



UNIVERSITÉ
LAVAL

Faculté des sciences
de l'administration

SIO-3103

Introduction à la programmation Web

Section Z1

3 crédits

Enseignant :

Dzenan Ridjanovic, Ph.D.

Pavillon Palasis-Prince

Université Laval, Québec, G1V 0A6

Courriel :

dzenanr [at] gmail [dot] com

Le cours sur Web :

Jeudi 15h30 – 18h15

<https://classevirtuelle.ulaval.ca/>

Site Web :

<https://www.assembla.com/wiki/show/ul-fsa-web>

Forum :

<http://ul-fsa-web.961705.n3.nabble.com/>

Remarque : le genre masculin est utilisé tout au long de ce plan de cours dans le seul but d'en alléger la lecture.

1. Description du cours

Ce cours a pour objectif de permettre à l'étudiant de se familiariser avec les concepts de base de la programmation web, le cycle de développement d'une application et les différents outils d'aide au développement. L'étudiant apprendra à développer une application web fonctionnelle avec le langage de programmation orienté objet Python et le cadre d'application Django, en utilisant une approche de développement par spirale et des outils d'aide au développement comme Eclipse, Assembla et Mercurial.

Ce cours s'adresse aux étudiants ayant peu d'expérience de programmation, mais peut aussi convenir aux étudiants voulant parfaire leurs connaissances dans la programmation web ou les différentes technologies utilisées dans le cadre de ce cours. Les concepts abordés aideront les futurs spécialistes en systèmes d'information à se familiariser avec les différents aspects de la conception et réalisation d'une application web, ce qui leur donnera la chance de pouvoir mieux estimer l'envergure, la durée et les embûches de la création d'un système d'information. En effet, il est important pour un analyste en systèmes d'information de maîtriser à la fois le langage des gestionnaires et celui des informaticiens.

Remarque concernant la charge de travail : Ce cours

universitaire de premier cycle exige en moyenne 9 heures de travail par semaine. Soyez donc bien conscient qu'il est essentiel pour votre apprentissage et pour la réussite du cours d'avoir du temps à y consacrer.

2. Objectifs

Ce cours a pour principal objectif de familiariser les étudiants aux différents concepts de base de la programmation web.

De plus, le cours vise à faire acquérir des connaissances de base dans ces différentes technologies et outils d'aide au développement:

Python

Langage de programmation orienté objet populaire, utilisé autant pour les applications clients que serveurs.

Django

Cadre d'application web qui utilise Python pour fournir du contenu dynamique à des pages web.

XHTML (eXtensible HyperText Markup Language)

Langage de balisage; version plus stricte et épurée de HTML.

CSS (Cascading Style Sheets)

Langage de présentation qui permet de contrôler l'agencement et la stylisation hors des documents XHTML.

Eclipse

Environnement de développement intégré (IDE) utilisé pour écrire le code de programmation.

Mercurial

Mercurial est un système de gestion de versions permettant en particulier la gestion de version décentralisée. Il est disponible sur plusieurs plateformes : systèmes Unix, Windows, Mac OS X. Mercurial est écrit principalement en Python.

Assembla

Site web dédié à faciliter la collaboration entre développeurs de logiciels à l'aide de différents outils avancés et de pratiques de gestion de projets.

Assembla fournit un espace Mercurial où le code de programmation peut être sauvegardé de façon similaire à la sauvegarde de

données dans une base de données.

Outre que Mercurial, Assembla offre plusieurs autres outils comme Wiki pour organiser de l'information de base sur notre projet et Scrum pour faire la gestion des contributions individuelles au projet.

Google App Engine (GAE)

GAE pour Python et Java propose une solide offre d'hébergement de serveur d'applications. Cette solution de cloud computing est conçue comme « platform as a service » : Google fournit l'infrastructure complète ainsi que l'environnement pour héberger l'application. App Engine propose ainsi de nombreux services dont notamment un système de base de données appelé datastore (basé sur Google Big Table).

GAE permet d'exécuter vos applications web sur l'infrastructure de Google. Les applications App Engine sont faciles à construire, faciles à maintenir et supportent facilement la montée en charge de votre trafic et de vos besoins croissants de stockage de données. Avec App Engine, il n'y a pas de serveurs à maintenir. Vous chargez juste vos applications et elles sont aussitôt disponibles pour vos utilisateurs.

Avec l'environnement d'exécution Python App Engine, vous pouvez implémenter votre application en utilisant le langage de programmation Python et l'exécuter dans un interpréteur Python optimisé. App Engine inclut des APIs riches et des outils pour le développement d'applications web en Python, incluant des APIs de modélisation de données riches, un framework d'application web simple d'utilisation et des outils pour manager et accéder aux données de votre application. Vous pouvez aussi profiter d'une grande variété de bibliothèques matures et de frameworks pour le développement d'application web en Python, tel que Django.

Vous pouvez mettre à disposition vos applications depuis votre propre nom de domaine (tel que <http://www.example.com/>) en utilisant Google Apps. ou vous pouvez utiliser directement le nom qui vous est fourni sur le domaine appspot.com. Vous pouvez partager votre application avec le monde ou limiter l'accès aux membres de votre organisation.

Google Datastore

Une application de GAE ne peut pas écrire sur le système de fichiers. Une application peut lire des fichiers mais seulement les fichiers chargés avec le code de l'application. L'application doit utiliser la base de données App Engine ("datastore"), memcache ou d'autres services pour toutes les données qui persistent entre les requêtes.

La base de données de GAE appelée datastore repose sur le SGBD

Big Table développé en interne par Google. Celui-ci a la particularité d'être hautement scalable et s'appuie sur un système de fichiers distribués géré par GFS (Google File System).

Big Table est une Map ordonnée multidimensionnelle. Les données (ou entités) sont stockées sous forme de <Clé, Valeur> dans des partitions appelées tablets. Des tablettes META1 référencent l'ensemble des tablettes contenant les entités. Une tablette maître appelée META0 référence les tablettes META1 et permet au client d'accéder aux tablettes de la base. Pour optimiser les performances, un mécanisme de cache permet au client de s'adresser directement aux tablettes META1 et aux tablettes contenant les entités.

3. Méthodes pédagogiques

Le matériel didactique et la formule utilisés vous permettent d'adopter une démarche d'apprentissage axée sur la collaboration, la communication et le travail individuel. Ces méthodes pédagogiques vous amèneront à gérer votre temps de manière flexible mais structurée, et ce, afin de respecter le rythme du cours.

Le cours est basé sur le cours donné par Charles Severance l'auteur du livre Using Google App Engine - Building Web Applications :

<http://www.si539.com/>

Voici la liste des activités d'apprentissage qui seront utilisées durant la session :

Lectures obligatoires

GAE : *Using Google App Engine - Building Web Applications*, Charles Severance, O'Reilly, 2009

<http://oreilly.com/catalog/9780596801601>

Django : *Écrire votre première application Django*, version web, 2010.

<http://docs.djangoproject-fr.org/intro/tutorial01.html>

Django de zéro : vos projets Python sur le Web !

<http://www.siteduzero.com/tutoriel-3-14384-django-de-zero-vos-projets-python-sur-le-web.html>

Lectures facultatives

The Django Book, Adrian Holovaty and Jacob Kaplan-Moss, version web, 2010.

<http://www.djangobook.com/>

<http://docs.djangoproject.com/en/dev/intro/whatsnext/#intro-whatsnext>

Présentations (Notes)

<http://www.si539.com/resources.php>

Nous utiliserons le contexte de la programmation web pour expliquer les concepts de base de la programmation. Ce choix est principalement justifié par les tendances de développements actuelles et le potentiel de réutilisabilité des concepts web à la suite du cours. En effet, plutôt que d'utiliser un système d'affichage relié à une plateforme ou un langage, l'affichage web permettra aux étudiants de créer ou éditer des pages web en utilisant n'importe quelle plateforme. De plus, l'utilisation de XHTML et CSS (et non HTML) permettra aux étudiants de connaître les standards web établis.

Nous utiliserons le cadre d'application Django pour fournir le contenu dynamique aux pages web. Le choix de Django est justifié par son accent sur le langage Python. De plus, Django utilise une approche de différenciation marquée entre la portion modèle et vue (MVC) d'une application web, ce qui permet de bien répliquer l'environnement de travail en entreprise pour les positions de designer web et programmeur.

La gestion de projets se fera en utilisant deux outils principaux, soit Assembla et Mercurial. Ces outils permettront aux étudiants d'avoir une version unique de leur projet (en ligne), de présenter leur projet, d'organiser le débogage et de faire la gestion de leur équipe.

Remarque : Étant donné la nature pratique des méthodes pédagogiques utilisées, il est impératif que vous ne preniez pas de retard. Les concepts appris chaque semaine sont préalables à la semaine suivante.

4. Évaluation des apprentissages

Il y aura trois exercices en équipe (2-3 membres). Le dernier travail est une simple application web basée sur un modèle librement choisi.

Exercice-0 (0 %)

Exercice-1 (20 %)

Exercice-2 (30 %)

Exercice-3 (35 %)

Pour les exercices en équipe (15 %)

Une évaluation d'effort en équipe, où chaque membre d'équipe évalue d'autres membres (avec quelques explications) :

1-Chacun des étudiants doit vous faire parvenir des évaluations un mon courriel avec le sujet et l'entête : SIO-2109 - Evaluation des

coéquipiers

2-Pour chaque membre de votre équipe, et pas pour vous-même, vous devrez mettre une note de 0 à 5 selon les points suivants :

- a. attitude au travail (min 0, 1, 2, 3, 4, 5 max)
- b. qualité du travail (min 0, 1, 2, 3, 4, 5 max)
- c. quantité du travail (min 0, 1, 2, 3, 4, 5 max)
- d. ajouter des commentaires

3-Date limite : 2010-12-09

Une évaluation de 0% à 3% de 15% vous donne un échec dans le cours.

Pondération et barème d'attribution des cotes

La note finale sera transformée en une cote en fonction du barème suivant :

Intervalle	Cote		Intervalle	Cote
[100 – 96]	A +		[70 – 66]	C+
[95 – 91]	A		[65 – 61]	C
[90 – 86]	A-		[60 – 56]	C -
[85 – 81]	B +		[55 – 51]	D +
[80 – 76]	B		[50 – 46]	D
[75 – 71]	B -		[45 – 0]	E (Échec)
Retard motivé	Z		Note retardée	M

Règles disciplinaires

Tout étudiant qui commet une infraction au Règlement disciplinaire à l'intention des étudiants de l'Université Laval dans le cadre du présent cours, notamment en matière de plagiat, est passible des sanctions qui sont prévues dans ce règlement. Il est très important pour tout étudiant de prendre connaissance des articles 28 à 32 du Règlement disciplinaire. Celui-ci peut être consulté à l'adresse suivante : http://www.ulaval.ca/sg/reg/Reglements/Reglement_disciplinaire.pdf .

5. Évaluation du cours

Une évaluation formative du cours sera effectuée. Cette évaluation confidentielle sera uniquement consultée par votre enseignant afin de valider si la formule pédagogique est correcte et si certains ajustements s'imposent avant la fin de la session.

À la fin de ce cours, la Faculté procédera à l'évaluation du cours afin de vérifier si la formule pédagogique a atteint ses buts et si vous êtes satisfait en recueillant vos commentaires et vos suggestions. Cette évaluation est très importante et les responsables du cours vous remercient à l'avance pour votre collaboration. Veuillez noter que l'évaluation est confidentielle.

Correspondance entre le cours et les buts et objectifs du programme de BAA

	Degré d'atteinte dans le cours¹	Méthode d'évaluation utilisée²
1. S'initier à la prise de décision	Amorce	
2. Communiquer	En développement	Exercices en équipe
3. Collaborer et travailler en équipe	Intégration	Exercices en équipe
4. Être ouvert sur le monde	Amorce	
5. Maîtriser les outils technologiques	Intégration	Exercices
6. Être ouvert au changement	Amorce	
7. Avoir un comportement éthique	Amorce	

¹ L'**amorce** veut dire qu'un apprentissage commence à se manifester chez l'étudiant en lien avec les buts ou les objectifs du programme. **En développement** indique des acquis identifiables en relation avec l'apprentissage de ces mêmes buts et objectifs. L'**intégration** indique que l'apprentissage des buts et objectifs est compris et appliqué de manière pertinente à une variété de contextes ou de situations nouvelles. Un cours donné peut couvrir un ou plusieurs buts et objectifs de programme. Aussi, un cours peut amorcer l'apprentissage d'un but et intégrer l'apprentissage d'un autre, selon les habiletés et les compétences visées par le cours.

² Nous faisons référence ici à la méthode d'évaluation utilisée dans le cours pour évaluer le degré d'atteinte du but du programme, par exemple : étude de cas, question ouverte à l'examen, projet, portfolio, quiz, etc. Ces mêmes méthodes d'évaluation pourraient

être utilisées pour collecter des données pour l'Assurance of
Learning.

6. Contenu des séances

Semaine	Date	Livre	Notes	Échéance
1	2010-9-2	Introduction GAE : Chapitres 1, 3	01-Intro 03-Python- Objects	
2	2010-9-9	GAE : Chapitre 2	02-HTML 02-CSS	
3	2010-9-16	GAE : Chapitre 4	La calculatrice Python, 28 - 35	
4	2010-9-23	GAE : Chapitre 5	Le contrôle, 58 - 62 La calculatrice Python, 36 - 44	Exercice-0
5	2010-9-30	GAE : Chapitre 6	La calculatrice Python, 45 - 57	
6	2010-10-07	GAE : Chapitre 7	Le contrôle, 63 - 69 Les fonctions, 70 - 79	
7	2010-10-14	GAE : Chapitre 8	Les fonctions, 80 - 89	Exercice-1
8	2010-10-21	GAE : Chapitre 10	OO, 94 - 104	
9	2010-10-28	<i>Semaine de lecture</i>		
10	2010-11-04	Django : partie 1	OO, 105 - 111	
11	2010-11-11	Django : partie 2	Les fichiers, 90 - 93	Exercice-2
12	2010-11-18	Django : partie 3		
13	2010-11-25	Django : partie 4		
14	2010-12-02	Django de zéro		
15	2010-12-09			Exercice-3

7. Logiciels du cours

Assembla

<http://www.assembla.com/>
http://www.assembla.com/wiki/show/breakoutdocs/Assembla_Video_Tutorials

Cloud Computing

http://en.wikipedia.org/wiki/Cloud_computing

CSS

<http://www.w3schools.com/css/>
<http://www.learn-css-tutorial.com/>
<http://100redesigns.com/>
<http://www.siteduzero.com/tutoriel-3-13666-apprenez-a-creer-votre-site-web.html>
http://fr.wikipedia.org/wiki/Feuilles_de_style_en_cascade
<http://www.cssdebutant.com/>
<http://www.csszengarden.com/tr/francais/>
<http://www.w3qc.org/ressources/traductions/trucs-astuces-css-1/>
<http://www.w3.org/Style/Examples/011/firstcss.fr.html>

Django

[http://en.wikipedia.org/wiki/Django_\(web_framework\)](http://en.wikipedia.org/wiki/Django_(web_framework))
<http://www.djangoproject.com/>
<http://www.djangobook.com/en/2.0/>
<http://code.djangoproject.com/wiki/Tutorials>
<http://docs.djangoproject.com/en/dev/faq/>
<http://code.djangoproject.com/wiki>
<http://www.djangosnippets.org/>
<http://www.django-fr.org/>
<http://docs.django-fr.org/>
<http://www.biologeeek.com/django/>
http://www.webmonkey.com/tutorial/Get_Started_With_Django
<http://djangosearch.com/>

Eclipse

<http://www.eclipse.org/>
<http://eclipse.developpez.com/>
<http://eclipse tutorial.sourceforge.net/>
<http://www.vogella.de/articles/Eclipse/article.html>
<http://eclipse.developpez.com/cours/>

Illuminate

<https://classevirtuelle.ulaval.ca/>
<http://java.com/fr/download/manual.jsp>

<http://www.illuminate.com/support/>
<http://www.learncentral.org/>

Pour utiliser Elluminate, voici la marche à suivre:

- 1- Allez à: <https://classevirtuelle.ulaval.ca/>
- 2- Entrer votre IDUL/NIP afin d'accéder au calendrier
- 3- Sélectionnez la date et cliquez sur le lien qui y apparaîtra.
- 4- Vous devez accepter le téléchargement d'une application Java.
- 5- Des haut-parleurs sont obligatoires et un micro est suggéré si vous désirez poser vos questions de vive voix. Pour en savoir plus sur Elluminate, je vous invite à consulter le site suivant: <http://www.bsp.ulaval.ca/classevirtuelle/aide/> . Ce site vous indique quoi faire avant, pendant et après une séance.

Cours

<http://www.si539.com/>
<http://www.stanford.edu/~ouster/CS349W/lectures/lectureNotes.html>
<http://crypto.stanford.edu/cs142/>

Forms

http://www.cxpartners.co.uk/thoughts/web_forms_design_guidelines_an_eyetracking_study.htm

Google App Engine

<http://code.google.com/appengine/>
<http://code.google.com/appengine/docs/whatisgoogleappengine.html>
<http://code.google.com/appengine/kb/>
<http://appgallery.appspot.com/>
<http://ae-book.appspot.com/>
<http://www.mattakis.com/blog/kisg/20090411/how-to-use-google-app-engine-in-education>
<http://www.gae-en-francais.fr/>
<http://www.gae-en-francais.fr/docs/qu'est-ce-que-google-app-engine.html>
<http://www.biologeeek.com/avenir/web-frameworks/google-app-engine-avantages-et-inconvenients/>

Google Datastore

<http://code.google.com/appengine/docs/python/datastore/>
<http://code.google.com/appengine/articles/datastore/overview.html>
<http://www.ctoedge.com/content/how-google-app-engine-datastore-works>
<http://code.google.com/appengine/articles/modeling.html>
<http://daily.profeth.de/2008/04/er-modeling-with-google-app-engine.html>

<http://labs.google.com/papers/bigtable.html>
<http://blog.xebia.fr/2009/07/16/la-persistence-dans-google-app-engine-partie-une-le-datastore/>

HTML

<http://www.w3schools.com/html/>
<http://www.learn-html-tutorial.com/>
<http://fr.html.net/>
<http://nephi.unice.fr/CoursHTML/>

Mercurial

<http://hginit.com/>
<http://bitbucket.org/>
<http://www.vogella.de/articles/Mercurial/article.html>
http://blog.medallia.com/2007/02/a_guided_tour_of_mercurial.html
http://echoreply.us/tuto/mercurial_site_management.html
<http://bitbucket.org/tortoisehg/stable/wiki/Home>
<http://bitbucket.org/mercurialeclipse/main/wiki/Home>
<http://www.vectrace.com/mercurialeclipse/>
<http://fr.wikipedia.org/wiki/Mercurial>

Python

<http://www.python.org/>
[http://en.wikipedia.org/wiki/Python_\(programming_language\)](http://en.wikipedia.org/wiki/Python_(programming_language))
[http://fr.wikipedia.org/wiki/Python_\(langage\)](http://fr.wikipedia.org/wiki/Python_(langage))
<http://www.python.org/download/>
http://www.annedawson.net/Python_Install_Run.htm
<http://docs.python.org>
<http://pythonfacile.free.fr/>
<http://www.openbookproject.net/>
<http://www.openbookproject.net/pybiblio/>
<http://www.openbookproject.net/pybiblio/gasp/course/>

Sur le site web officiel de Python vous trouverez dans la section « Download » des logiciels d'installation automatique pour les différentes versions de Python. Vous pouvez en confiance choisir la dernière version « de production ».

Par exemple, au 2009-12-29, il s'agit de la version 2.6.4 - Fichier à télécharger : python-2.6.4.msi . Copiez ce fichier dans un répertoire temporaire de votre machine et exécutez-le. Python s'installera par défaut dans un répertoire nommé « Python** » (** indiquant les deux premiers chiffres du n° de la version), et des icônes de lancement seront mises en place automatiquement.

Lorsque l'installation est terminée, vous pouvez effacer le contenu du répertoire temporaire.

XHTML

<http://www.w3schools.com/Xhtml/>
<http://www.la-grange.net/w3c/xhtml1/>
<http://www.tbs-sct.gc.ca/clf2-nsi2/tb-bo/td-dt/mxhtmls-fra.asp>
http://fr.wikipedia.org/wiki/Extensible_HyperText_Markup_Language
http://openweb.eu.org/articles/xhtml1_une_heure
http://doc.ubuntu-fr.org/projets/ecole/web/initiation_xhtml

Web Design

<http://news.ycombinator.com/item?id=1604915>
<http://credibility.stanford.edu/guidelines/index.html>

World Wide Web Consortium (W3C)

<http://www.w3.org/>