

VINCENT LE GOFF

# APPRENEZ À PROGRAMMER EN PYTHON

DÉVELOPPER EN PYTHON N' A JAMAIS  
ÉTÉ AUSSI FACILE !



Issu du célèbre  
**Site du Zéro**  
[www.siteduzero.com](http://www.siteduzero.com)



[www.siteduzero.com](http://www.siteduzero.com)



Sauf mention contraire, le contenu de cet ouvrage est publié sous la licence :  
Creative Commons BY-NC-SA 2.0

La copie de cet ouvrage est autorisée sous réserve du respect des conditions de la licence  
Texte complet de la licence disponible sur : <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>

Simple IT 2011 - ISBN : 979-10-90085-03-9

# Sommaire

<b>Avant-propos</b>	<b>i</b>
Qu'allez-vous apprendre en lisant ce livre? . . . . .	ii
Comment lire ce livre? . . . . .	iii
Remerciements . . . . .	iv
<b>I Introduction à Python</b>	<b>1</b>
<b>1 Qu'est-ce que Python ?</b>	<b>3</b>
Un langage de programmation? Qu'est-ce que c'est? . . . . .	4
Pour la petite histoire . . . . .	5
Installer Python . . . . .	7
<b>2 Premiers pas avec l'interpréteur de commandes Python</b>	<b>13</b>
Où est-ce qu'on est, là? . . . . .	14
Vos premières instructions : un peu de calcul mental pour l'ordinateur . . . . .	15
<b>3 Le monde merveilleux des variables</b>	<b>19</b>
C'est quoi, une variable? Et à quoi cela sert-il? . . . . .	20
Les types de données en Python . . . . .	22
Première utilisation des fonctions . . . . .	26
<b>4 Les structures conditionnelles</b>	<b>31</b>
Vos premières conditions et blocs d'instructions . . . . .	32

De nouveaux opérateurs . . . . .	36
Votre premier programme! . . . . .	38
<b>5 Les boucles</b> . . . . .	<b>45</b>
En quoi cela consiste-t-il? . . . . .	46
La boucle <code>while</code> . . . . .	47
La boucle <code>for</code> . . . . .	49
Un petit bonus : les mots-clés <code>break</code> et <code>continue</code> . . . . .	51
<b>6 Pas à pas vers la modularité (1/2)</b> . . . . .	<b>53</b>
Les fonctions : à vous de jouer . . . . .	54
Les fonctions <code>lambda</code> . . . . .	59
À la découverte des modules . . . . .	60
<b>7 Pas à pas vers la modularité (2/2)</b> . . . . .	<b>67</b>
Mettre en boîte notre code . . . . .	68
Je viens pour conquérir le monde. . . et créer mes propres modules . . . . .	71
Les packages . . . . .	74
<b>8 Les exceptions</b> . . . . .	<b>79</b>
À quoi cela sert-il? . . . . .	80
Forme minimale du bloc <code>try</code> . . . . .	81
Forme plus complète . . . . .	82
Les assertions . . . . .	85
Lever une exception . . . . .	86
<b>9 TP : tous au ZCasino</b> . . . . .	<b>89</b>
Notre sujet . . . . .	90
Organisons notre projet . . . . .	90
Correction! . . . . .	92
<b>II La Programmation Orientée Objet côté utilisateur</b> . . . . .	<b>95</b>
<b>10 Notre premier objet : les chaînes de caractères</b> . . . . .	<b>97</b>
Vous avez dit objet? . . . . .	98
Les méthodes de la classe <code>str</code> . . . . .	98

---

Parcours et sélection de chaînes . . . . .	105
<b>11 Les listes et tuples (1/2)</b>	<b>109</b>
Créons et éditons nos premières listes . . . . .	110
Le parcours de listes . . . . .	115
Un petit coup d’œil aux tuples . . . . .	118
<b>12 Les listes et tuples (2/2)</b>	<b>121</b>
Entre chaînes et listes . . . . .	122
Les listes et paramètres de fonctions . . . . .	124
Les compréhensions de liste . . . . .	127
<b>13 Les dictionnaires</b>	<b>133</b>
Création et édition de dictionnaires . . . . .	134
Les méthodes de parcours . . . . .	139
Les dictionnaires et paramètres de fonction . . . . .	141
<b>14 Les fichiers</b>	<b>145</b>
Avant de commencer . . . . .	146
Lecture et écriture dans un fichier . . . . .	148
Enregistrer des objets dans des fichiers . . . . .	152
<b>15 Portée des variables et références</b>	<b>155</b>
La portée des variables . . . . .	156
Les variables globales . . . . .	161
<b>16 TP : un bon vieux pendu</b>	<b>163</b>
Votre mission . . . . .	164
Correction proposée . . . . .	165
<b>III La Programmation Orientée Objet côté développeur</b>	<b>171</b>
<b>17 Première approche des classes</b>	<b>173</b>
Les classes, tout un monde . . . . .	174
Nos premiers attributs . . . . .	175
Les méthodes, la recette . . . . .	180

Un peu d'introspection . . . . .	185
<b>18 Les propriétés</b>	<b>189</b>
Qu'est-ce que l'encapsulation? . . . . .	190
Les propriétés à la casserole . . . . .	191
Les propriétés en action . . . . .	191
<b>19 Les méthodes spéciales</b>	<b>195</b>
Édition de l'objet et accès aux attributs . . . . .	196
Les méthodes de conteneur . . . . .	202
Les méthodes mathématiques . . . . .	203
Les méthodes de comparaison . . . . .	207
Des méthodes spéciales utiles à <code>pickle</code> . . . . .	208
<b>20 L'héritage</b>	<b>213</b>
Pour bien commencer . . . . .	214
L'héritage simple . . . . .	214
L'héritage multiple . . . . .	219
Retour sur les exceptions . . . . .	220
<b>21 Derrière la boucle for</b>	<b>225</b>
Les itérateurs . . . . .	226
Les générateurs . . . . .	229
<b>22 TP : un dictionnaire ordonné</b>	<b>235</b>
Notre mission . . . . .	236
Correction proposée . . . . .	238
<b>23 Les décorateurs</b>	<b>245</b>
Qu'est-ce que c'est? . . . . .	246
En théorie . . . . .	246
Exemples d'applications . . . . .	255
<b>24 Les métaclasses</b>	<b>261</b>
Retour sur le processus d'instanciation . . . . .	262
Créer une classe dynamiquement . . . . .	264
Définition d'une métaclassse . . . . .	267

---

<b>IV Les merveilles de la bibliothèque standard</b>	<b>273</b>
<b>25 Les expressions régulières</b>	<b>275</b>
Que sont les expressions régulières? . . . . .	276
Le module <code>re</code> . . . . .	279
<b>26 Le temps</b>	<b>285</b>
Le module <code>time</code> . . . . .	286
Le module <code>datetime</code> . . . . .	290
<b>27 Un peu de programmation système</b>	<b>295</b>
Les flux standard . . . . .	296
Les signaux . . . . .	298
Interpréter les arguments de la ligne de commande . . . . .	301
Exécuter une commande système depuis Python . . . . .	306
<b>28 Un peu de mathématiques</b>	<b>309</b>
Pour commencer, le module <code>math</code> . . . . .	310
Des fractions avec le module <code>fractions</code> . . . . .	311
Du pseudo-aléatoire avec <code>random</code> . . . . .	313
<b>29 Gestion des mots de passe</b>	<b>317</b>
Réceptionner un mot de passe saisi par l'utilisateur . . . . .	318
Chiffrer un mot de passe . . . . .	319
<b>30 Le réseau</b>	<b>323</b>
Brève présentation du réseau . . . . .	324
Les sockets . . . . .	326
Le serveur . . . . .	330
Le client . . . . .	331
Un serveur plus élaboré . . . . .	332
<b>31 Des interfaces graphiques avec Tkinter</b>	<b>337</b>
Présentation de Tkinter . . . . .	338
Votre première interface graphique . . . . .	338
De nombreux widgets . . . . .	340
Les commandes . . . . .	344

<b>V</b>	<b>Annexes</b>	<b>347</b>
<b>32</b>	<b>Écrire nos programmes Python dans des fichiers</b>	<b>349</b>
	Mettre le code dans un fichier . . . . .	350
	Préciser l'encodage de travail . . . . .	352
	Mettre en pause notre programme . . . . .	352
<b>33</b>	<b>Distribuer facilement nos programmes Python avec cx_Freeze</b>	<b>355</b>
	En théorie . . . . .	356
	En pratique . . . . .	356
<b>34</b>	<b>De bonnes pratiques</b>	<b>361</b>
	Pourquoi suivre les conventions des PEP ? . . . . .	362
	La PEP 20 : tout une philosophie . . . . .	362
	La PEP 8 : des conventions précises . . . . .	364
	La PEP 257 : de belles documentations . . . . .	370
<b>35</b>	<b>Pour finir et bien continuer</b>	<b>375</b>
	Quelques références . . . . .	376
	Des bibliothèques tierces . . . . .	378