

SCI6373 Programmation documentaire

Plan de cours Hiver 2021

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

Table des matières

1. [Informations générales](#)
 2. [Objectifs](#)
 3. [Description](#)
 4. [Évaluation](#)
 5. [Calendrier des activités](#)
 6. [Règlements, politiques](#)
 - 6.1 [Modes de communication](#)
 - 6.2 [Autres règlements et politiques](#)
 7. [Bibliographie](#)
 - 7.1 [Lectures obligatoires \(par ordre de lecture dans le cours\)](#)
 - 7.2 [Ressources utiles et/ou amusantes](#)
 - 7.3 [Références diverses](#)
-

1. Informations générales

- Cours de trois crédits, ce qui correspond à 135 heures de travail pour l'étudiante¹.
- [Description officielle du cours](#) :

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: (SCI6001 ou SCI6002) et SCI6005.

- **Professeur:** [Yves MARCOUX](#) <ymarcoux@gmail.com>.
- **Site du cours:** <<https://cours.ebsi.umontreal.ca/SCI6373/>>.

Modalités :

- La **première rencontre** aura lieu sur Zoom le **jeudi 14 janvier 2021 à 13h**. Le lien sera mis sur la [page StudiUM du cours](#) quelques minutes avant (vous devrez vous identifier avec votre [Code d'accès et UNIP](#) de l'UdeM et être inscrite au cours pour y avoir accès).
- De façon générale, il y aura une **courte partie « magistrale » synchrone** à chaque jeudi 13h, suivie de la disponibilité du professeur

jusqu'à 15h50.

- **Toutes les autres activités sont en asynchrone**, y compris les quiz, qui se feront sur StudiUM.
- **Les parties magistrales synchrones seront toutes enregistrées** et pourront être (re)visionnées en différé.

¹: Vous remarquerez que, dans ce cours, la question des épécènes est habituellement réglée par l'utilisation du féminin.

2. Objectifs

À la fin du cours, l'étudiante possédera les rudiments de la programmation procédurale et orientée-objet moderne. Plus spécifiquement, l'étudiante:

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

3. Description

Le cours est une introduction à la programmation dans les environnements de scriptage. Le principal langage utilisé est le JavaScript. Une progression de travaux pratiques permet d'intégrer les connaissances acquises. Les étudiantes sont aussi appelées 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. Une connaissance de base de HTML et de CSS est présumée.

Plusieurs exercices sont donnés au cours de la session, en plus de travaux pratiques. 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. Ce peut être une occasion d'explorer dans des projets amusants les ressources programmables des laboratoires d'informatique documentaire de l'EBSI, comme le micro:bit, le Raspberry Pi et le Arduino.

4. Évaluation

Activité	Pondération
Trois « mini-quiz » sur StudiUM (3 x 10%)*	30%
TP 1 : Rallye d'accueil*	5%
TP 2-6 : Projets de programmation (10% + 10% + 10% + 10% + 10%)	50%
Exploration et présentation devant la classe d'un langage ou environnement	15%

*Les évaluations marquées d'un astérisque sont individuelles. Les autres sont réalisées en équipes de deux personnes, mais celles qui le souhaitent peuvent aussi travailler individuellement (SVP, en avisant le professeur le plus tôt possible).

N.B. : Pour réduire le stress des quiz en temps de pandémie, les « mini-quiz » sont des quiz sur StudiUM dans lesquels le simple fait de répondre à toutes les questions donne 40% des points. Toute bonne réponse fait augmenter la note obtenue.

Il n'y a pas de rapport à rendre pour le travail d'exploration d'un langage ou environnement. L'évaluation est basée sur la prestation de la présentation et sur le matériel présenté, que vous devez déposer dans StudiUM.

Le respect des directives données dans les protocoles de travaux est un critère d'évaluation.

La méthode de calcul de la note finale du cours est présentée au <https://marcoux.ebsi.umontreal.ca/enseign/calculs-NNGv2.html>.

Pour vous aider à suivre l'évolution des résultats de vos évaluations au cours du trimestre, et notamment connaître les notes requises dans les évaluations à venir pour atteindre une cote donnée, un [Évaluateur de réussite](#) est disponible. N'hésitez pas à l'utiliser.

5. Calendrier des activités

Notes : Les contenus et leur répartition entre les cours seront ajustés en cours de session pour tenir compte de la progression réelle.

Calendrier des activités

Date – Cours	Contenu	Évaluations
2021-01-14 – C1	Logistique/ressources du cours / Prérequis / Qu'est-ce que la programmation / Pourquoi programmer ? / Programmation et société / <i>End-user programming</i> / Programmation et écriture/esthétique / Ce cours et ses choix / Votre premier script !	
2021-01-21 – C2	Arbres d'exécution.	✘ Rallye d'accueil Travail individuel (TP 1) 5%. Remise : vendredi 22 janvier 23h55.
2021-01-28 – C3	Entrées-sorties de base. Enchaînement d'instructions, construction d'un script, méthode de travail, utilisation des commentaires. Notion d'« approche algorithmique ».	🔗 Mini-quiz 1 (10%). À faire pendant le vendredi 29 janvier.
2021-02-04 – C4	Conversions, opérations sur les chaînes de caractères. Rappels HTML et CSS. Dynamisme dans une page HTML par stylage. Entrées-sorties par formulaire.	✘ TP 2 (10%). Remise : vendredi 5 février 23h55.
2021-02-11 – C5	Traitement conditionnel (IF) et structures de contrôle. Organigrammes.	
2021-02-18 – C6	Débogage avec les outils de développement en navigateurs.	✘ TP 3 (10%). Remise : vendredi 19 février 23h55.

2021-02-25 – C7	Fonctions définies par le programmeur. Dynamisme par propriétés CSS.	🔗 Mini-quiz 2 (10%). À faire pendant le vendredi 26 février.
2021-03-04	<i>Semaine de lecture — Pas de cours</i>	
2021-03-11 – C8	Tableaux.	
2021-03-18 – C9	Comparaisons de chaînes et tri d'un tableau. Propriété innerHTML.	✂ TP 4 (10%). Remise : vendredi 19 mars 23h55.
2021-03-25 – C10	La fonction document.write; fenêtres surgissantes (<i>pop-up</i>). Dates. Désaccentuation. Traitement des textes et des mots.	🔗 Mini-quiz 3 (10%). À faire pendant le vendredi 25 mars.
2021-04-01 – C11	Traitements côté serveur et accès à une base de données. Scriptage au niveau du système d'exploitation, accès aux fichiers locaux.	✂ TP 5 (10%). Remise : vendredi 2 avril 23h55.
2021-04-08 – C12	Conférencière invitée (à confirmer) / Processus et modèles de développement informatique, méthodologies. <i>Design Patterns</i> . Logiciel libre.	
2021-04-15 – C13	Si le temps le permet, choix parmi les thèmes suivants : <ul style="list-style-type: none"> • AJAX (Asynchronous JavaScript and XML). • Principe <i>Model-view-controller</i> (MVC). • 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. • Intelligence artificielle, réseaux de neurones artificiels, apprentissage profond. • Informatique et cryptographie quantiques. • <i>Bitcoin</i>, <i>Blockchains</i> et grands livres distribués (<i>distributed ledgers</i>). 	✂ TP 6 (10%). Remise : vendredi 16 avril 23h55.
2021-04-22 – C14	Exploration d'un langage ou environnement : présentations. Si le temps le permet, choix parmi les thèmes mentionnés sous C13.	Prestation de la présentation pendant la semaine (modalités à déterminer)
2021-04-29 – C15	Retour sur le TP 6.	
2021-04-30 (dernier jour de la session)		Matériel de présentation (+ prestation) 15%. Remise : vendredi 30 avril 23h55.

6. Règlements, politiques

6.1 Modes de communication

Le mode de communication privilégié du professeur vers les étudiantes est la section *Nouvelles* de la [page StudiUM du cours](#).

Le professeur peut également utiliser *le courriel* pour joindre les étudiantes. 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 votre courriel à tous les jours.

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

Il est également possible de poser des questions dans le forum de discussion de la [page StudiUM du cours](#).

6.2 Autres règlements et politiques

Tous les règlements, politiques et directives énoncés dans le [Guide étudiant de la MSI](#) s'appliquent, incluant le *Code d'honneur de l'EBSI*. 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.

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 quiz).

7. Bibliographie

7.1 Lectures obligatoires (par ordre de lecture dans le cours)

- Marcoux, Yves. « Introduction aux expressions JavaScript ». En ligne <<https://cours.ebsi.umontreal.ca/SCI6373/H2021/matthem/introExpJS.html>>.
- Goodman, Danny; et al. *JavaScript Bible Seventh edition*. Indianapolis, Ind. : Wiley Pub., c2010, env. 2000 pp. [En ligne sur ProQuest Ebook Central](#) (de umontreal.ca) <<https://umontreal.on.worldcat.org/oclc/682621164>>. Exemplaire papier de la 5^e édition en réserve à la BLSH (QA 76.73 J39 G66 2004) <<https://umontreal.on.worldcat.org/oclc/424114217>>.
- Kolawole, Emi. "Tim Berners-Lee on the making of new worlds." *The Washington Post (Blogs) – Innovations*, 2013-03-11. En ligne <<https://tinyurl.com/ygbnb9kj>>.
- Breeding, Marshall. *ASIS&T Bulletin*, December 2008/January 2009. "The Viability of Open Source ILS." En ligne <<https://asistdl.onlinelibrary.wiley.com/doi/full/10.1002/bult.2008.1720350207>>.
- Colford, Scot. *ASIS&T Bulletin*, December 2008/January 2009 "Explaining Free and Open Source Software." En ligne <<https://asistdl.onlinelibrary.wiley.com/doi/10.1002/bult.2008.1720350205>>.

- Morgan, Eric Lease. “Open Source Software in Libraries.” *ASIS&T Bulletin*, December 2008/January 2009. En ligne <<https://asistdl.onlinelibrary.wiley.com/doi/10.1002/bult.2008.1720350204>>.

7.2 Ressources utiles et/ou amusantes

- 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/topics>>, dans laquelle les tutoriels sont regroupés par langage et par type de technologie. En ligne <<https://www.bento.io/>>.
- Broussard, Meredith. *Artificial unintelligence : how computers misunderstand the world*. Cambridge, Massachusetts : The MIT Press, 2018. En réserve à la BLSH (QA 76.9 C66 B76 2018).
- 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) <<https://umontreal.on.worldcat.org/oclc/900523093>>.
- Classements de langages de programmation selon différents critères. [TIOBE](#), [UBUNTUPIT](#), [PYPL](#), etc.
- Code Studio. Site pour apprendre aux jeunes à programmer dans différents environnements attrayants. En ligne <<https://code.org/>>.
- Freeman, Jason; Brian Magerko; Doug Edwards; Tom Mcklin; Taneisha Lee; Roxanne Moore. 2019. EarSketch: engaging broad populations in computing through music. *Commun. ACM* 62, 9 (August 2019), 78–85. En ligne <<https://doi.org/10.1145/3333613>>.
- Krill, Paul. “Scripting Languages Spark New Programming Era.” *InfoWorld*, 2008-06-23. Résumé en ligne <<https://technews.acm.org/archives.cfm?fo=2008-06-jun/jun-25-2008.html&hdr=1#367980>>. Article complet au <<https://www.infoworld.com/article/2651797/application-development/scripting-languages-spark-new-programming-era.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/>>.
- Marji, Majed. *Le grand livre de Scratch*. Paris : Eyrolles 2017. En réserve [ÉPC-Biologie - Didacthèque Jeunesse \(005.13 M3444g 2017\)](#).
- Mighton, John. *All things being equal : why math is the key to a better world*. Canada: Alfred A. Knopf, 2020. BLSH QA 10.7 M54 2020 <<https://umontreal.on.worldcat.org/oclc/1154691205>>.
- Mohan, Mehul. “Why JavaScript Is the Programming Language of the Future.” [En ligne sur freeCodeCamp](#), 2020-04-04.
- Morgan, Nick. *JavaScript for Kids: A Playful Introduction to Programming*. Los Angeles (É.-U.) : No Starch Press, 2015, 305 pp. Version française à l'UdeM : *Javascript pour les kids*. Paris : Eyrolles 2017. En réserve [ÉPC-Biologie - Didacthèque Jeunesse \(006.77 M8494j 2017\)](#).
- Mozilla Developer Network. « Référence JavaScript. » En ligne <<https://developer.mozilla.org/fr/docs/Web/JavaScript/>>.
- 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) <<https://umontreal.on.worldcat.org/oclc/1109981484>>.
- W3Schools. *JavaScript Tutorial*. En ligne <<https://www.w3schools.com/js/>>.
- 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>>.
- Wikipedia. “Software as a service.” En ligne <https://en.wikipedia.org/wiki/Software_as_a_service>. En particulier, la section [Criticism](#).
- Wright, Nolan. “Why JavaScript Will Become The Dominant Programming Language Of The Enterprise.” *readwrite*, 2013-08-09. En ligne <<https://readwrite.com/2013/08/09/why-javascript-will-become-the-dominant-programming-language-of-the-enterprise/>>.

7.3 Références diverses

- Babcock, Charles. “Ajax 101: From Toolkits To Strategy, How Companies Can Put It To Use.” *InformationWeek*, 2006-06-12. En ligne <<https://www.informationweek.com/ajax-101-from-toolkits-to-strategy-how-companies-can-put-it-to-use/d/d-id/1044053>>.

- Ecma International. *ECMAScript® 2020 Language Specification*, Standard ECMA-262, 11th Edition, June 2020. En ligne (PDF et HTML) <<https://www.ecma-international.org/publications/standards/Ecma-262.htm>>.

C'est la spécification ultime du JavaScript, mise à jour pratiquement à chaque année. Malheureusement, illisible pour les non spécialistes.

- Festa, Paul. "Standards Activists Target Scripts." *CNET News* 2005-07-19. En ligne <<https://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*, 2007-06-21. En ligne <<https://www.cnet.com/news/will-ajax-help-google-clean-up/>>.
- Kaneshige, Tom. "A future without programming." *InfoWorld*, 2008-11-20. En ligne <<https://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 <<https://www.infoworld.com/article/2653798/application-development/javascript-creator-ponders-past--future.html>>.
- Kumar, Ritesh. "On Scripting Languages" *OSnews*, 2004-05-13. En ligne <<https://www.osnews.com/story/7038>>.
- Microsoft Corp. "Windows Script Host." 2015-05-07. En ligne <[https://msdn.microsoft.com/en-us/library/9bbdkx3k\(v=vs.84\).aspx](https://msdn.microsoft.com/en-us/library/9bbdkx3k(v=vs.84).aspx)>.
- Savill, John. "How do I run a windows script from the command line?" *WindowsITPro*. En ligne <<https://www.itprotoday.com/devops-and-software-development/how-do-i-run-windows-script-command-line>>.

