé	24
Facilité de prise en main	24
Création ou utilisation d'outils spécialisés	24
Applications de gestion	25
Conception d'interface utilisateur	25
Stockage de données	26
<i>Sérialisation des objets</i>	26
<i>Les bases de données relationnelles</i>	28
Applications web	29
En un mot...	30

CHAPITRE 3

Environnement de développement 31

Installation sous Linux	31
Installation par distribution	32
<i>Paquets Debian</i>	32
<i>Paquets RedHat</i>	33
<i>Distributions Mandrake et Fedora Core</i>	33
Compilation des sources	33
<i>Étapes d'installation</i>	34
<i>Options de compilation</i>	34
<i>Compilation et installation de Python</i>	36
Gérer plusieurs versions de Python	37
Installation sous MS-Windows	37
Installation sous Mac OS X	39
Premiers tests de Python en mode interactif	39
Script de démarrage du mode interactif	40
Le choix d'un éditeur	41
La coloration syntaxique	42
La standardisation automatique	43
Les raccourcis clavier et les macros	43
L'édition multiple	43
Le repliement de code et la recherche	43
L'autocomplétion	43

L'interpréteur et le débogueur embarqués	44
La licence	44
Les plates-formes reconnues	44
En un mot...	46

DEUXIÈME PARTIE

Éléments du langage **47**

CHAPITRE 4

Syntaxe du langage..... **49**

L'instruction <code>print</code>	50
<code>print</code> devient fonction	50
Les commentaires	52
Modèle de données	52
Les littéraux	54
Littéraux alphanumériques	54
<i>Normes ASCII et Unicode</i>	55
<i>Évolution de l'Unicode de Python 2 à Python 3</i>	56
<i>Caractères spéciaux</i>	57
Littéraux numériques	58
<i>Littéraux pour les entiers</i>	59
<i>Littéraux pour les valeurs à virgule flottante</i>	61
<i>Littéraux pour les nombres complexes</i>	62
Les types standards	62
Les types à valeur unique	63
<i>None</i>	63
<i>NotImplemented</i>	63
<i>Ellipsis</i>	64
Les nombres	64
<i>Les nombres entiers</i>	64
<i>Les nombres à virgule flottante</i>	65
<i>Les nombres complexes</i>	66
<i>Les décimaux</i>	66
Les séquences	66
<i>Les séquences immuables</i>	67
<i>Les séquences modifiables</i>	72
Les mappings	76
Les opérateurs	81
Opérateurs de base	81
Autres opérateurs	82

<i>Modulo</i>	82
<i>Négation</i>	83
<i>Inversion</i>	83
<i>Puissance</i>	83
<i>Appartenance</i>	83
<i>Opérateurs binaires</i>	85
Opérateurs de comparaison	85
<i>Principes de la comparaison</i>	86
Ordre de traitement des opérations	86
Construction de comparaisons complexes	87
L'indentation	87
Les structures conditionnelles	88
L'instruction if	89
L'instruction for..in	89
L'instruction while	91
L'instruction with	92
En un mot...	95

CHAPITRE 5

Structuration du code **97**

Fonctions	97
Contexte d'exécution et directive global	98
Directive return	99
Paramètres d'une fonction	100
<i>Paramètres explicites et valeurs par défaut</i>	100
<i>Les paramètres non explicites</i>	102
<i>Les paramètres arbitraires</i>	103
<i>Collisions de paramètres</i>	104
<i>Signatures multiples de fonctions</i>	104
Directive lambda	105
Documentation strings (docstrings)	105
Decorators	106
Classes	111
Définition	111
Espace de noms	112
Paramètre self	113
Héritage	113
<i>Héritage multiple</i>	114
<i>Surcharge des attributs</i>	115
<i>Constructeur et destructeur</i>	116
Attributs privés	117

Méthodes spéciales	119
<i>Représentation et comparaison de l'objet</i>	119
<i>Utilisation de l'objet comme fonction</i>	121
<i>Accès aux attributs de l'objet</i>	121
<i>Utilisation de l'objet comme conteneur</i>	122
<i>Utilisation de l'objet comme type numérique</i>	124
New-style classes	126
<i>Le nouveau Method Resolution Order</i>	127
<i>Constructeur statique</i>	128
<i>Surcharge de type() par metaclass</i>	129
<i>Descriptors</i>	130
<i>Properties</i>	131
<i>Slots</i>	132
<i>Decorators pour les classes</i>	133
Modules	133
Directive import	134
Primitive reload	135
Directives from et as	136
Paquets	137
Organisation d'un paquet	137
Import * et __all__	138
Références relatives	139
Exceptions	139
Exceptions du langage	140
<i>Classes d'exceptions de base</i>	141
<i>Classes concrètes</i>	142
try..except..else	142
try..finally	144
try..except..finally	144
Les list comprehensions	145
Generators et iterators	147
Iterators	147
Generators	148
Generator expression (genexp)	149
En un mot...	150

CHAPITRE 6

Les primitives	151
Primitives du langage	152
__import__	152
abs	153

all	153
any	153
apply	154
callable	154
chr	154
classmethod	154
cmp	156
coerce	156
compile	157
delattr	158
dir	158
divmod	160
enumerate	160
eval	160
execfile	161
exit	161
file	161
filter	163
getattr	164
globals	164
hasattr	165
hash	165
help	165
hex	166
id	166
input	167
int	167
intern	167
isinstance	168
issubclass	168
iter	169
len	169
license	170
list	171
locals	171
map	171
max	172
min	172
oct	173
open	173
ord	173

pow	174
property	174
quit	175
range	175
raw_input	176
reduce	177
reload	177
repr	177
round	178
set	178
setattr	179
slice	179
sorted	180
staticmethod	181
str	182
sum	182
super	183
type	184
type	184
unichr	185
unicode	185
vars	186
xrange	186
zip	186
Exceptions du langage	187
Erreurs	187
<i>AssertionError</i>	187
<i>AttributeError</i>	188
<i>EOFError</i>	188
<i>FloatingPointError</i>	188
<i>IOError</i>	188
<i>ImportError</i>	189
<i>IndentationError</i>	189
<i>IndexError</i>	189
<i>KeyError</i>	189
<i>KeyboardInterrupt</i>	190
<i>MemoryError</i>	190
<i>NameError</i>	190
<i>NotImplementedError</i>	190
<i>OSError</i>	191
<i>OverflowError</i>	191

<i>ReferenceError</i>	191
<i>RuntimeError</i>	191
<i>StopIteration</i>	191
<i>SyntaxError</i>	191
<i>SystemError</i>	192
<i>SystemExit</i>	192
<i>TabError</i>	192
<i>TypeError</i>	192
<i>UnboundLocalError</i>	193
<i>UnicodeEncodeError</i>	193
<i>UnicodeDecodeError</i>	193
<i>UnicodeTranslateError</i>	194
<i>ValueError</i>	194
<i>WindowsError</i>	194
<i>ZeroDivisionError</i>	194
Avertissements	194
<i>UserWarning</i>	195
<i>DeprecationWarning</i>	195
<i>FutureWarning</i>	195
<i>OverflowWarning</i>	195
<i>PendingDeprecationWarning</i>	195
<i>RuntimeWarning</i>	196
<i>SyntaxWarning</i>	196
En un mot...	196

CHAPITRE 7

Conventions de codage.....	197
Mise en page du code	198
Indentation	198
Taille maximum d'une ligne	198
Commentaires	199
<i>Commentaires simples</i>	200
<i>Commentaires en fin de ligne</i>	200
<i>Blocs de commentaires</i>	200
<i>Documentation strings ou docstrings</i>	201
Espaceur du code	202
Espaces dans les expressions et définitions	203
Conventions de nommage	204
Modules	205
Classes	206
Fonctions et variables globales d'un module, méthodes et attributs d'une classe	206

Constantes	207
Structure d'un module	207
En-tête	207
<i>Interpréteur</i>	207
<i>Encodage</i>	207
<i>Copyright et licence</i>	208
<i>Tags</i>	208
Docstring de module	208
Variables globales spécifiques	209
Clauses d'importations	209
<i>Les jokers</i>	209
<i>Organisation des clauses</i>	210
Variables globales	211
Fonctions et classes, le corps du module	211
<i>Structuration d'une classe</i>	211
Conseils pour le choix des noms	212
Règles générales	212
<i>Du sens</i>	212
<i>Choix de la langue</i>	213
<i>Unicité des noms</i>	213
<i>La bonne longueur</i>	213
<i>Éviter le mélange domaine/technique</i>	213
Règles pour chaque type	213
<i>Modules</i>	213
<i>Classes</i>	214
<i>Méthodes et fonctions</i>	214
<i>Variables</i>	215
En un mot...	215

TROISIÈME PARTIE

La bibliothèque standard217

CHAPITRE 8

Principaux modules 219

Interaction avec l'interpréteur	220
sys	220
<i>argv</i>	220
<i>executable</i>	220
<i>exc_info</i>	220
<i>exit</i>	220

<i>modules</i>	221
<i>last_type, last_value, last_traceback</i>	221
<i>path</i>	221
<i>platform</i>	221
<i>stdin, stdout et stderr</i>	222
Accès au système	222
os	223
<i>Opérations sur les descripteurs de fichiers</i>	223
<i>Manipulation des fichiers et répertoires</i>	225
<i>Manipulation des processus</i>	233
<i>Informations sur le système</i>	239
subprocess	241
<i>call</i>	241
<i>class Popen</i>	241
os.path	243
platform	245
Utilitaires fichiers	247
shutil	247
<i>copy</i>	247
<i>copy2</i>	247
<i>copytree</i>	247
<i>rmtree</i>	248
<i>move</i>	248
dircache	249
filecmp	249
<i>cmp</i>	249
<i>class dircmp</i>	249
Outils de compression	251
gzip	251
<i>class GzipFile</i>	251
<i>open</i>	252
zipfile	254
<i>class ZipFile</i>	254
<i>class ZipInfo</i>	256
<i>is_zipfile</i>	256
Programmation réseau	256
urllib2	257
ftplib	260
En un mot...	263

CHAPITRE 9

Principaux modules, partie 2	265
Persistence	265
cPickle	266
<i>dump</i>	266
<i>load</i>	266
<i>dumps</i>	267
<i>loads</i>	267
shelve	268
<i>open</i>	269
Conversion, transformation de données	270
base64	270
<i>b64encode</i>	270
<i>b64decode</i>	270
haslib	271
<i>haslib.md5</i>	271
<i>class md5</i>	272
<i>hashlib.sha</i>	272
Calculs numériques	273
math	273
<i>fonctions de conversion</i>	273
<i>fonctions trigonométriques</i>	274
<i>constantes</i>	275
Structures de données	276
array	276
<i>array</i>	276
abc	278
collections	281
<i>Le type deque</i>	281
<i>Le type defaultdict</i>	282
<i>La fonction namedtuple</i>	284
<i>Les Abstract Base Classes</i>	284
decimal	285
<i>class Decimal</i>	285
cStringIO	286
<i>class StringIO</i>	286
Utilitaires divers	286
atexit	287
pdb	288
<i>Le mode pas-à-pas</i>	288

<i>Alias et fichier .pdbrc</i>	292
<i>Le mode post mortem</i>	293
getpass	294
copy	295
difflib	296
<i>Affichage des différences</i>	296
<i>Restauration</i>	298
time	299
<i>Epoch</i>	299
<i>UTC/GMT</i>	299
<i>Fonctions de manipulation</i>	300
<i>Formatage des dates</i>	301
datetime	303
<i>class timedelta</i>	303
<i>class date</i>	304
<i>class time</i>	306
<i>class datetime</i>	308
<i>random</i>	308
En un mot...	309

CHAPITRE 10

Principaux modules, partie 3..... 311

Le module itertools	311
chain	311
count	312
cycle	312
dropwhile	313
groupby	313
ifilter	314
ifilterfalse	314
imap	315
islice	315
izip	315
izip_longest	316
repeat	316
starmap	316
takewhile	317
tee	317
Le module re	317
Expressions régulières ?	317
Notation pour les expressions régulières	319

Syntaxe des expressions régulières	319
<i>Symboles simples</i>	319
<i>Symboles de répétition</i>	321
<i>Symboles de regroupement</i>	323
<i>Exemples plus complets</i>	324
<i>Fonctions et objets de re</i>	325
Le module Tkinter	328
Programmation événementielle	328
La classe Tk	328
Les widgets de base de Tkinter	329
<i>Positionnement d'un widget</i>	330
<i>Options et méthodes d'un widget</i>	331
Binding d'événements	339
Application type avec Tkinter	341
Extensions pour Tkinter	343
Le module lib2to3 et le script 2to3	343
En un mot...	345

CHAPITRE 11

Exercices corrigés 347

Mode d'emploi du chapitre	347
Programme	348
Exercice 1 : programme paramétrable	348
<i>Description</i>	348
<i>Points abordés</i>	348
<i>Solution</i>	348
<i>Discussion</i>	349
<i>Extension</i>	351
Texte	351
Exercice 2 : le chiffrement de César	351
<i>Description</i>	351
<i>Points abordés</i>	352
<i>Solution</i>	352
<i>Discussion</i>	352
<i>Extension</i>	353
Exercice 3 : transformer les adresses e-mails et les URL d'un texte en liens	353
<i>Description</i>	353
<i>Points abordés</i>	354
<i>Solution</i>	354
<i>Discussion</i>	354
<i>Extension</i>	355

Exercice 4 : trier des phrases suivant le nombre de mots	356
<i>Description</i>	356
<i>Points abordés</i>	356
<i>Solution</i>	356
<i>Discussion</i>	357
<i>Extension</i>	357
Fichiers	358
Exercice 5 : recherche et remplacement de texte	358
<i>Description</i>	358
<i>Points abordés</i>	358
<i>Solution</i>	358
<i>Discussion</i>	359
<i>Extension</i>	360
Exercice 6 : recopie conditionnelle et récursive de fichiers	360
<i>Description</i>	360
<i>Points abordés</i>	360
<i>Solution</i>	360
<i>Discussion</i>	361
Exercice 7 : ajout d'un fichier dans une archive zip	361
<i>Description</i>	361
<i>Points abordés</i>	361
<i>Solution</i>	361
<i>Discussion</i>	362
<i>Extension</i>	363
Threads et processus	364
Exercice 8 : Tkinter, recherche d'un texte dans des fichiers en tâche de fond	364
<i>Description</i>	364
<i>Points abordés</i>	364
<i>Solution</i>	364
<i>Discussion</i>	368
<i>Extension</i>	368
Exercice 9 : Un web spider rapide	369
<i>Description</i>	369
<i>Points abordés</i>	369
<i>Solution</i>	369
<i>Discussion</i>	371
<i>Extension</i>	371
Persistance	372
Exercice 10 : rendre persistants tous les objets d'un programme	372
<i>Description</i>	372
<i>Points abordés</i>	372

<i>Solution</i>	372
<i>Discussion</i>	373
<i>Extension</i>	374
Web et réseau	374
Exercice 11 : vérificateur de liens	374
<i>Description</i>	374
<i>Points abordés</i>	375
<i>Solution</i>	375
<i>Discussion</i>	375
<i>Extension</i>	375
Exercice 12 : aspirateur de page web	375
<i>Description</i>	375
<i>Points abordés</i>	376
<i>Solution</i>	376
<i>Discussion</i>	378
<i>Extension</i>	379
Exercice 13 : récupération d'un résumé des nouveaux e-mails reçus	379
<i>Description</i>	379
<i>Points abordés</i>	379
<i>Solution</i>	379
<i>Discussion</i>	381
<i>Extension</i>	382
Divers	382
Exercice 14 : système de documentation en ligne des modules	382
<i>Description</i>	382
<i>Points abordés</i>	382
<i>Solution</i>	382
<i>Discussion</i>	383
<i>Extension</i>	384
En un mot...	384

QUATRIÈME PARTIE

Techniques avancées 385

CHAPITRE 12

Programmation dirigée par les tests 387

À quoi servent les tests ?	388
Barrière culturelle	388
Principes	389
Tests unitaires	389

<i>Construction d'un test unitaire</i>	390
<i>Évolution des use cases</i>	391
<i>Non-régression</i>	392
<i>Regroupement des tests</i>	394
<i>Tests plus complexes : raconter une histoire</i>	394
<i>Les bouchons</i>	395
<i>Test coverage</i>	400
<i>Qualité des tests</i>	400
Tests fonctionnels	401
<i>Tests de l'interface</i>	401
<i>Tests de l'ergonomie</i>	402
<i>Dépendance forte à l'outil utilisé et au type d'interface</i>	402
Outils	403
unittest	403
<i>Définition des test cases</i>	404
<i>Organisation d'une campagne de tests</i>	407
doctests	411
<i>Exécution des doctests</i>	411
<i>Syntaxe des doctests</i>	412
<i>Environnement et options d'exécution</i>	415
<i>doctests dans un fichier texte séparé</i>	420
<i>Script de test</i>	423
Coverage	424
Intégration dans l'environnement d'un projet	427
Le futur de PyUnit	428
En un mot...	430

CHAPITRE 13

Bonnes pratiques et optimisation du code 431

Quand optimiser ?	432
Profiling	432
Méthodes de profiling	433
Outils de profiling	433
<i>Le module profile</i>	433
<i>Le module hotshot</i>	434
<i>Le module cProfile</i>	435
<i>Le module pstats</i>	436
<i>hotshot et pstats</i>	437
<i>timeit</i>	438
Amélioration des performances	440
Code patterns	441

<i>Quel type de conteneur choisir ?</i>	441
<i>Trier des valeurs</i>	441
<i>Concaténer des chaînes</i>	444
<i>Remplacer certains tests par une gestion d'exception</i>	445
<i>Minimiser les appels et rapprocher le code</i>	446
<i>Utiliser les list comprehensions</i>	448
<i>Utiliser les generators et les genexp</i>	449
<i>Préférer les fonctions d'itertools</i>	449
Caching	450
Multithreading	452
<i>Ressources partagées : difficultés de programmation</i>	452
<i>Le module threading</i>	454
<i>Le module Queue</i>	462
<i>Le Global Interpreter Lock et multiprocessing</i>	463
Le côté obscur de la force : extension du langage	464
<i>Environnement de compilation</i>	464
<i>Binding de bibliothèque</i>	465
<i>Création d'un module d'extension</i>	469
Optimisation de l'utilisation de mémoire vive	475
<i>Économie de mémoire</i>	476
Optimisation du bytecode	477
Psyco et Cython	477
<i>Psyco</i>	477
<i>Cython</i>	479
Les tests de performance continus	480
Rapport sur les performances	481
Tests de performance ciblés	481
decorator timed	482
En un mot...	485

CHAPITRE 14

Programmation orientée objet 487

Principes généraux	487
Typage, classification et encapsulation	488
<i>Typage de Liskov</i>	488
<i>Encapsulation</i>	490
Héritage et polymorphisme	492
<i>Héritage</i>	492
<i>Polymorphisme</i>	493
<i>Duck typing et interfaces</i>	495
Relations entre objets	496

<i>Relation simple</i>	497
<i>Relation multiple</i>	497
Héritage multiple	498
Métaclasses	499
Garbage collecting	499
Design patterns orientés objet	500
Patterns de génération d'objets	500
<i>Singleton et Borg</i>	501
<i>Factory</i>	504
Patterns fonctionnels	505
<i>Visitor</i>	505
<i>Observer</i>	507
<i>Memento</i>	510
<i>Chain of responsibility</i>	513
<i>State</i>	516
Patterns structurels	518
<i>Adapter</i>	518
<i>Facade</i>	520
<i>Proxy</i>	521
En un mot...	522

ANNEXE A

L'histoire de Python	523
Le langage ABC	523
Environnement de développement	523
Types de données	524
Indentation du code	524
Le projet Amoeba	525
Le CNRI	526
PythonLabs et BeOpen.com	526
Python Software Foundation et Digital Creations	527
Python et Zope	527

ANNEXE B

Bibliothèques tierces	531
Installer une bibliothèque externe	532
Utilisation de <i>setuptools</i>	533
Bases de données	535
<i>Gadfly</i>	535
<i>pysqlite</i>	535

mysql-python	535
psycopg	536
ODBC	536
python-ldap	536
SQLAlchemy	536
Traitement de texte	536
lxml	537
Beautiful Soup	537
Packaging, distribution	537
Tests fonctionnels et contrôle qualité	538
Twill	538
Funkload	538
guitest	538
PyLint	539
Pyflakes	539
MS-Windows	539
Win32 Extensions	539
win32com	540
Interfaces graphiques	540
wxPython	540
PyQT	540
PyGTK	540
Reporting et conversion	541
ReportLab	541
RML2PDF	541
reStructuredText	541
rest2web	542
Jeux et 3D	542
Pygame	542
Soya 3D	542
vpython	542
PyOpenGL	543
Audio et Vidéo	543
PyMedia	543
PyAlsa	543
Bibliothèques scientifiques	543
Numerical Python	544
SciPy	544
Biopython	544
Web	544

ANNEXE C

Sites, flux RSS, blogs et autres friandises...	545
Flux RSS	545
Blogs	546
Sites	547
Index	549