Historique des mises à jour :
2018-01-09: Mise en ligne initiale.
2018-01-16: Ajout d'une lecture préalable pour le Cours 3.
2018-01-24 Correction du jour de remise du Devoir 0 : le 2 février est un **vendredi**.

SCI6373 Programmation documentaire

Plan de cours Hiver 2018

Yves MARCOUX - EBSI - Université de Montréal

Table des matières

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 – FAS:**

  > 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 C-2027 du Pavillon Lionel-Groulx. *Certains travaux de laboratoire s'effectuent au même endroit et à l'intérieur de la même plage horaire.*.
- **Professeur:** Yves MARCOUX <ymarcoux@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 aussi 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.

Plusieurs exercices sont donnés au cours de la session, en plus de devoirs évalués. Il est **essentiel** de faire les exercices proposés au fur et à mesure de façon à ne pas « décrocher ».

Le langage ou environnement choisi pour exploration doit être approuvé par le professeur suivant un protocole qui sera présenté en classe. Le projet de fin de session est un devoir évalué d'un peu plus grande envergure que les autres.

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 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
2018-01-09 – 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: »).	
2018-01-16 – C2	**Lectures préalables:** Goodman, chap. 1 et 8; lectures obligatoires de Kolawole et de Manfre. Types de données de base: booléen, chaîne de caractères, nombre, pointeur à un objet. Quelques opérations: arithmétiques, concaténation, égalité, expressions conditionnelles. Notions d'expression évaluée, d'instruction et d'énoncé. Notation « . »	**Jeudi 2018-01-18 23h55 :** Remise du travail individuel de Validation des prérequis (5%) (fichiers à déposer sur votre site web gin-ebsi et sur StudiUM)

	(point) pour les objets et leurs membres (propriétés et méthodes). Fonctions. Arbres d'exécution.	
2018-01-23 – C3	**Lectures préalables:** Goodman, chap. 9, 10, 15 et 16; Un peu de terminologie. Variables, assignations. Scripts. Commentaires. Entrées-sorties de base. Construction d'un script, méthode de travail. Notion d'« approche algorithmique ». Présentation du Devoir 0.	
2018-01-30 – C4	**Lectures préalables:** Goodman, chap. 18, 21 (jusqu'à p. 396) et 22. Quiz 1 (en classe). Conversions, opérations sur les chaînes de caractères. Rappels HTML et CSS. Dynamisme dans une page HTML par stylage. Entrées-sorties par formulaire. Fonctions définies par l'usager.	**13h :** Quiz 1 (10%) **Vendredi 2018-02-02 23h55 :** Devoir 0 à rendre sur StudiUM (5%)
2018-02-06 – C5	Traitement conditionnel (IF) et structures de contrôle. Organigrammes. Tableaux. Boucles. Environnements de développement : aides à la programmation. JSEclipse. Présentation du Devoir 1.	
2018-02-13 – C6	Débogage avec les outils de développement en navigateurs.	**Jeudi 2018-02-15 23h55 :** Devoir 1 à rendre sur StudiUM (10%)
2018-02-20 – C7	Propriété `innerHTML`. Approche algorithmique et exercices en vue du Devoir 2. Présentation du Devoir 2.	
2018-02-27 – C8	Quiz 2 (en classe). Comparaisons de chaînes et tri d'un tableau. Approche algorithmique en vue du Devoir 2 (suite).	**13h :** Quiz 2 (10%)
2018-03-06	*Semaine de lecture (pas de cours)*	
2018-03-13 – C9	Structures de données complexes. Objets arbitraires. JSON. La fonction `document.write` revisitée; fenêtres surgissantes (*pop-up*). Dates. Désaccentuation.	
2018-03-20 – C10	Approche algorithmique (début) et exercices en vue du Devoir 3. Présentation du Devoir 3.	**Jeudi 2018-03-22 23h55 :** Devoir 2 à rendre sur StudiUM (10%)
2018-03-27 – C11	Approche algorithmique pour le Devoir 3 (suite). Présentation du projet de fin de session.	

2018-04-03 – C12	Conférencier invité : Patrick Beaulieu (Raymond Chabot Grant Thornton), diplômé de la MSI. Approche algorithmique pour le projet de fin de session. **Lectures préalables:** Les (3) articles de Morgan, Colford et Breeding; les (2) articles de Wikipedia.	**Jeudi 2018-04-05 23h55 :** Devoir 3 à rendre sur StudiUM (15%)
2018-04-10 – C13	Approche algorithmique pour le projet de fin de session (suite). Si le temps le permet, choix parmi les thèmes suivants : • Traitements côté serveur et accès à une base de données. • Scriptage au niveau du système d'exploitation, accès aux fichiers locaux. • AJAX (Asynchronous JavaScript and XML). • Principe *Model-view-controller* (MVC). • Logiciel libre. • SaaS (Software as a Service). • Modèles de calcul, machines de Turing, concepts d'algorithmique et de complexité. • API, bibliothèques logicielles, « frameworks » JavaScript : JQuery, AngularJS, etc. • Compilation versus interprétation. • Analyse versus programmation. • Méthodologies: découpage, modularité, documentation. *Design patterns*. • Intelligence artificielle, réseaux de neurones artificiels, apprentissage profond. • Informatique et cryptographie quantiques. • *Bitcoin*, *Blockchains* et grands livres distribués (*distributed ledgers*).	
2018-04-17 – C14	Présentations (en classe) des langages ou environnements explorés par les étudiants. Si le temps le permet, choix parmi les thèmes mentionnés sous C13.	**13h :** Présentation en classe (Prestation de la présentation + matériel déposé sur StudiUM : 15%)
Lundi 2018-04-30 (dernier jour de la session)		**Lundi 2018-04-30 23h55 :** Matériel de présentation à déposer sur StudiUM **Lundi 2018-04-30 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: prestation et matériel présenté	15%

Toutes 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 de tout autre matériel utilisé pour la présentation.

N.B. : Vous n'avez pas à faire le travail de *Validation des prérequis* si vous avec déjà suivi, ou suivez de façon concomitante, le cours *INU3011 Documents structurés* : le travail analogue dans INU3011 comptera pour les deux cours.

Les directives données dans les différents protocoles de travaux doivent être suivies et, pour les travaux évalués, leur respect est un critère d'évaluation.

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 TI; 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 8ième 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 sur MyLibrary (de `umontreal.ca`) <http://atrium.umontreal.ca/UM:UM-ALEPH002244488>. Exemplaire papier de la 5e édition en réserve à la BLSH (QA 76.73 J39 G66 2004) <http://atrium.umontreal.ca/UM:UM-ALEPH000383396>.
- Kolawole, Emi. "Tim Berners-Lee on the making of new worlds." *The Washington Post (Blogs) – Innovations*, 2013-03-11. En ligne <https://www.washingtonpost.com/blogs/innovations/post/tim-berners-

lee-on-the-making-of-new-worlds/2013/03/11/3d34f08c-89e9-11e2-98d9-3012c1cd8d1e_blog.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/>.
- Marcoux, Yves. « Un peu de terminologie ». En ligne <http://cours.ebsi.umontreal.ca/SCI6373/H2018/matthem/termino.html>.
- 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 utiles

- 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>.
- bento : Teach yourself to code. Site recensant un grand nombre de tutoriels pour apprendre à programmer dans différents langages. D’intérêt particulier est la grille <https://www.bento.io/grid>, dans laquelle les tutoriels sont regroupés par langage et par type de technologie. En ligne <https://www.bento.io/>.
- 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 réserve à la BLSH (PN 56 T37 C36 2014) <http://atrium.umontreal.ca/UM:UM-ALEPH002208262>.
- Code Studio. Site pour apprendre aux jeunes à programmer dans différents environnements attrayants. En ligne <https://code.org/>.
- Ecma International. *ECMAScript® 2017 Language Specification*, Standard ECMA-262, 8th Edition, June 2017. 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://www.cnet.com/news/standards-activists-target-scripts/>. 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://www.cnet.com/news/will-ajax-help-google-clean-up/>.
- 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. “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>.
- 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>.
- Morgan, Nick. *JavaScript for Kids: A Playful Introduction to Programming.* Los Angeles (É.-U.) : No Starch Press, 2015, 305 pp.
- 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>.

- Roque, Celine. “What Will It Take to Be a Web Worker in 2009?” *GIGAOM Research*, 2008-12-22. En ligne <https://gigaom.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://www.itprotoday.com/management-mobility/how-do-i-run-windows-script-command-line>.
- Scratch. Langage de programmation graphique pour apprendre aux jeunes à programmer dans un environnement attrayant. En ligne <https://scratch.mit.edu/>.
- Stavely, Allan M. *Writing in software development.* New Mexico Tech Press, 2011. En réserve à la BLSH (QA 76.76 D63 S73 2011) <http://atrium.umontreal.ca/UM:UM-ALEPH002013882>.
- Tommasi, Marc. *Le robot.* Petit logiciel en français pour s’initier à la pensée algorithmique. En ligne <http://www.grappa.univ-lille3.fr/ftp/Softs/Robot/>.
- 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>.