

Pierre Alexis et Hugues Bersini

*1 étude de
cas inspirée de
Facebook !*

Apprendre la programmation web avec **Python** et **Django**

**Principes et bonnes pratiques
pour les sites web dynamiques**

EYROLLES

Pierre Alexis et Hugues Bersini



Apprendre la programmation web avec **Python** et **Django**

**Principes et bonnes pratiques
pour les sites web dynamiques**

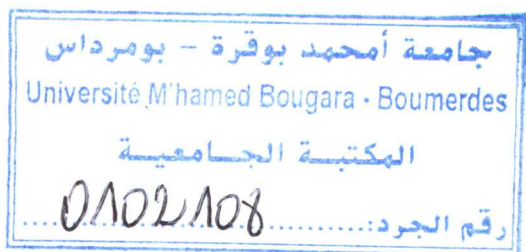


Table des matières

Avant-propos	V
Pourquoi cet ouvrage ?	VII
Le choix de Python et de Django	VII
À qui s'adresse cet ouvrage ?	X
Le plan du cours	XI
Remerciements	XIII

PREMIÈRE PARTIE

Les notions essentielles	1
---------------------------------------	----------

CHAPITRE 1

Comment fonctionne un site web ?.....	3
Qu'est-ce que le Web ?	4
Fonctionnement d'un site web statique	5
Le protocole HTTP	7
L'URL, adresse d'une page web	8
Le serveur web : à la fois ordinateur et logiciel	9
Des sites web qui se mettent à jour tout seuls	9
Fonctionnement d'un site web dynamique	11
Utilisation d'une base de données	12
Passage de paramètres à une page web	13
Ai-je bien compris ?	14

CHAPITRE 2

Programmation orientée objet et framework MVC	15
Des programmes modulaires	16
Une écriture simplifiée	16
Des modules réutilisables	16
Un travail d'équipe facilité et plus efficace	16
Les langages orientés objet	17
Les cas d'utilisation (use cases) et le MVC	18

Le principe de la séparation modèle-vue-contrôleur (MVC)	18
Le diagramme des cas d'utilisation (use cases)	20
Correspondances entre MVC et cas d'utilisation	21
Django et le MVC	23
Le framework Django	23
Ai-je bien compris ?	24

CHAPITRE 3

Rappels sur le langage Python 25

Qualités et défauts de Python	26
Qualité : la simplicité	26
Défaut : la simplicité !	27
Les bases : variables et mémoire centrale	28
Déclaration et initialisation des variables	28
Type des variables	29
Modification et transtypage des variables	29
Copie de variables	30
Se renseigner sur une variable	30
Quelques opérations sur les types simples	31
Le type int	31
Le type float	32
Le type string	33
Les types composites : listes et dictionnaires	34
Les listes	35
Les dictionnaires	36
Les instructions de contrôle	37
Les instructions conditionnelles	37
<i>Aiguillage à deux directions : instruction if (...) else</i>	37
<i>Aiguillage à plus de deux directions : instruction if (...) elif (...) else</i>	39
Les boucles	40
<i>La boucle while</i>	40
<i>La boucle for (...) in</i>	41
Les fonctions	43
Les variables locales et les variables globales	44
La programmation objet	45
Les classes et les objets	45
L'association entre classes	47
<i>Héritage et polymorphisme</i>	51
Import et from : accès aux bibliothèques Python	54
Ai-je bien compris ?	55

CHAPITRE 4

Rappels sur HTML5, CSS et JavaScript..... 57

Structurer les pages web avec HTML5	59
Le concept de « balises »	60
Structure d'un document HTML	63
<i>L'encodage de la page</i>	65
Quelques éléments HTML	66
<i>Principaux éléments de structuration HTML</i>	66
<i>Éléments de structuration annexes : en-têtes, pieds de page et sections</i>	68
<i>Les liens hypertextes</i>	70
<i>Les listes</i>	71
<i>Les images</i>	72
<i>Mise en évidence du texte</i>	73
<i>Les formulaires</i>	75
<i>Autres éléments HTML</i>	78
Mettre en forme avec les feuilles de styles CSS	78
Les propriétés CSS	79
Les sélecteurs CSS	79
<i>Sélectionner toutes les balises de même nom.</i>	80
<i>Sélectionner un élément particulier : id en HTML et # en CSS</i>	80
<i>Sélectionner quelques éléments de même nature : class en HTML et . en CSS</i> ...	81
<i>Appliquer une propriété seulement quand l'élément est dans un état donné.</i> ...	81
<i>Combiner les sélecteurs</i>	82
<i>Sélectionner tous les éléments</i>	83
Lier CSS et HTML	83
<i>Placer le code CSS dans les balises HTML</i>	83
<i>Placer le code CSS dans l'en-tête du fichier HTML</i>	84
<i>Placer le code CSS dans un fichier séparé</i>	85
Dimensions des éléments en CSS	86
Imbrication des éléments (boîtes)	87
Positionnement par défaut des éléments en CSS	88
Sortir certains éléments du flux	89
Application à un exemple concret plus complet	91
Dynamiser les pages web « à la volée » avec JavaScript	96
Les événements	96
Langage de programmation de scripts	96
Un premier exemple de DHTML	97
jQuery et les frameworks JavaScript	99
Ai-je bien compris ?	101

CHAPITRE 5

Mise en application : un site web inspiré de Facebook 103

Notre site web : Trombinoscoop	104
Les cas d'utilisation	105
Maquette du site : les wireframes	106
L'écran d'authentification	107
L'écran de création d'un compte	108
L'écran d'accueil	108
L'écran de modification du profil	108
L'écran d'affichage d'un profil	108
L'écran d'ajout d'un ami	108
Scénario complet du site	112
Modèle de données et petit rappel sur les bases de données relationnelles	113
Clé primaire et clé étrangère	114
<i>Relation 1-n</i>	114
<i>Relation 1-1</i>	116
<i>Relation n-n</i>	117
La problématique de la mise en correspondance relationnel/objet	118
<i>Avec Django</i>	119
Retour au modèle de données de Trombinoscoop : son diagramme de classes	119
<i>Des personnes : étudiants et employés</i>	120
<i>... qui travaillent ou étudient à l'université</i>	120
<i>... et qui échangent des messages avec des amis</i>	122
Ai-je bien compris ?	123

CHAPITRE 6

Premier contact avec les bases relationnelles et SQL

à partir d'un exemple en CGI.....	125
Analyse du carnet d'adresses : des cas d'utilisation au modèle de données	127
Trois cas d'utilisation simples	127
Maquette des écrans (wireframes)	127
Le modèle de données du carnet	127
Création de la base de données avec SQLite	129
Accès à la base de données via SQL	131
Syntaxe des requêtes SQL les plus courantes	131
Quelques exemples liés à notre base de données	132
Réalisation du carnet d'adresses avec SQL et CGI	133
Lancement d'un serveur web Python	133
L'écran d'accueil de l'ULB	135
La page principale du carnet d'adresses	136

La liste des employés d'un service	140
Ajout d'un employé	143
CGI : ce qu'il ne faut plus faire	145
Ai-je bien compris ?	145

DEUXIÈME PARTIE

Mise en application avec Django 147

CHAPITRE 7

Les vues Django : orchestration et architecture 149

Utilité des vues	151
Le fichier urls.py	152
Le fichier views.py	153
Enfin ! Notre première page web en Django	155
Lancement de l'environnement Eclipse et création du projet	155
Le fichier urls.py	158
Le fichier views.py	158
Importation de la fonction dans urls.py	160
Configuration de Django pour qu'il écrive du XHTML 5	160
Test de notre ébauche de site	161
Bel effort, mais... ..	163
Ai-je bien compris ?	163

CHAPITRE 8

Les templates Django : séparation et réutilisation des rendus HTML 165

Principe des templates	166
Notre premier template	167
Dynamisons ce premier template	170
Le langage des templates	172
Les variables	172
Formatage des variables	172
Sauts conditionnels et boucles	173
Héritage et réutilisation de templates	174
Et si on avançait dans la réalisation de Trombinoscoop ?	176
Amélioration visuelle de la page de login	180
Ai-je bien compris ?	184

CHAPITRE 9

Les formulaires Django	185
Patron courant de gestion des formulaires	186
L'objet request	187
Traitement du formulaire de login	188
Ajout du formulaire dans la vue	188
Gestion du message d'erreur	189
Présentation du message d'erreur	190
La bibliothèque forms de Django	190
Création d'un formulaire avec la bibliothèque forms	191
Intégration du formulaire dans la vue et le template	192
Validation du formulaire	194
Présentation des messages d'erreur	196
Validation de l'adresse de courriel et du mot de passe	197
Faut-il se contenter d'un seul visiteur autorisé ?	199
Ai-je bien compris ?	199

CHAPITRE 10

Les modèles Django	201
Les modèles Django	202
Création d'un premier modèle	202
Le modèle Personne	203
Configuration	204
Création de la base de données et du compte administrateur (superutilisateur)	205
Création des autres modèles et de leurs liens	207
Le modèle Message : relation 1-n	207
La relation « ami » : relation n-n	208
Les modèles simples Faculté, Campus, Fonction et Cours	209
Les modèles Employé et Étudiant : héritage	209
Le lien entre Faculté et Personne : relation 1-n	210
Régénération de la base de données	211
Utilisation des modèles	211
Création et modification d'un enregistrement	211
Récupération de plusieurs enregistrements	212
Tri des données	212
Récupération d'un enregistrement unique	212
Suppression d'enregistrements	213
Accès à des objets « liés »	213
Remplissage de la base de données	214
Configuration de l'interface d'administration des bases de données	214

Gestion de la base de données avec l'interface d'administration	216
Authentification utilisant la base de données	219
Les ModelForm	221
Création du formulaire Étudiant dans le fichier forms.py	221
Création de l'URL et de la vue de création de compte	222
Création du template de création de compte	223
Un peu de mise en forme	223
Finalisation de la page de création de compte	225
Création de deux formulaires : un pour les étudiants et un pour les employés ..	225
Gestion des deux formulaires dans la vue	226
Gestion des deux formulaires dans le template	227
Un peu de dynamisme	229
Ai-je bien compris ?	230

CHAPITRE 11

Comprendre et utiliser les sessions

231

À quoi servent les sessions	232
Les sessions selon Django	233
Utilisation d'une session	234
Configuration	234
Maniement d'une variable de session	235
Enregistrement de l'utilisateur authentifié	235
Vérification que l'utilisateur est bien authentifié	236
Utilisation des données de la session	237
Que trouve-t-on dans le cookie ?	238
Que trouve-t-on dans la session ?	238
Protection des pages privées	238
Amélioration de notre page d'accueil	239
Personnalisation de la bannière	240
Division du corps de la page	243
Liste des messages	245
<i>Récupération des messages de la liste</i>	<i>245</i>
<i>Présentation de la liste des messages</i>	<i>247</i>
Liste des amis	248
Publication d'un message à destination de ses amis	250
Ai-je bien compris ?	251

CHAPITRE 12

En finir avec Trombinoscoop

253

La page d'ajout d'un ami	254
--------------------------------	-----

Ajout d'un formulaire dans forms.py	254
Création de la vue add_friend	255
Création du template add_friend.html	256
Ajout d'une URL dans urls.py	256
Ajout du lien dans la page d'accueil	256
La page de visualisation d'un profil	257
Création du template show_profile.html	258
Création de la vue show_profile	259
Ajout de l'URL dans urls.py	260
Amélioration de la présentation dans style.css	260
Ajout des liens dans la page d'accueil	260
La page de modification d'un profil	262
Création du template modify_profile.html	262
Création de la vue modify_profile	263
Ajout de l'URL dans urls.py	264
Ajout des liens dans la page d'accueil	264
Ai-je bien compris ?	265

CHAPITRE 13

Des sites web encore plus dynamiques avec Ajax..... 267

Exemple de l'interactivité attendue entre client et serveur	268
Validation de la création d'un compte avec Ajax	271
Ajout d'une URL pour la requête Ajax	272
Vue traitant la requête Ajax	272
Ajout du code JavaScript	274
<i>Détecter que l'utilisateur a bien terminé de remplir le champ courriel</i>	<i>274</i>
<i>Validation du courriel saisi</i>	<i>276</i>
<i>Insérer la liste des erreurs au-dessus du champ « fautif »</i>	<i>277</i>
Ajout d'un ami via Ajax	279
Ajout d'un champ texte et d'un lien	279
Ajout de l'URL et de la vue	280
Création du JavaScript d'ajout d'un ami	281
Insertion du HTML dans la page web	283
Conclusions	284
Ai-je bien compris ?	285

ANNEXE A

Installation de l'environnement de développement 287

Que faut-il installer ?	288
Python	288

Django	289
Eclipse	289
En résumé	290
Installation de Python	292
Pour Windows	292
Pour Mac OS X	295
Vérification de l'installation	296
Installation de Django	297
Pour Windows	297
Pour Mac OS X	299
Pour Ubuntu	300
Vérification de l'installation	301
Installation de Java	302
Pour Windows et Mac OS X	302
Pour Ubuntu	302
Installation d'Eclipse	303
Installation du plug-in Eclipse PyDev	305
Installation de Web Developer Tools pour Eclipse	310
Premier projet de test	311
Ai-je bien compris ?	314
Index.....	315

Hugues Bersini et Pierre Alexis

Membre de l'Académie Royale de Belgique, **Hugues Bersini** enseigne l'informatique et la programmation aux facultés polytechnique et Solvay de l'Université Libre de Bruxelles, dont il dirige le laboratoire d'Intelligence Artificielle. Il est l'auteur de très nombreuses publications (systèmes complexes, génie logiciel, sciences cognitives et bioinformatique).

Pierre Alexis est titulaire d'une licence en informatique et travaille au service informatique de la plus grande banque privée belge. Intéressé depuis toujours par l'enseignement et la vulgarisation informatique, il assiste Hugues Bersini à l'Université libre de Bruxelles et dispense un cours de Django/Python à destination d'étudiants en ingénierie commerciale.

Enfin un ouvrage pour étudiants détaillant tous les principes de la programmation web moderne, avec l'un des frameworks de développement web les plus ambitieux : Django, basé sur le langage Python !

Un manuel autonome reprenant tous les fondements de la programmation web, au fil d'une étude de cas inspirée de Facebook

Abondamment illustré d'exemples inspirés de Facebook et rappelant les bonnes pratiques du domaine (modèle MVC, diagrammes UML, patterns), voici un livre de cours magistral et moderne sur la programmation web dynamique, que tous les enseignants en informatique peuvent utiliser. Complet et autonome, il pose solidement les fondamentaux de la conception web, avec ou sans framework : HTML5/CSS3, dynamisme alimenté par bases relationnelles SQL, sessions, JavaScript et Ajax, sans oublier de fournir au lecteur d'essentiels rappels en programmation objet, voire de montrer... ce qu'il ne faut pas faire en CGI !

Le langage Python et le framework Django sont introduits en douceur, et l'utilisation des vues, templates, formulaires et modèles Django, conformément aux principes MVC exposés dans la première partie, est illustrée au fil de l'étude de cas. L'annexe complète le manuel par une description pas à pas de l'installation de l'environnement de développement, tant sous Windows et Mac OS X que sous GNU/Linux : Python, Django, Eclipse, PyDev et les Web Developer Tools.

À qui s'adresse ce livre ?

- Étudiants en informatique (IUT, écoles d'ingénieurs) et leurs enseignants ;
- Développeurs web (PHP, Java, etc.) qui souhaitent passer à Python & Django ;
- Développeurs C, C++ qui souhaitent une introduction systématique à la programmation web.

Au sommaire

Fonctionnement d'un site web • HTTP et notion de serveur web • Statique versus dynamique • Passer des paramètres à une page • **Programmation orientée objet et framework MVC** • Modularité, simplicité et réutilisabilité • Travail d'équipe • Cas d'utilisation (use cases) • Séparation modèle-vue-contrôleur • **Le langage Python** • Déclaration et initialisation des variables • Types et transtypage • Copie de variables • Opérations sur les types simples int, float et string • Listes et dictionnaires • Instructions conditionnelles et boucles while • Fonctions • Variables locales et globales • Classes et objets • Variables de classe (attributs statiques) et d'instance • Héritage et polymorphisme • Accès aux bibliothèques Python avec import et from • **Structurer les pages avec HTML5** • Encodage • Balises et éléments HTML : en-têtes, pieds de page et sections, liens hypertextes, listes, images, mise en évidence du texte, formulaires, etc. • **Mettre en forme avec les feuilles de styles CSS** • Propriétés CSS et principes de sélection d'éléments (sélecteurs) • Dimension et imbrication des éléments • Positionnement par défaut • Sortir des éléments du flux • **Dynamiser les pages à la volée avec JavaScript** • Événements • DHTML, jQuery et frameworks JavaScript • **Une étude de cas inspirée de Facebook** • Cas d'utilisation et maquette du site (wireframes) • Modèle de données et rappel sur les bases relationnelles • Clés primaires et étrangères • Relation 1-n, 1-1 et n-n • Correspondance relationnel/objet (ORM) • Diagramme de classes • **Premier contact avec les bases relationnelles SQL** • Un exemple en CGI : ce qu'il ne faut plus faire • **Les vues Django : orchestration et architecture** • Configurer Django pour écrire du XHTML 5 • **Les templates Django : séparation et réutilisation des rendus HTML** • Langage des templates • Variables, sauts conditionnels et boucles, héritage et réutilisation de templates • **Les formulaires Django** • L'objet request • Formulaire de login • Gestion du message d'erreur • La bibliothèque forms de Django • Valider l'adresse et le mot de passe • **Les modèles Django** • Créer la base de données et le compte administrateur • Régénérer la base • Configurer l'interface d'administration de la base de données • Générer des formulaires à partir de modèles (ModelForms) • **Comprendre et utiliser les sessions** • Manier une variable de session • Authentification • Cookie versus session • Protéger des pages privées • **Des sites encore plus dynamiques grâce à Ajax** • **Annexe : installer l'environnement de développement sous Linux, Mac OS X et Windows** • Python, Django, Eclipse, PyDev et les Web Developer Tools.



Sur le site www.editions-eyrolles.com

Le code source de l'étude de cas est disponible sur le site d'accompagnement du livre.

Code article : G13499
ISBN : 978-2-212-13499-5



29,00 €

www.editions-eyrolles.com
Groupe Eyrolles | Diffusion Geodif