

# Table des matières

<b>Prologue</b> .....	5
<b>1 Ouvrir IDLE Python et éditer un script</b> .....	23
<b>2 Mots réservés et fonctions internes</b> .....	29
2.1 Mots réservés .....	29
2.2 Fonctions internes .....	29
2.3 Module <code>builtins</code> .....	31
2.4 Méthodes internes .....	32
2.5 Méthodes spéciales .....	34
2.6 Type des fonctions et méthodes internes .....	37
<b>3 Données et types de données</b> .....	39
3.1 Types importants .....	39
3.1.1 Données de type <code>bool</code> (booléens) .....	39
3.1.2 Données de type <code>int</code> (entiers) .....	39
3.1.3 Données de type <code>float</code> (nombres décimaux) .....	40
3.1.4 Données de type <code>complex</code> (nombres complexes) .....	40
3.1.5 Données de type <code>str</code> (chaînes de caractères) .....	40
3.1.6 Données de type <code>bytes</code> (chaînes de bytes) .....	40
3.1.7 Données de type <code>bytearray</code> (chaînes de bytes) .....	41
3.1.8 Données de type <code>list</code> (listes) .....	41
3.1.9 Données de type <code>tuple</code> (uplets) .....	41
3.1.10 Données de type <code>range</code> (plages d'entiers) .....	41
3.1.11 Données de type <code>dict</code> (dictionnaires) .....	41
3.1.12 Données de type <code>set</code> (ensembles) .....	42
3.1.13 Données de type <code>frozenset</code> (ensembles) .....	42
3.1.14 Données de type <code>generator</code> (listes virtuelles) .....	42
3.1.15 Données de type <code>function</code> (fonctions) .....	42
3.1.16 Données de type <code>builtin_function_or_method</code> .....	43
3.1.17 Données de type <code>method-wrapper</code> .....	43
3.1.18 Données de type <code>NoneType</code> (type de <code>None</code> ) .....	43
3.1.19 Données de type <code>Objet-fichier-texte</code> .....	44

3.1.20	Données de type <code>Objet-fichier-binaire</code> . . . . .	44
3.1.21	Données de type <code>module</code> (bibliothèques) . . . . .	45
3.1.22	Donnée de type <code>type</code> . . . . .	45
3.1.23	Autres types . . . . .	45
3.2	Tester le type d'une donnée . . . . .	45
3.3	Itérables, séquences, mutables, etc. . . . .	46
3.4	Typage dynamique . . . . .	48
<b>4</b>	<b>Affectations, identificateurs et aliasing</b> . . . . .	<b>51</b>
4.1	Différentes manières d'affecter une donnée à une variable . . . . .	51
4.1.1	Manière classique . . . . .	51
4.1.2	Affectation avec décompression . . . . .	52
4.1.3	Affectations à la queue leu leu . . . . .	54
4.2	Affectation et opération simultanées . . . . .	54
4.3	Notion d'identificateur . . . . .	55
4.4	Identificateurs autorisés par Python . . . . .	59
4.5	Aliasing . . . . .	60
4.6	Fonction <code>id</code> . . . . .	62
4.7	Résolution d'un identificateur . . . . .	64
4.7.1	Variables locales, globales et internes . . . . .	64
4.7.2	Règle Local-Global-Interne . . . . .	65
4.7.3	Fonctions <code>globals</code> et <code>locals</code> . . . . .	66
4.7.4	Mot réservé <code>global</code> . . . . .	67
<b>5</b>	<b>Modules et bibliothèques</b> . . . . .	<b>69</b>
5.1	Importer un module . . . . .	69
5.2	Importer le contenu d'un module . . . . .	70
5.3	Ses propres modules à soi . . . . .	71
5.4	Auto-test . . . . .	72
5.5	<code>sys.modules</code> et <code>sys.builtin_modules_names</code> . . . . .	74
5.6	La liste <code>sys.path</code> . . . . .	77
5.7	D'autres bibliothèques . . . . .	78
<b>6</b>	<b>Afficher : <code>print</code>, <code>str</code> et <code>repr</code></b> . . . . .	<b>81</b>
6.1	Introduction . . . . .	81
6.2	Argument <code>end</code> . . . . .	81
6.3	Afficher plusieurs données et argument <code>sep</code> . . . . .	82
6.4	Argument <code>file</code> . . . . .	82
6.5	Fonction <code>str</code> versus fonction <code>repr</code> . . . . .	83
<b>7</b>	<b>Calcul et arithmétique</b> . . . . .	<b>85</b>

7.1	Types de nombres et méthodes	85
7.1.1	Type <code>complex</code>	85
7.1.2	Type <code>float</code>	86
7.1.3	Type <code>int</code>	87
7.2	Opérations élémentaires	89
7.3	Type <code>int</code> versus type <code>float</code>	90
7.4	Dysfonctionnement avec les <code>float</code>	92
7.5	Module <code>decimal</code>	94
7.6	Module <code>fractions</code>	96
7.7	Fonctions internes pour le calcul	97
7.7.1	Fonction valeur absolue : <code>abs</code>	97
7.7.2	Fonction troncature : <code>int</code>	97
7.7.3	Fonction arrondir : <code>round</code>	98
7.7.4	Fonction division euclidienne : <code>divmod</code>	98
7.7.5	Fonction <code>float</code>	99
7.7.6	Fonction puissance : <code>pow</code>	99
7.7.7	Fonction de sommation : <code>sum</code>	99
7.7.8	Fonction d'évaluation : <code>eval</code>	100
7.7.9	Fonction <code>complex</code>	100
7.7.10	Fonctions <code>min</code> et <code>max</code>	100
7.7.11	Fonction <code>range</code>	101
7.8	Calcul en base 2, 8 ou 16	101
7.8.1	Calcul en binaire	101
7.8.1.1	Conjonction : <code>&amp;</code>	101
7.8.1.2	Disjonction : <code> </code>	102
7.8.1.3	Disjonction exclusive : <code>^</code>	102
7.8.1.4	Négation : <code>~</code>	103
7.8.1.5	Décalage à gauche : <code>&lt;&lt;</code>	103
7.8.1.6	Décalage à droite : <code>&gt;&gt;</code>	103
7.8.2	Calcul en octal	104
7.8.3	Calcul en hexadécimal	104
7.9	Objet <code>NaN</code> et nombres infinis	105
7.10	Modules <code>math</code> et <code>cmath</code>	107
7.10.1	Exploration	107
7.10.2	Description rapide de quelques fonctions	108
7.10.2.1	Fonction <code>atan2</code>	108
7.10.2.2	Fonction <code>expm1</code>	108
7.10.2.3	Fonction hypoténuse : <code>hypot</code>	108
7.10.2.4	Fonction logarithme en base quelconque	108
7.10.2.5	Fonction <code>log1p</code>	109
7.10.2.6	Fonction <code>lgamma</code>	109

7.10.2.7	Fonction <code>erf</code> et <code>erfc</code> . . . . .	109
7.10.2.8	Fonction <code>frexp</code> . . . . .	109
7.10.2.9	Fonction <code>ldexp</code> . . . . .	109
7.10.2.10	Fonctions <code>isinf</code> et <code>isnan</code> . . . . .	109
7.10.2.11	Fonction de comparaison avec tolérance . . .	110
7.10.3	Fonctions internes améliorées par <code>math</code> . . . . .	110
7.10.3.1	Fonction partie entière : <code>floor</code> . . . . .	110
7.10.3.2	Fonction d'arrondi à l'unité par excès : <code>ceil</code> .	110
7.10.3.3	Fonction troncature : <code>trunc</code> . . . . .	110
7.10.3.4	Fonction de décomposition : <code>modf</code> . . . . .	110
7.10.3.5	Fonction valeur absolue : <code>fabs</code> . . . . .	111
7.10.3.6	Fonction modulo (reste euclidien) : <code>fmod</code> . . .	111
7.10.3.7	Fonction de sommation : <code>fsum</code> . . . . .	111
7.10.4	Module <code>cmath</code> . . . . .	112
7.10.4.1	Fonction <code>phase</code> . . . . .	112
7.10.4.2	Fonction <code>polar</code> . . . . .	112
7.10.4.3	Fonction <code>rect</code> . . . . .	112
7.11	Module <code>random</code> . . . . .	113
7.11.1	Exploration . . . . .	113
7.11.2	Description de quelques fonctions . . . . .	113
7.11.2.1	Fonction <code>choice</code> . . . . .	113
7.11.2.2	Fonction <code>sample</code> . . . . .	114
7.11.2.3	Fonction <code>shuffle</code> . . . . .	114
7.11.2.4	Fonction <code>randint</code> . . . . .	114
7.11.2.5	Fonction <code>randrange</code> . . . . .	114
7.11.2.6	Fonction <code>seed</code> . . . . .	115
7.11.2.7	Fonction <code>random</code> . . . . .	115
7.11.2.8	Fonction <code>uniform</code> . . . . .	115
7.11.2.9	Fonction <code>triangular</code> . . . . .	115
7.11.2.10	Fonction <code>expovariate</code> . . . . .	115
7.11.2.11	Fonction <code>gauss</code> . . . . .	116
7.11.2.12	Fonction <code>normalvariate</code> . . . . .	116
<b>8</b>	<b>Saisie au clavier : <code>input</code></b> . . . . .	<b>117</b>
<b>9</b>	<b>Structure <code>if-elif-else</code></b> . . . . .	<b>119</b>
9.1	Structure <code>if</code> simple . . . . .	119
9.2	Structure <code>if-else</code> . . . . .	119
9.3	Instruction <code>elif</code> (else if) . . . . .	120
9.4	Syntaxe <code>if</code> compacte . . . . .	121
<b>10</b>	<b>Boucles avec <code>for</code></b> . . . . .	<b>123</b>

10.1	Introduction	123
10.2	Sauter avec <code>continue</code>	124
10.3	Casser une boucle avec <code>break</code>	124
10.4	Itérer avec <code>enumerate</code> et <code>zip</code>	125
<b>11</b>	<b>Boucles avec <code>while</code></b>	<b>127</b>
11.1	Introduction	127
11.2	Sauter avec <code>continue</code>	128
11.3	Sortir avec <code>break</code>	128
<b>12</b>	<b>Gestion des erreurs avec <code>try-except-else</code></b>	<b>131</b>
12.1	Structure <code>try-except</code>	131
12.2	Structure <code>try-except-else</code>	133
12.3	Structure <code>try-except-else-finally</code>	134
12.4	La classe <code>BaseException</code> et ses sous-classes	136
12.5	Traiter une erreur en particulier	142
12.6	Lever une exception avec <code>raise</code>	143
12.7	Tester une assertion avec <code>assert</code>	144
12.8	Instruction <code>with</code>	145
12.9	Désamorcer toutes les erreurs	145
12.10	Exemple amusant	146
<b>13</b>	<b>Fonctions : <code>type function</code></b>	<b>147</b>
13.1	Définir et appeler une fonction	147
13.2	Fonction multivaleurs	148
13.3	Type <code>function</code>	148
13.4	Fonction versus procédure	150
13.5	L'ordre des arguments	151
13.6	N'importe quel type en argument	152
13.7	Avantage du typage dynamique	152
13.8	Documentation ( <code>docstring</code> )	153
13.9	Doctests et module <code>doctest</code>	154
13.10	Variables locales et variables globales	155
13.11	Une fonction dans une fonction	157
13.12	Valeur d'un argument par défaut	159
13.13	Effet de bord	161
13.14	Décompresser un itérable avec <code>*</code>	162
13.15	Décompresser un <code>dict</code> avec <code>**</code>	165
13.16	Arguments avec nom obligatoire	167
13.17	Règle à suivre pour les arguments (signature)	168

13.18 Fonctions lambda	169
<b>14 Décorateurs de fonctions</b>	<b>171</b>
14.1 Introduction	171
14.2 Opérateur @	172
14.3 Exemple	173
14.4 Un conseil pour les arguments de la décorée	174
14.5 Préserver le docstring	175
14.6 Décorateurs avec paramètres	177
14.7 Exemples classiques	178
14.7.1 Affichage du nom et des arguments : logging	178
14.7.2 Cache d'une fonction : mémoïsation	179
14.7.3 Chronomètre attaché à une fonction : benchmark	181
14.7.4 Compteur attaché à une fonction	181
<b>15 Gestionnaires de contexte : with</b>	<b>183</b>
15.1 Introduction	183
15.2 Faire passer un argument au gestionnaire	186
15.3 Mots réservés with et as	187
15.4 Question de style	188
15.5 Context Decorator	189
15.6 Une alternative avec contextlib	191
15.7 try versus with	192
<b>16 Plages d'entiers : type range</b>	<b>195</b>
16.1 Introduction	195
16.2 Convertir un range	196
16.3 Le type range est itérable	197
16.4 Indexing et slicing	197
16.5 Fonction slice	199
16.6 Initialiser des variables avec un range	199
16.7 Exploration	200
<b>17 Booléens : type bool</b>	<b>201</b>
17.1 Type bool	201
17.2 Opérateurs de comparaison	201
17.3 Convertir une donnée en booléen	203
17.4 Calcul numérique avec un booléen	204
17.5 Opérations sur les booléens	205
17.5.1 Négation	205

17.5.2	Conjonction	205
17.5.3	Disjonction	205
17.5.4	Disjonction exclusive	206
17.5.5	Règles de priorité	206
17.6	Short-cut	207
17.7	Exploration	209
<b>18</b>	<b>Chaînes de caractères : type <code>str</code></b>	<b>211</b>
18.1	Délimiteurs	211
18.2	Code d'échappement	212
18.3	Fonction <code>input</code>	215
18.4	Représentation d'une chaîne : <code>repr</code>	216
18.5	Triple guillemets ou triple quotes	217
18.6	Chaîne brute ( <i>raw string</i> )	218
18.7	Indices et slicing	219
18.8	Parcourir une chaîne	220
18.9	Opérateurs <code>+</code> et <code>*</code>	221
18.10	Interpréter le contenu d'une chaîne	221
18.10.1	Fonction <code>int</code>	222
18.10.2	Fonction <code>float</code>	222
18.10.3	Fonction <code>complex</code>	222
18.10.4	Fonction <code>eval</code>	222
18.11	Conversion avec <code>list</code> , <code>tuple</code> , <code>set</code> et <code>frozenset</code>	223
18.12	Méthode <code>format</code>	224
18.13	Standard <code>Unicode</code>	224
18.14	Comparer des chaînes	225
18.15	Exploration de <code>str</code>	225
18.16	Description des méthodes de <code>str</code>	226
18.16.1	Méthode <code>count</code>	226
18.16.2	Méthode <code>find</code>	226
18.16.3	Méthode <code>rfind</code>	227
18.16.4	Méthode <code>index</code>	227
18.16.5	Méthode <code>rindex</code>	227
18.16.6	Méthode <code>replace</code>	227
18.16.7	Méthode <code>split</code>	228
18.16.8	Méthode <code>rsplit</code>	228
18.16.9	Méthode <code>splitlines</code>	228
18.16.10	Méthode <code>partition</code>	229
18.16.11	Méthode <code>rpartition</code>	229
18.16.12	Méthode <code>join</code>	229
18.16.13	Méthode <code>upper</code>	229

18.16.14	Méthode <code>isupper</code> . . . . .	230
18.16.15	Méthode <code>lower</code> . . . . .	230
18.16.16	Méthode <code>islower</code> . . . . .	230
18.16.17	Méthode <code>casefold</code> . . . . .	230
18.16.18	Méthode <code>title</code> . . . . .	230
18.16.19	Méthode <code>istitle</code> . . . . .	231
18.16.20	Méthode <code>capitalize</code> . . . . .	231
18.16.21	Méthode <code>swapcase</code> . . . . .	231
18.16.22	Méthode <code>isalnum</code> . . . . .	231
18.16.23	Méthode <code>isalpha</code> . . . . .	231
18.16.24	Méthode <code>isdigit</code> . . . . .	231
18.16.25	Méthode <code>isdecimal</code> . . . . .	232
18.16.26	Méthode <code>isnumeric</code> . . . . .	232
18.16.27	Méthode <code>isspace</code> . . . . .	232
18.16.28	Méthode <code>startswith</code> . . . . .	232
18.16.29	Méthode <code>endswith</code> . . . . .	232
18.16.30	Méthode <code>strip</code> . . . . .	232
18.16.31	Méthode <code>rstrip</code> . . . . .	233
18.16.32	Méthode <code>rstrip</code> . . . . .	233
18.16.33	Méthode <code>expandtabs</code> . . . . .	233
18.16.34	Méthode <code>center</code> . . . . .	233
18.16.35	Méthode <code>ljust</code> . . . . .	234
18.16.36	Méthode <code>rjust</code> . . . . .	234
18.16.37	Méthode <code>zfill</code> . . . . .	234
18.16.38	Méthode <code>format</code> . . . . .	234
18.16.39	Méthode <code>encode</code> . . . . .	235
18.16.40	Méthode <code>isidentifier</code> . . . . .	235
18.16.41	Méthode <code>isprintable</code> . . . . .	235
18.16.42	Méthode <code>format_map</code> . . . . .	235
18.16.43	Méthode <code>maketrans</code> . . . . .	236
18.16.44	Méthode <code>translate</code> . . . . .	236
<b>19</b>	<b>Formater les chaînes avec <code>format</code></b> . . . . .	<b>239</b>
19.1	Introduction . . . . .	239
19.2	Syntaxe d'une balise . . . . .	239
19.3	Coder une référence . . . . .	240
19.3.1	Code nom . . . . .	240
19.3.2	Code attribut, clé ou indice . . . . .	241
19.4	Coder une conversion . . . . .	243
19.5	Coder un formatage . . . . .	243
19.5.1	Codes <code>fill</code> , <code>align</code> et <code>width</code> . . . . .	243

19.5.2	Code sign	244
19.5.3	Code précision	244
19.5.4	Code type	245
19.5.5	Code dièse #	248
19.5.6	Code séparateur anglosaxon <code>,</code>	248
19.5.7	Code 0	248
19.6	Cas particulier des dates	249
19.7	Application : affichage d'un tableau	249
19.8	Échappement	249
19.9	Plus compliqué	250
19.10	Fonction <code>format</code>	250
19.11	Ancien formatage	251
<b>20</b>	<b>Listes : type <code>list</code></b>	<b>253</b>
20.1	Introduction	253
20.2	Le type <code>list</code> est itérable	253
20.3	Concaténation et multiplication	254
20.4	Convertir avec la fonction <code>list</code>	255
20.5	Indexing et slicing	255
20.6	Remplacer une tranche par une autre	257
20.7	Aliasing	259
20.8	Simuler une matrice	260
20.8.1	Adapter les indices	260
20.8.2	Initialisation	260
20.9	Compréhension	262
20.10	Fusionner et aplatir	263
20.11	Trier avec <code>sort</code> ou <code>sorted</code> (arguments <code>reverse</code> , <code>key</code> )	264
20.12	Exploration	266
20.13	Fonctions, instructions et méthodes	266
20.13.1	Instruction <code>del</code>	267
20.13.2	Méthode <code>sort</code>	267
20.13.3	Fonction <code>sorted</code>	268
20.13.4	Méthode <code>reverse</code>	268
20.13.5	Fonction <code>reversed</code>	268
20.13.6	Méthode <code>index</code>	268
20.13.7	Méthode <code>count</code>	269
20.13.8	Méthode <code>remove</code>	269
20.13.9	Méthode <code>append</code>	269
20.13.10	Méthode <code>insert</code>	270
20.13.11	Méthode <code>extend</code>	270
20.13.12	Méthode <code>pop</code>	270

20.13.13	Fonction <code>max</code>	271
20.13.14	Fonction <code>min</code>	271
20.13.15	Méthode <code>copy</code>	271
20.13.16	Méthode <code>clear</code>	271
<b>21</b>	<b>Uplets : type <code>tuple</code></b>	<b>273</b>
21.1	Introduction	273
21.2	Le type <code>tuple</code> est itérable	274
21.3	Fonction <code>tuple</code>	275
21.4	Slicing	276
21.5	Pas de mutation	277
21.6	Concaténation et multiplication	277
21.7	Exploration	278
21.8	Intérêt du <code>tuple</code>	278
21.9	Pas de compréhension	279
21.10	Uplets à champs nommés : type <code>namedtuple</code>	279
<b>22</b>	<b>Générateurs : type <code>generator</code></b>	<b>281</b>
22.1	Introduction	281
22.2	Fonction <code>next</code>	282
22.3	Caractéristiques	282
22.4	Fonction <code>reversed</code>	283
22.5	Fonction <code>iter</code>	284
22.6	Fonction <code>iter</code> avec sentinelle	285
22.7	Instruction <code>yield</code>	286
22.8	Fonction <code>enumerate</code>	288
22.9	Fonction <code>zip</code>	289
22.10	Fonctions <code>map</code> , <code>filter</code> et <code>reduce</code>	290
22.10.1	Fonction <code>filter</code>	291
22.10.2	Fonction <code>map</code>	291
22.10.3	Ancienne fonction <code>reduce</code>	292
22.11	Module <code>itertools</code>	293
22.11.1	Fonction <code>itertools.chain</code>	293
22.11.2	Fonction <code>itertools.islice</code>	294
22.12	Terminologie officielle	294
<b>23</b>	<b>Ensembles : types <code>set</code> et <code>frozenset</code></b>	<b>297</b>
23.1	Introduction	297
23.2	Les types <code>set</code> et <code>frozenset</code> sont itérables	298
23.3	Fonctions <code>set</code> et <code>frozenset</code>	298

23.4	Type <code>set</code> et aliasing	299
23.5	Compréhension	300
23.6	Opérations sur les ensembles	300
23.7	Méthodes spécifiques à <code>set</code>	302
23.7.1	Méthode <code>add</code>	302
23.7.2	Méthode <code>remove</code>	302
23.7.3	Méthode <code>discard</code>	302
23.7.4	Méthode <code>update</code>	303
23.7.5	Méthode <code>intersection_update</code>	303
23.7.6	Méthode <code>difference_update</code>	303
23.7.7	Méthode <code>symmetric_difference_update</code>	304
23.7.8	Méthode <code>pop</code>	304
23.7.9	Méthode <code>clear</code>	304
23.8	Explorons le type <code>set</code>	305
23.9	Explorons le type <code>frozenset</code>	305
<b>24</b>	<b>Dictionnaires : type <code>dict</code></b>	<b>307</b>
24.1	Introduction	307
24.2	Notion de dictionnaire	307
24.3	Clés autorisées (données hashables)	308
24.4	Définir un dictionnaire d'un seul coup	309
24.5	Convertir avec <code>dict</code>	310
24.6	Compréhension	310
24.7	Modifier un item	311
24.8	Méthodes <code>keys</code> , <code>values</code> et <code>items</code>	311
24.9	Itérer un dictionnaire	312
24.10	Phénomène d'aliasing	313
24.11	Méthode <code>get</code>	314
24.12	Attributs de <code>dict</code>	315
24.13	Instructions, méthodes et fonctions	315
24.13.1	Méthode <code>clear</code>	315
24.13.2	Instruction <code>del</code>	316
24.13.3	Méthode <code>fromkeys</code>	316
24.13.4	Fonction <code>len</code>	316
24.13.5	Méthode <code>pop</code>	317
24.13.6	Méthode <code>popitem</code>	317
24.13.7	Méthode <code>setdefault</code>	317
24.13.8	Méthode <code>update</code>	318
24.14	Programmer avec un dictionnaire	318
<b>25</b>	<b>Chaînes d'octets : types <code>bytes</code> et <code>bytearray</code></b>	<b>321</b>

25.1	Introduction	321
25.2	Code ASCII standard	323
25.3	Représentation des octets de 32 à 126	325
25.4	Écrire des bytes avec des caractères ASCII standard	326
25.5	Indexation, slicing et itération	327
25.6	Convertir une donnée en bytes	328
25.7	Type idéal pour ouvrir un fichier quelconque	329
25.8	Type <code>bytearray</code>	331
25.9	Exploration	332
25.10	Quelques méthodes	334
25.10.1	Méthode <code>decode</code>	334
25.10.2	Méthode <code>fromhex</code>	335
<b>26</b>	<b>Modules <code>os</code> et <code>os.path</code></b>	<b>337</b>
26.1	Vocabulaire	337
26.2	Avertissement	337
26.3	Répertoire courant et utilisateur	338
26.4	Lister le contenu d'un répertoire	338
26.5	Distinguer entre fichiers et répertoires	340
26.6	Traitement des noms de fichiers	341
26.7	Variables d'environnement	342
26.8	Bilan	343
<b>27</b>	<b>Lire et écrire des fichiers</b>	<b>345</b>
27.1	Changer le répertoire courant	345
27.2	Types fichier-texte et fichier-binaire	345
27.3	Créer un fichier-texte et écrire à l'intérieur	346
27.4	Écrire dans un fichier-texte déjà existant	347
27.5	Lecture du contenu d'un fichier-texte	348
27.6	Lecture d'un fichier-texte caractère par caractère	348
27.7	Notion de ligne dans un texte	349
27.8	Lecture d'un fichier-texte ligne par ligne	350
27.9	Itérer un objet-fichier	351
27.10	Intérêt de la commande <code>break</code>	351
27.11	Un mot sur l'encodage	352
27.12	Objet fichier-binaire	352
27.13	Bilan sur les différents modes d'accès	353
27.14	Module <code>pickle</code>	353
27.15	Intérêt de <code>try</code> et <code>with</code>	355
27.16	Quelques attributs	356
27.16.1	Attribut <code>mode</code>	356

27.16.2	Attribut <code>name</code> . . . . .	357
27.16.3	Attribut <code>closed</code> . . . . .	357
27.17	Exploration . . . . .	357
<b>28</b>	<b>Encodage et décodage</b> . . . . .	<b>361</b>
28.1	Encodage de texte . . . . .	361
28.2	Normes <code>Latin-1</code> et <code>utf-8</code> . . . . .	362
28.3	Choisir l'encodage pour les fichiers textes . . . . .	364
28.4	Encodage des scripts écrits sous IDLE Python . . . . .	364
28.5	Module <code>encodings</code> . . . . .	365
<b>29</b>	<b>Module <code>turtle</code></b> . . . . .	<b>367</b>
29.1	Introduction . . . . .	367
29.2	Fonctions de base pour dessiner . . . . .	368
29.2.1	Lever et baisser le stylo . . . . .	368
29.2.2	Avancer et reculer . . . . .	369
29.2.3	Pivoter . . . . .	369
29.2.4	Remplir une surface . . . . .	370
29.2.5	Tracer un polygone . . . . .	370
29.3	Angles . . . . .	370
29.3.1	Unité d'angle . . . . .	370
29.3.2	Angle de la tortue . . . . .	371
29.4	Déplacements en coordonnées absolues . . . . .	372
29.5	Couleurs . . . . .	373
29.5.1	Choisir le <code>colormode</code> . . . . .	373
29.5.2	Couleur du stylo . . . . .	373
29.5.3	Couleur de remplissage . . . . .	374
29.5.4	Fond d'écran . . . . .	375
29.6	Effacer, remettre à zéro . . . . .	375
29.7	Cercles et points . . . . .	376
29.8	Aspect . . . . .	377
29.8.1	Épaisseur du stylo . . . . .	377
29.8.2	Écrire du texte . . . . .	378
29.8.3	Titre et taille de la fenêtre Turtle Graphics . . . . .	378
29.8.4	Apparence de la tortue . . . . .	378
29.9	D'autres fonctions . . . . .	379
29.10	Exemples . . . . .	380
<b>30</b>	<b>Introspection chez Python</b> . . . . .	<b>381</b>
30.1	Fonction <code>type</code> . . . . .	381

30.2 Fonctions pour les attributs . . . . .	383
30.2.1 Fonction <code>hasattr</code> . . . . .	383
30.2.2 Fonction <code>deltattr</code> . . . . .	384
30.2.3 Fonction <code>setattr</code> . . . . .	384
30.2.4 Fonction <code>getattr</code> . . . . .	385
30.3 Fonctions pour les classes . . . . .	386
30.3.1 Fonction <code>isinstance</code> . . . . .	386
30.3.2 Fonction <code>issubclass</code> . . . . .	387
30.4 Fonction <code>dir</code> . . . . .	387
30.5 Attribut <code>__doc__</code> et fonction <code>help</code> . . . . .	388
30.6 Fonction <code>callable</code> . . . . .	389
30.7 Module <code>builtins</code> . . . . .	390
30.8 Attribut <code>__name__</code> . . . . .	393
30.9 Un petit code pour résumer . . . . .	394
<b>Bibliographie</b> . . . . .	<b>397</b>
<b>Index</b> . . . . .	<b>399</b>