

Les exemples à télécharger sont disponibles à l'adresse suivante :  
**<http://www.editions-eni.fr>**  
Saisissez la référence ENI de l'ouvrage **RI2PYAPOO** dans la zone de recherche  
et validez. Cliquez sur le titre du livre puis sur le bouton de téléchargement.

## Avant-propos

### Chapitre 1

#### L'émergence de la POO

- 1. Ancêtres de la POO . . . . . 9
- 2. Besoin d'un langage de plus haut niveau . . . . . 11

### Chapitre 2

#### Les concepts de la POO

- 1. Modélisation . . . . . 13
- 2. Objet et classe . . . . . 16
- 3. Encapsulation . . . . . 18
- 4. Agrégation et composition . . . . . 22
  - 4.1 Agrégation . . . . . 22
  - 4.2 Composition . . . . . 23
- 5. Interface . . . . . 23
- 6. Énumération . . . . . 26
- 7. Héritage . . . . . 27
  - 7.1 Héritage simple . . . . . 27
  - 7.2 Classe abstraite . . . . . 29
  - 7.3 Héritage multiple . . . . . 31
  - 7.4 Du « bon » usage de l'héritage . . . . . 33

# 2 — Apprendre la POO

avec le langage Python

8.	Diagramme UML . . . . .	36
8.1	Structure vs comportement . . . . .	36
8.2	Diagramme de cas d'utilisation . . . . .	36
8.3	Diagramme de séquence . . . . .	38
9.	Exercices corrigés . . . . .	40
9.1	Classe simple . . . . .	40
9.2	Relations entre classes . . . . .	41
9.3	Héritage ? . . . . .	43
9.4	Repérage des classes . . . . .	45
9.5	Contenants et contenus . . . . .	46
9.6	Membres . . . . .	52

## Chapitre 3

### Présentation de l'environnement Python

1.	Python, troisième du nom ? . . . . .	57
2.	Installation . . . . .	58
2.1	python.org . . . . .	58
2.2	Windows . . . . .	59
2.3	Mac OS X . . . . .	64
2.4	Unix/Linux . . . . .	65
3.	Outillage . . . . .	66
3.1	pip . . . . .	66
3.2	IDLE . . . . .	67
3.3	PyCharm . . . . .	68
4.	Quelques concepts de base de Python . . . . .	69
4.1	Introduction . . . . .	69
4.2	Philosophie . . . . .	69
4.3	Langage interprété et compilé . . . . .	70
4.4	Duck typing . . . . .	71
4.5	Modules . . . . .	72

**Chapitre 4**  
**Les concepts de la POO avec Python**

- 1. Classe . . . . . 75
  - 1.1 Déclaration . . . . . 75
  - 1.2 Instance . . . . . 77
  - 1.3 Membres d'une classe . . . . . 79
    - 1.3.1 Attribut . . . . . 79
    - 1.3.2 Méthode . . . . . 82
  - 1.4 Constructeur . . . . . 86
  - 1.5 Destructeur . . . . . 89
  - 1.6 Exercices . . . . . 91
    - 1.6.1 Palindrome - méthode de classe . . . . . 91
    - 1.6.2 Palindrome - méthode d'instance . . . . . 92
    - 1.6.3 Puzzle . . . . . 93
    - 1.6.4 Logger . . . . . 95
- 2. Héritage . . . . . 96
  - 2.1 Construction . . . . . 96
  - 2.2 Polymorphisme . . . . . 101
  - 2.3 Héritage multiple . . . . . 103
  - 2.4 Exercices . . . . . 109
    - 2.4.1 Héritage « simple » . . . . . 109
    - 2.4.2 Puzzle . . . . . 111
    - 2.4.3 Héritage multiple - Diamant et paramètres  
de constructeur . . . . . 114
    - 2.4.4 Héritage multiple - Cas « réel » . . . . . 116
- 3. Agrégation et composition . . . . . 120
  - 3.1 Agrégation . . . . . 120
  - 3.2 Composition . . . . . 122
  - 3.3 Exercices . . . . . 124
    - 3.3.1 Le jour d'après . . . . . 124
    - 3.3.2 Immortel ? . . . . . 127
    - 3.3.3 Alternative à l'héritage multiple . . . . . 128

# 4 --- Apprendre la POO

avec le langage Python

4. Exception .....	132
4.1 Levée .....	132
4.2 Rattrapage .....	136
4.3 Éviter le masquage d'exception .....	140
4.4 Exception personnalisée .....	142
4.5 Exercice .....	144
5. Concepts de la POO non natifs .....	147
5.1 Classe abstraite .....	147
5.2 Interface .....	150
5.3 Encapsulation .....	151
6. Énumération .....	152
7. Duck typing .....	154

## Chapitre 5

### Un aperçu de quelques design patterns

1. Introduction .....	157
2. Singleton .....	159
3. Visiteur .....	166
3.1 Présentation .....	166
3.2 Exercice .....	172
4. Modèle - Vue - Contrôleur (MVC) .....	175
4.1 Présentation .....	175
4.2 Exercice .....	179
5. Abstract Factory .....	182
5.1 Présentation .....	182
5.2 Exercice .....	186

## Chapitre 6 Plus loin avec Python

1. Introduction	189
2. XML	190
2.1 Présentation	190
2.2 DOM	191
2.2.1 Lecture	191
2.2.2 Méthode par accès direct	192
2.2.3 Méthode par analyse hiérarchique	192
2.2.4 Écriture	194
2.3 SAX	196
2.3.1 Lecture	196
2.3.2 Écriture	198
3. JSON	199
4. IHM	202
4.1 Tkinter	202
4.1.1 Création d'une fenêtre	202
4.1.2 Ajout de widgets	204
4.1.3 Gestion des événements	206
4.2 Qt	208
4.2.1 Présentation	208
4.2.2 Création d'une fenêtre	208
4.2.3 Ajout de widgets	210
4.2.4 Gestion des événements	212
5. Bases de données	214
5.1 Présentation	214
5.2 SQLite	216
6. Multithreading	221
6.1 Présentation	221
6.2 Python et la programmation concurrente	223
6.3 Utilisation du module threading	224
6.4 Synchronisation	227

# 6 --- Apprendre la POO

avec le langage Python

6.5	Interblocage	229
7.	Développement web	231
7.1	Présentation	231
7.2	Création d'un projet Django	232
7.3	Développement web MVC	235
7.4	Quelques utilitaires intéressants	242

## Chapitre 7

### Quelques bonnes pratiques

1.	Introduction	245
2.	S'assurer avec des bases solides	246
2.1	De l'importance du socle	246
2.2	Structures de données et algorithmes communs	246
2.3	Un problème, plusieurs solutions	248
2.4	Choisir et maîtriser le bon outil	249
3.	Rester concis et simple	250
3.1	DRY	250
3.2	KISS	252
4.	Harmoniser l'équipe	253
4.1	Rédiger des conventions d'écriture	254
4.2	Revoir le code, tous ensemble	255
4.3	Documenter	256
4.4	Tester	256
4.5	Installer un environnement de déploiement continu	257
5.	Rejoindre une communauté	258
5.1	Participer à des conférences et groupes utilisateurs	259
5.2	Écouter la sagesse des foules	259
6.	Maîtriser les problèmes récurrents	261
6.1	Débogage	261
6.2	Traces	262
6.3	Monitoring	262

---

6.4	Performance	263
6.5	ETL	264
6.6	Bases de données : relationnelles ou non	265
6.7	Bases de données et ORM	266
6.8	Intégrations	267
6.9	Autres environnements logiciels	268
7.	Poursuivre sa croissance personnelle	268
7.1	Savoir poser des questions	269
7.2	Open source	270
7.3	Architecture	271
8.	Poursuivre sa croissance professionnelle	271
8.1	L'humain	271
8.2	Développement rapide et itérations	272
9.	Conclusion	272
	Index	273

# 8 \_\_\_\_\_ Apprendre la POO

avec le langage Python