



Copyright © 2015 [Yves MARCOUX](#); dernière modification de cette page: 2015-01-11
Cette création est mise à disposition sous un [contrat Creative Commons](#)

SCI6373 Programmation documentaire

Plan de cours Hiver 2015

[Yves MARCOUX](#) - [EBSI](#) - [Université de Montréal](#)

Informations générales

- Cours de trois crédits, ce qui correspond à 135 heures de travail pour l'étudiant.
- [Description officielle](#) à l'[Annuaire général de l'Université](#):

Introduction à la programmation procédurale et orientée-objet. Développement d'applications documentaires dans un langage de scriptage orienté-objet. Aperçu d'autres langages. Notions d'algorithmique et de complexité. Préalables: SCI6051 et SCI6052.

- **Horaire et locaux:** Mardi de 13h à 15h50 au C2031 du [Pavillon Lionel-Groulx](#) (un des [Laboratoires d'informatique documentaire de l'EBSI](#); entrer par le C2035). *Certains travaux de laboratoire s'effectuent à l'intérieur de cette même plage horaire et au même endroit.*
 - **Professeur:** [Yves MARCOUX](#) <y marcoux@gmail.com>.
 - **Site du cours:** <<http://cours.ebsi.umontreal.ca/SCI6373/>>.
-

Objectifs

À la fin du cours, l'étudiant(e) possèdera les rudiments de la programmation procédurale et orientée-objet moderne. Plus spécifiquement, l'étudiant(e):

- Comprendra le concept de scriptage en général et dans le contexte du web en particulier.
 - Comprendra l'articulation client-serveur(s) du point de vue programmatique.
 - Pourra développer à partir de zéro des scripts pour des traitements simples.
 - Pourra adapter des scripts complexes existants.
-

Description

Le cours est une introduction à la programmation dans les environnements de scriptage. Le principal langage utilisé est le JavaScript. Un projet de fin de session permet d'intégrer les

connaissances acquises. Les étudiants sont appelés à explorer un autre langage ou environnement de programmation (de type scriptage ou non).

L'accent est mis sur le scriptage de pages web pour exécution en navigateur côté client, mais l'exécution côté serveur sera aussi touchée (si possible avec accès à une base de données), de même que le scriptage au niveau du système d'exploitation pour la manipulation de fichiers en lot. Une connaissance de base de HTML et de CSS est présupposée.

Le langage ou l'environnement choisi pour exploration et la nature du projet de fin de session doivent être approuvés par le professeur.

Plusieurs exercices sont donnés au cours de la session, en plus de devoirs évalués. Il est essentiel de faire *tous* les exercices proposés, et non seulement les devoirs évalués.

Calendrier des activités

Notes:

- *L'heure limite de remise de certains travaux est fixée au **jeudi à 23h55**. Comparativement à une remise en début de cours, cette formule (à l'essai cette année) offre la chance de consulter le professeur une fois de plus avant une remise.*
- *Les contenus, incluant le travail supervisé en laboratoire, et leur répartition entre les cours seront constamment ajustés au cours de la session.*

Calendrier des activités

Date – Cours	Contenu et lectures	Évaluations
2015-01-13 – C1	Bienvenue. Concepts de scriptage et de « programmation documentaire ». Notions d'environnement et d'objet. Les trois environnements touchés dans le cours. « End-user programming » et « end-user development ». Configuration des environnements: navigateurs et éditeur de scripts. JavaScript en ligne adresse du navigateur (pseudo-protocole « javascript: »).	
2015-01-20 – C2	Lectures préalables: Goodman, chap. 1 et 2; lectures obligatoires de Krill et Manfre. Notion d'expression évaluée. Notation « . » (point) pour les objets et leurs membres (propriétés et méthodes). Scripts. Commentaires. Concaténation et autres opérations. Entrées-sorties de base. Variables. Rappels HTML. Dynamisme dans une page HTML: modifications, ajouts, retraits, stylage.	
2015-01-27 – C3	Lectures préalables: Goodman, chap. 7, 8, 10 et 12 Quiz 1 (en classe).	13h : Quiz 1 (10%) Jeudi 2015-01-

	Types de données de base: chaînes de caractères, entiers, réels, dates; opérations associées. Comparaisons. Conversions.	29 23h55 : Remise (sur votre espace web gin-ebisi + courriel au professeur) du travail de Validation des prérequis (5%)
2015-02-03 – C4	Lectures préalables: Goodman, chap. 9 et 11 Traitement conditionnel (IF) et structures de contrôle. Notions de procédure et d'algorithme. Organigramme.	Jeudi 2015-02-05 23h55 : Devoir 0 à rendre sur StudiUM (5%)
2015-02-10 – C5	Objets. Tableaux. Boucles. Structures de données complexes. Tri. Fonctions définies par l'utilisateur.	
2015-02-17 – C6	Travail avec Notepad++, débogage avec Firebug	Jeudi 2015-02-19 23h55 : Devoir 1 à rendre sur StudiUM (10%)
2015-02-24 – C7	Quiz 2 (en classe). Traitements côté serveur et accès à une base de données.	13h : Quiz 2 (10%)
2015-03-03	<i>Semaine de lecture (pas de cours)</i>	
2015-03-10 – C8	Notions d'API et de bibliothèque logicielle. JQuery.	
2015-03-17 – C9	Scriptage au niveau du système d'exploitation, accès aux fichiers locaux.	
2015-03-24 – C10	Analyse versus programmation. Compilation versus interprétation.	Jeudi 2015-03-26 23h55 : Devoir 2 à rendre sur StudiUM (10%)
2015-03-31 – C11	<i>Évaluation du cours.</i> Méthodologies: découpage, modularité, conventions de nommage, documentation. <i>Design patterns</i> .	
2015-04-07	<i>Pas de cours</i>	Jeudi 2015-04-09 23h55 : Devoir 3 à rendre sur StudiUM (15%)
2015-04-14 – C12	Lectures préalables: Les (3) articles de Morgan, Colford et Breeding; les (2) articles de Wikipedia AJAX (Asynchronous JavaScript and XML). Principe <i>Model-view-controller</i> (MVC). Logiciel libre et bibliothèques. SaaS (Software as a Service). Modèles de calcul, machines de Turing, concepts	

	d'algorithmique et de complexité.	
2015-04-21 – C13	Présentations (en classe) des langages ou environnements explorés par les étudiants.	13h : Présentation en classe Judi 2015-04-23 23h55 : Matériel de présentation à rendre sur StudiUM (présentation + matériel : 15%)
2015-04-28 – C14	Travail en laboratoire sur le projet de fin de session.	23h55 : Projet de fin de session à rendre sur StudiUM (20%)

Évaluation

Activité	Pondération
Travail individuel de validation des prérequis	5%
Deux quiz individuels à livres fermés en classe (2 x 10%)	20%
Quatre devoirs (5% + 10% + 10% + 15%)	40%
Projet de fin de session	20%
Exploration et présentation en classe d'un langage ou environnement de programmation	15%

Les activités évaluées se font en équipe de deux, sauf le travail de validation des prérequis et les quiz. Les travaux de validation des prérequis et les devoirs sont décrits dans des protocoles rendus disponibles en temps opportun.

Il n'y a pas de rapport à rendre pour le travail d'exploration d'un langage ou environnement de programmation, mais vous devez remettre sur StudiUM une copie des diapositives et autre matériel utilisé pour la présentation.

La méthode de calcul de la note finale du cours est présentée au <http://marcoux.ebsi.umontreal.ca/enseign/calculs.htm>.

Règlements, politiques

Remise des travaux à évaluer

Les artefacts à évaluer (fichiers, rapports, etc.) doivent être remis **en format numérique**. Sauf exception (mentionnée le cas échéant dans le protocole du travail) ils doivent être déposés à

l'emplacement prévu sur [StudiUM](#) en un seul fichier (compressé si nécessaire), au plus tard à l'heure prévue pour la remise. Ce fichier unique doit être conforme aux directives données dans le protocole du travail concerné; en particulier, il doit être **nommé** conformément à ces directives.

Si la remise comporte un rapport (le cas échéant, cela est spécifié dans le protocole du travail), ce rapport doit être en format PDF, OpenOffice ou Word, et doit être formaté pour du papier 8,5 x 11 po. **Les rapports doivent impérativement être structurés en sections numérotées et paginés.**

Exceptionnellement, après entente avec le professeur, certains artefacts peuvent être remis en format papier.

Si vous ne respectez pas les directives ci-dessus, vous êtes considéré comme n'ayant pas remis le travail.

Vous devez *impérativement* conserver une copie de sécurité intégrale des artefacts remis, *au moins jusqu'à réception de leur correction*. Vous devez être prêt à transmettre rapidement sur demande au correcteur (par courriel ou autrement) une copie du matériel remis originellement, en cas de problème de lecture ou autre; c'est votre responsabilité.

Modes de communication

Le mode de communication privilégié du professeur vers les étudiant(e)s est la section [Nouvelles concernant le cours](#) de la [page d'accueil du cours](#). Vous pouvez, si désiré, vous abonner à ces nouvelles par le truchement du [fil RSS associé](#); sinon, vous devez consulter la section *Nouvelles* très régulièrement (au moins une fois par jour).

Le professeur peut également utiliser *le courriel* pour joindre les étudiant(e)s. L'adresse utilisée pour vous joindre sera celle enregistrée dans votre [profil DGTIC](#); assurez-vous que cette adresse soit valide et fonctionnelle en tout temps (voyez notamment à ne pas laisser votre boîte aux lettres se remplir). Vous devez lire très régulièrement votre courriel (au moins une fois par jour).

Le mode de communication privilégié des étudiant(e)s vers le professeur est le courriel. L'adresse courriel du professeur est <ymarcoux@gmail.com>. **SVP, toujours inscrire la mention [SCI6373] (incluant les crochets) au début de la ligne sujet de votre message.**

Autres règlements et politiques

Tous les règlements, politiques et directives énoncés dans le [Guide de l'étudiant de la MSI](#) s'appliquent. Une attention particulière est à porter aux éléments suivants :

Travaux en équipe

Pour les travaux réalisés en équipe, le professeur se réserve le droit d'évaluer séparément chaque membre d'une équipe.

Code d'honneur

Il est attendu que les étudiant(e)s respectent le [code d'honneur de l'EBSI](#).

Règlement disciplinaire sur le plagiat ou la fraude concernant les étudiants

Toute infraction au règlement sur le plagiat ou la fraude sera traitée suivant la procédure indiquée dans le règlement.

Retard dans la remise des travaux

Tout retard non justifié dans la remise d'un travail sera sanctionné : 5% de la note est retranché par jour de calendrier de retard jusqu'à un maximum de 35%; à la 8^{ième} journée de calendrier, la note F ou zéro (0) est attribuée.

Qualité de la langue

Un maximum de 10% de la note globale d'un travail pourra être retranché pour mauvaise qualité de la langue dans les travaux (ne s'applique pas aux examens ou quiz en classe).

Bibliographie

Lectures obligatoires

- Breeding, Marshall. *ASIS&T Bulletin*, December 2008/January 2009. "The Viability of Open Source ILS." En ligne <http://www.asis.org/Bulletin/Dec-08/DecJan09_Breeding.html>.
- Colford, Scot. *ASIS&T Bulletin*, December 2008/January 2009 "Explaining Free and Open Source Software." En ligne <http://www.asis.org/Bulletin/Dec-08/DecJan09_Colford.html>.
- Goodman, Danny; et al. *JavaScript Bible Seventh edition*. Indianapolis, Ind. : Wiley Pub., c2010, env. 2000 pp. En ligne (de umontreal.ca) <<http://atrium.umontreal.ca/notice/UM-ALEPH001488696>>. [Sur books24x7](#). [Exemplaire papier de la 5e édition c2004](#) en réserve à la BLSH (QA 76.73 J39 G66 2004).
- Krill, Paul. "JavaScript creator ponders past, future." *InfoWorld*, 2008-06-23. En ligne <<http://www.infoworld.com/article/2653798/application-development/javascript-creator-ponders-past--future.html>>.
- Manfre, Charles. "Why JavaScript is the Future of Programming." *sitepoint - JavaScript*, 2012-11-30. En ligne <<http://www.sitepoint.com/why-javascript-is-the-future-of-programming/>>.
- Morgan, Eric Lease. "Open Source Software in Libraries." *ASIS&T Bulletin*, December 2008/January 2009. En ligne <http://www.asis.org/Bulletin/Dec-08/DecJan09_Morgan.html>.
- Wikipedia. "Ajax (programming)." En ligne <<http://en.wikipedia.org/wiki/AJAX>>.
- Wikipedia. "Model-view-controller." En ligne <<http://en.wikipedia.org/wiki/Model-view-controller>>.

Ressources recommandées

- Babcock, Charles. "Ajax 101: From Toolkits To Strategy, How Companies Can Put It To Use." *InformationWeek*, 2006-06-12. En ligne <<http://www.informationweek.com/ajax-101-from-toolkits-to-strategy-how-companies-can-put-it-to-use/d/d-id/1044053>>.
- Cagle, Kurt. "Why JavaScript is the Future of Enterprise Computing." *Avalon Consulting, LLC - Blogs*, 2013-02-18. En ligne <<http://blogs.avalonconsult.com/blog/generic/why-javascript-is-the-future-of-enterprise-computing/>>.
- Chandra, Vikram. *Geek Sublime : Une vision esthétique, littéraire, mathématique et pleine*

- d'autodérision du codage*. Paris : Robert Laffont, 2014, 332 pp. Version anglaise *Geek Sublime: The Beauty of Code, the Code of Beauty* [en commande à la BLSH](#).
- Ecma International. *ECMAScript® Language Specification*, Standard ECMA-262, Edition 5.1, June 2011. En ligne (PDF et HTML) <<http://www.ecma-international.org/publications/standards/Ecma-262.htm>>.
 - Festa, Paul. “Standards Activists Target Scripts.” *CNET News* 2005-07-18. En ligne <http://news.cnet.com/Standards-activists-target-scripts/2100-1032_3-5793767.html>. Résumé en ligne <<http://technews.acm.org/archives.cfm?fo=2005-07-jul/jul-22-2005.html&hdr=2#item14>>.
 - Festa, Paul. “Will Ajax help Google clean up?” *CNET News*, 2005-05-17. En ligne <http://news.cnet.com/Web-tools-blaze-trail-to-the-past/2100-1032_3-5621010.html>.
 - Kaneshige, Tom. “A future without programming.” *InfoWorld*, 2008-11-20. En ligne <<http://www.infoworld.com/article/2654154/application-development/a-future-without-programming.html>>.
 - Krill, Paul. “Scripting Languages Spark New Programming Era.” *InfoWorld*, 2008-06-23. Résumé en ligne <<http://technews.acm.org/archives.cfm?fo=2008-06-jun/jun-25-2008.html&hdr=1#367980>>. Article complet au <<http://www.infoworld.com/article/2651797/application-development/scripting-languages-spark-new-programming-era.html>>.
 - Kumar, Ritesh. “On Scripting Languages” *OSnews*, 2004-05-13. En ligne <<http://www.osnews.com/story/7038>>.
 - Mozilla Developer Network. « Référence JavaScript. » En ligne <<https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference>>.
 - Microsoft Corp. “Windows Script Host.” En ligne <[http://msdn.microsoft.com/en-us/library/9bbdkx3k\(v=vs.84\).aspx](http://msdn.microsoft.com/en-us/library/9bbdkx3k(v=vs.84).aspx)>.
 - Roque, Celine. “What Will It Take to Be a Web Worker in 2009?” *WebWorkerDaily*, 2008-12-22. En ligne <<http://webworkerdaily.com/2008/12/22/what-will-it-take-to-be-a-web-worker-in-2009/>>.
 - Savill, John. “How do I run a windows script from the command line?” *WindowsITPro*. En ligne <<http://windowsitpro.com/article/articleid/15226/how-do-i-run-a-windows-script-from-the-command-line.html>>.
 - Stavelly, Allan M. *Writing in software development*. New Mexico Tech Press, 2011. BLSH [QA 76.76 D63 S73 2011](#).
 - Tommasi, Marc. *Le robot*. Petit logiciel pour s’initier à la pensée algorithmique. En ligne <<http://www.grappa.univ-lille3.fr/ftp/Softs/Robot/>>. Choisissez l’ancienne génération. Nécessite un décompresseur .tar, par exemple WinZip ou 7-Zip. Il faut décompresser le fichier deux fois. Il se peut que vous obteniez une erreur à la seconde décompression; elle n’affecte ni l’installation ni le fonctionnement du logiciel. Après décompressions, exécutez le fichier setup.exe.
 - W3Schools. *JavaScript Tutorial*. En ligne <<http://www.w3schools.com/js/>>.
 - Wikipedia. “Software as a service.” En ligne <http://en.wikipedia.org/wiki/Software_as_a_service>.
 - Wright, Nolan. “Why JavaScript Will Become The Dominant Programming Language Of The Enterprise.” *readwrite*, 2013-08-09. En ligne <<http://readwrite.com/2013/08/09/why-javascript-will-become-the-dominant-programming-language-of-the-enterprise>>.

