

# Table des matières

<b>Chapitre 1. Introduction</b> . . . . .	13
1.1. Pourquoi un autre langage ? Pourquoi Scala ? . . . . .	13
1.2. Contexte . . . . .	14
1.3. Les caractéristiques de Scala . . . . .	16
1.3.1. Un langage orienté objet . . . . .	16
1.3.2. Un langage fonctionnel . . . . .	17
1.3.3. Un langage statiquement (et fortement) typé . . . . .	17
1.3.4. Un langage extensible . . . . .	17
1.4. Présentation de l'ouvrage . . . . .	18
<b>Chapitre 2. L'interpréteur Scala</b> . . . . .	23
2.1. Installer Scala . . . . .	24
2.2. Commentaires . . . . .	24
2.3. Exemple d'expressions . . . . .	25
2.4. Déclaration de constante . . . . .	26
2.5. Déclaration de variable . . . . .	27
2.6. Instances et invocations de méthodes . . . . .	27
2.7. Simplification de la syntaxe . . . . .	28
2.8. Conditionnelle . . . . .	31
2.9. Type d'une conditionnelle . . . . .	32
2.10. Blocs . . . . .	34
2.11. Fonction-objet . . . . .	34
2.12. La méthode <i>apply</i> . . . . .	36
2.13. Passer des fonctions en argument . . . . .	37
2.14. Définition de fonction-méthode . . . . .	39
2.15. Procédure . . . . .	39
2.16. Tableaux . . . . .	40
2.16.1. Accès aux éléments d'un tableau . . . . .	40
2.16.2. Construction d'un tableau . . . . .	40
2.16.3. <code>ArrayBuffer</code> et taille variable . . . . .	41
2.16.4. Itération . . . . .	42
2.17. Transformation de collections . . . . .	44
<b>Chapitre 3. Scala essentiel</b> . . . . .	51
3.1. Environnement de développement . . . . .	52
3.1.1. Installer Scala . . . . .	52

3.1.2.	Eclipse . . . . .	53
3.1.3.	Intellij IDEA . . . . .	53
3.1.4.	Netbeans . . . . .	54
3.1.5.	Autres environnements . . . . .	54
3.2.	Premier programme . . . . .	54
3.3.	Import . . . . .	56
3.4.	Déclaration de variables et constantes . . . . .	56
3.5.	Affichage et chaînes de caractères . . . . .	57
3.6.	Blocs et expressions . . . . .	59
3.7.	lazy . . . . .	61
3.8.	Def et définition de méthodes . . . . .	62
3.9.	Le point virgule et séparation des expressions . . . . .	69
3.10.	Conditionnelle . . . . .	71
3.11.	match / case et <i>pattern matching</i> . . . . .	71
3.12.	Exceptions . . . . .	73
3.13.	Collections et type paramétrique . . . . .	74
3.14.	Programmation objet . . . . .	77
3.14.1.	Hiérarchie générale des classes / traits / objets Scala . . . . .	77
3.14.2.	object . . . . .	78
3.14.3.	class . . . . .	80
3.14.4.	Trait . . . . .	83
3.15.	Java et Scala . . . . .	83
3.16.	Scaladoc . . . . .	85
3.16.1.	Comprendre les pages scaladoc . . . . .	85
3.16.2.	Ecrire des pages scaladoc . . . . .	93
3.17.	Conclusion . . . . .	95
<b>Chapitre 4.</b>	<b>Les tableaux . . . . .</b>	<b>97</b>
4.1.	Définition d'un tableau . . . . .	98
4.2.	Accès aux éléments . . . . .	100
4.3.	Itération sur les éléments d'un tableau . . . . .	101
4.4.	Création d'un tableau par transformation : for/yield . . . . .	102
4.5.	Transformation d'un tableau avec filter, map, sortWith, reduce . . . . .	104
4.5.1.	filter . . . . .	104
4.5.2.	map . . . . .	105
4.5.3.	sortWith . . . . .	106
4.5.4.	reduce . . . . .	107
4.6.	Tableaux de taille variable : ArrayBuffer . . . . .	108
<b>Chapitre 5.</b>	<b>Les principales collections . . . . .</b>	<b>113</b>
5.1.	Principe . . . . .	114
5.2.	Muable ou immuable? . . . . .	115
5.3.	Tuples . . . . .	117
5.4.	Map . . . . .	120
5.5.	Option . . . . .	127
5.6.	Either . . . . .	130
5.7.	Les listes . . . . .	131

5.8.	Les ensembles Set . . . . .	134
5.9.	Vector . . . . .	136
5.10.	Stack . . . . .	137
5.11.	Queue . . . . .	138
5.12.	Range . . . . .	138
5.13.	Stream . . . . .	139
5.13.1.	Rappel sur les listes . . . . .	139
5.13.2.	Liste et Stream . . . . .	140
5.14.	String . . . . .	143
5.15.	ArrayBuffer . . . . .	144
<b>Chapitre 6.</b>	<b>Les collections Scala en détail . . . . .</b>	<b>147</b>
6.1.	scala.collection . . . . .	148
6.1.1.	Traversable . . . . .	149
6.1.2.	Iterable . . . . .	149
6.1.3.	Seq . . . . .	149
IndexedSeq	. . . . .	150
LinearSeq	. . . . .	150
6.1.4.	Set . . . . .	151
6.1.5.	Map . . . . .	151
6.2.	Opérations courantes . . . . .	151
6.2.1.	Créer une nouvelle collection . . . . .	151
6.2.2.	Ajouter/enlever un élément . . . . .	153
6.2.3.	Mettre à jour un élément . . . . .	153
6.2.4.	Opérations ensemblistes sur les collections . . . . .	154
6.2.5.	Conversions . . . . .	154
6.2.6.	Quelques méthodes importantes de Iterable . . . . .	155
Propriétés . . . . .		155
Sélection et découpage . . . . .		156
Chaînes de caractères . . . . .		159
Transformation d'une collection . . . . .		160
6.2.7.	Quelques méthodes importantes de Seq . . . . .	168
6.3.	Collections parallèles : par et seq . . . . .	169
6.3.1.	Conversion : par . . . . .	169
6.3.2.	Ordre des traitements . . . . .	171
6.3.3.	Mise à jour de variables partagées . . . . .	171
6.3.4.	Convertir en version non parallèle : seq . . . . .	172
6.3.5.	Collections parallèles : fold, reduce et aggregate . . . . .	173
6.4.	Les vues ( <i>lazy view</i> ) : view et force . . . . .	174
6.5.	Java et Scala : JavaConversions . . . . .	176
<b>Chapitre 7.</b>	<b>Programmation objet . . . . .</b>	<b>179</b>
7.1.	Déclaration d'une classe . . . . .	180
7.2.	Accès public, private, protected . . . . .	180
7.3.	Accesseurs ( <i>getters</i> et <i>setters</i> ) . . . . .	182
7.4.	Constructeurs . . . . .	184
7.5.	Classes imbriquées . . . . .	189

7.6.	Classes anonymes . . . . .	191
7.7.	Type et this . . . . .	193
7.8.	Transtypage . . . . .	194
7.9.	Object . . . . .	195
7.9.1.	Objet singleton . . . . .	195
7.9.2.	Type . . . . .	198
7.10.	Méthodes . . . . .	198
7.10.1.	Nommage . . . . .	198
7.10.2.	Invocation . . . . .	200
	Principe . . . . .	200
	Arité nulle . . . . .	201
	Arité une . . . . .	201
	Opérateurs . . . . .	202
	Opérateurs associatifs à droite et le caractère ':' . . . . .	202
	Fonctions d'ordre supérieur . . . . .	204
	Arguments nommés . . . . .	204
7.10.3.	apply . . . . .	204
7.10.4.	toString . . . . .	205
7.10.5.	==, eq, equals, hashCode . . . . .	206
7.10.6.	Code du constructeur . . . . .	209
7.10.7.	Extracteurs et unapply . . . . .	210
	Principe . . . . .	210
	Extracteur d'un argument . . . . .	215
	Extracteur booléen . . . . .	215
	Extracteurs de séquences . . . . .	216
7.11.	case class . . . . .	217
7.12.	Héritage . . . . .	219
7.12.1.	extends . . . . .	219
7.12.2.	Constructeurs et super-classe . . . . .	219
7.12.3.	override . . . . .	220
7.12.4.	Classe abstraite . . . . .	220
7.12.5.	final . . . . .	221
7.12.6.	sealed . . . . .	222
7.12.7.	Héritage et classes anonymes . . . . .	223
7.13.	Conversions implicites . . . . .	224
7.13.1.	Classe implicite . . . . .	224
7.13.2.	Méthode implicite . . . . .	226
7.13.3.	Contrôle de conversion . . . . .	227
7.13.4.	Applications . . . . .	228
	Extension de classes . . . . .	228
	Interpolation de chaînes . . . . .	228
	Opérateurs et conversion implicite . . . . .	230
7.14.	Traits . . . . .	232
7.14.1.	Principe . . . . .	232
7.14.2.	Illustration et exemple simple . . . . .	234
7.14.3.	Synthèse . . . . .	243

7.15. Classe Valeur (ou <i>Value Class</i> ) . . . . .	246
7.16. Paquetages . . . . .	247
7.16.1. Déclaration . . . . .	247
7.16.2. <code>private</code> et <code>protected</code> . . . . .	252
7.16.3. <code>package object</code> . . . . .	253
7.16.4. <code>import</code> . . . . .	254
<b>Chapitre 8. <i>Pattern matching</i> (filtrage de motifs)</b> . . . . .	<b>259</b>
8.1. <code>match</code> . . . . .	260
8.1.1. Principes . . . . .	260
8.1.2. <code>match</code> et <code>switch</code> . . . . .	261
8.1.3. Détection de type . . . . .	262
8.1.4. Conditionnelle de clause (garde) . . . . .	264
8.1.5. Appariement de constantes par motif . . . . .	264
8.1.6. tuples . . . . .	265
8.1.7. Collections . . . . .	268
8.1.8. Expressions régulières . . . . .	269
8.1.9. <code>case class</code> . . . . .	271
Déclaration . . . . .	271
Affichage . . . . .	271
Évaluation . . . . .	272
Dérivation . . . . .	273
Simplification . . . . .	274
8.1.10. Exceptions . . . . .	275
8.2. <code>for/yield</code> . . . . .	275
8.2.1. Parcours d'une collection . . . . .	276
8.2.2. Filtrage . . . . .	276
8.2.3. Itérateur . . . . .	277
8.3. Fonctions partielles et <code>case</code> . . . . .	279
<b>Chapitre 9. Les fonctions</b> . . . . .	<b>283</b>
9.1. Introduction . . . . .	284
9.1.1. Programmation impérative . . . . .	284
9.1.2. Programmation fonctionnelle . . . . .	285
9.2. Fonctions et méthodes . . . . .	287
9.3. Conversion de méthode en fonction . . . . .	290
9.4. Fonctions anonymes . . . . .	291
9.5. Fonctions partielles . . . . .	292
9.6. Type des paramètres . . . . .	294
9.7. Fonctions d'ordre supérieur . . . . .	295
9.8. Curryfication . . . . .	297
9.9. Valeurs implicites de paramètres . . . . .	299
9.10. Structures de contrôle . . . . .	300
9.11. Fermeture (clôture ou <i>closure</i> ) . . . . .	302
9.12. Exemples . . . . .	302
9.12.1. <code>repeat... until</code> . . . . .	302
9.12.2. Délais et Threads . . . . .	303

<b>Chapitre 10. Types paramétriques</b>	305
10.1. Principes	306
10.2. Bornes	306
10.2.1. Borne supérieure ( <i>upper bound</i> )	306
10.2.2. Borne de conversion ( <i>view bound</i> )	308
10.2.3. Borne inférieure ( <i>lower bound</i> )	309
10.2.4. Borne contextuelle ( <i>context bound</i> )	310
10.2.5. Composition des bornes	311
10.3. Variances	312
10.3.1. Covariance	312
10.3.2. Contravariance	313
<b>Chapitre 11. XML</b>	315
11.1. Document XML	316
11.2. child	317
11.3. label	318
11.4. Attributs	318
11.5. Insertion d'expressions	319
11.6. CData	321
11.7. <i>Pattern matching</i>	321
11.8. Réécriture	323
11.9. Recherche de chemins	324
<b>Chapitre 12. Future et Promise</b>	327
12.1. Future	328
12.1.1. Principe	328
12.1.2. Try	329
12.1.3. Attendre la fin d'un Future	329
12.1.4. Duration	330
12.1.5. Gestionnaires d'événements	330
12.1.6. map	331
12.1.7. recover	333
12.1.8. fallbackTo	333
12.1.9. for	334
12.1.10.andThen	335
12.2. Promise	335
<b>Chapitre 13. Acteurs Akka</b>	339
13.1. Future et acteurs	340
13.2. Création d'un acteur	342
13.3. Communication et case class	346
13.4. Logging	347
13.5. context	348
13.6. Hiérarchie d'acteurs	350
13.7. preStart et postStop	352
13.8. ask et Future	353
13.9. pipeTo	355

13.10. Superviseur . . . . .	356
13.11. system.shutdown . . . . .	357
13.12. Et plus . . . . .	357
<b>Chapitre 14. Les continuations délimitées . . . . .</b>	<b>359</b>
14.1. Motivation . . . . .	360
14.2. <i>Continuation Passing Style</i> . . . . .	361
14.3. reset et shift . . . . .	363
14.3.1. Principe . . . . .	363
14.3.2. Type de reset . . . . .	366
14.3.3. Type de la continuation . . . . .	367
14.4. Arguments de continuation . . . . .	367
14.5. Clôture et contexte . . . . .	368
14.5.1. Itération . . . . .	368
14.5.2. Récursivité . . . . .	370
14.6. Conclusion . . . . .	372
<b>Chapitre 15. Conclusion générale . . . . .</b>	<b>375</b>
<b>Bibliographie . . . . .</b>	<b>379</b>
<b>Index . . . . .</b>	<b>381</b>