



INTRODUCTION

Ce travail consiste à rendre un carnet d'adresses interactif en ajoutant un document JavaScript à ce dernier. Toute la programmation doit se faire à l'aide des modules du DOM 2. Seuls les éléments du DOM 0 qui ne sont pas remplaçables par le DOM 2 sont permis. Pour ce faire, vous devez, à partir d'une nouvelle version du fichier XHTML du Carnet, ajouter un document de scripts permettant de mettre en évidence les lignes cliquées et d'ajouter ou supprimer des lignes au carnet à l'aide d'un menu contextuel. De plus, vous devez rendre le contenu des cellules modifiable et valider les modifications à l'aide d'expressions rationnelles. Ce travail peut être réalisé seul ou en équipe (3 personnes au maximum par équipe). Par contre, son poids a été pensé pour une équipe de trois.

COMPÉTENCES NÉCESSAIRES

- Connaître et savoir utiliser JavaScript et les modules DOM2.

COMPÉTENCES VISÉES

- Après ce travail, vous serez capable de:
 - Utiliser les modules principaux du DOM2 pour réaliser une page WEB interactive.

SOURCES

- Le fichier compressé du cours 11 « 11.10_Exercice_08_carnet_JsDOMCore.zip », disponible sur LÉA. Seul le fichier suivant est nécessaire:
 - La feuille de styles CSS « 11.12_exercice_08_carnet_xhtml_css.css ».
- Le fichier compressé « 13.10_TP2_JsDOM.zip », disponible sur LÉA, comprennent :
 - Le présent document;
 - Le document XHTML « 13.11_TP2_JsDOM.html »;
 - Le fichier PDF du PowerPoint du cours « 13.20_PPT_JavaScript_DOMEvents.pdf »
- <http://www.yoyodesign.org/doc/w3c/dom2-events/Overview.html>
- <http://www.yoyodesign.org/doc/w3c/dom2-style/Overview.html>
- <http://www.yoyodesign.org/doc/w3c/dom2-core/Overview.html>

ÉVALUATION

Vous serez évalué sur la qualité de votre travail.

- Votre code doit être facilement modifiable. Ceci veut dire que:
 - les éléments constants utilisés dans votre code doivent se retrouver dans la section "CONSTANTES" dans le haut de votre programme;
 - les variables globales doivent se retrouver dans une section sous la section "CONSTANTES";



- si certains bouts de code sont répétitifs, ils doivent être regroupés dans une procédure.
- Votre code doit être lisible. Ceci veut dire que:
 - tous les noms utilisés pour identifier des éléments (variable, constante, procédure, etc.) sont représentatifs des éléments qu'ils décrivent (nom significatif);
 - tout le code est indenté correctement;
 - des commentaires ont été ajoutés au besoin pour clarifier certains éléments;
 - vous avez utilisé une façon visuelle de mettre en évidence les sections du programme.
 - Une procédure appelée se retrouve toujours au-dessus de la procédure qui l'appelle.
- Votre code doit être structuré. Ceci veut dire que:
 - les concepts du programme sont regroupés dans des procédures ou objets;
 - vous utilisez des structures de données adaptées à la situation (tableau, objet, etc.).
- Votre code doit être exempt de bogues. Ceci veut dire que:
 - toutes les sections du programme ont été testées à fond;
 - les appels des procédures défectueuses ont été mis en commentaire. Je vous invite dans ces cas à m'indiquer les éléments non fonctionnels avec le nom des procédures impliquées ainsi que la description du dysfonctionnement. Je regarderai si je peux vous accorder des points.
- En résumé:
 - Votre programme doit faire ce qui est demandé dans le présent document. Les bogues seront fortement pénalisés (je dois pouvoir me fier sur le bon fonctionnement de ce que vous créez, sinon, ça ne vaut rien en entreprise).
 - Quelqu'un qui a une application pleinement fonctionnelle (sans bogue) et qui répond à la demande, mais que son code laisse à désirer aura une note supérieure à quelqu'un dont l'application ne répond pas à la demande (ou a des bogues) et qui a un code parfait. Par contre, quelqu'un qui n'a pas une application pleinement fonctionnelle et que le code laisse à désirer, risque d'avoir du chagrin lorsqu'il recevra sa note.

PRODUCTION

Avant de débiter, vous devez vous créer un document « 13.13_TP2_JsDOM.js » pour contenir votre travail. Seul ce document sera à remettre. Je vous invite fortement à utiliser le gabarit fourni au « cours 8 » pour réaliser ce document. Ceci vous assurera d'avoir un cartouche qui vous identifie clairement et d'avoir une structure de base pour le code. Notez que vous pouvez ajouter des lignes au cartouche du gabarit pour indiquer les noms des coéquipiers (ne pas répéter le titre « Auteur », mais lui ajouter un « s »). En aucun cas, les fichiers « 11.12_exercice_08_carnet_xhtml_css.css » et « 13.11_TP2_JsDOM.html » ne doivent être modifiés.

J'ai divisé, ci-dessous, le travail en éléments. Le but ici n'est pas de vous imposer une façon de faire, mais de bien définir chaque partie à réaliser.



Élément 1 - Mise en évidence des lignes sélectionnées:

Cette partie consiste à mettre en évidence la ligne active. Pour activer une ligne, l'internaute n'aura qu'à cliquer dessus (bouton droit ou gauche de la souris).

1. Lorsqu'une ligne est cliquée, elle devra prendre la couleur « #C1D2EE ». Utilisez le DOM2 Style pour modifier la couleur.
2. La ligne précédemment cliquée devra revenir à sa couleur d'origine « #FFFFFF ».
3. L'ajout de la gestion des événements doit se faire à l'aide du DOM2 Events. Notez que le clic droit de la souris devra aussi faire apparaître un menu contextuel éventuellement.
 - a. Attention: Assurez-vous d'avoir des fonctions écouteurs bien formées. Entre autres, n'oubliez pas de mettre un paramètre (et un seul) à vos fonctions écouteurs pour recevoir l'événement.
4. Conseil: Gardez-vous un lien sur le nœud de la ligne cliquée dans une variable globale. Ce dernier sera très pratique lorsque viendra le temps de manipuler cette ligne par le menu contextuel.

Élément 2 - Gestion du survol des lignes (pré requis: Élément 1) :

Cette partie consiste à s'assurer que le survol des lignes fonctionne correctement. Probablement que pour rendre les cellules sélectionnées, vous avez modifié la couleur d'arrière-plan des cellules. Cette action risque d'affecter le survol (hover) des lignes. Donc, pour cette partie, vous devez vous assurer que:

1. Une ligne cliquée (sélectionnée) ne change pas de couleur lors du survol de la souris.
2. Assurez-vous que l'effet de survol fonctionne sur toutes les lignes non sélectionnées. En particulier, vérifiez que le survol fonctionne toujours sur une ligne qui a déjà été sélectionnée, mais qui ne l'est plus.
3. Cette partie nécessite l'utilisation du DOM2 Style et Events.

Élément 3 - Menu contextuel:

Cette partie consiste à créer un menu contextuel qui permettra éventuellement d'ajouter ou de supprimer des lignes. Le but de cette partie n'est pas de mettre en œuvre l'ajout et la suppression, mais bien d'avoir un menu contextuel qui s'affiche. Je vous conseille de réaliser ce menu contextuel à l'aide d'un tableau. N'oubliez pas d'ajouter le menu au document.

2. En vous référant aux saisies d'écrans disponibles à la fin de ce document:
 - a. Créer un menu contextuel à l'aide du DOM2 Core.
 - Les lignes contiendront les éléments: « Ajouter un individu AVANT », « Ajouter un individu APRÈS » et « Enlever l'individu courant ».
 - b. Créer un style CSS pour ce menu contextuel qui sera ajouté à la feuille de style. La création doit se faire à l'aide du DOM2 Style. Bien entendu, les informations nécessaires pour ces styles sont dans un tableau au début du programme (section variables globales ou constantes).
 - Tableau



- Bordure extérieure: couleur « #316AC5 », épaisseur de 2px.
 - Arrière-plan « #FFFFFF ».
 - Cellules du tableau:
 - Pas de ligne séparatrice entre les lignes de ce tableau.
 - Couleur du texte « #000000 ».
 - Lors du survol: arrière-plan « #316AC5 » et couleur du texte « #FFFFFF ».
3. À l'aide du DOM2 Events et Style, modifiez l'événement lié au clic droit d'une ligne de façon à ce qu'il fasse afficher notre menu contextuel à côté de pointeur de souris.
- a. L'événement « click » lorsqu'il est réalisé avec le bouton droit de la souris, permet d'afficher un menu contextuel. Le problème, c'est que nous ne voulons pas que ce menu s'affiche, nous voulons le remplacer par notre menu contextuel. De plus, sur Firefox, il n'est pas possible d'empêcher le menu contextuel par défaut de s'afficher lorsqu'on clique sur le bouton droit de la souris. Pour cette raison, vous devez utiliser l'événement « contextmenu » qui n'est malheureusement pas très bien documenté. De toute façon, il vous offre les mêmes informations que « click » et il vous permettra d'annuler l'affichage du menu par défaut.
 - b. Le positionnement de votre menu contextuel est très important. Lorsque vous remplacez un élément courant, vous devez vous assurer qu'il se présente comme l'internaute s'y attend. Dans notre cas, je vous invite à faire afficher des menus contextuels dans votre navigateur pour déterminer la position d'affichage par rapport au pointeur de souris. Par la suite, il ne vous reste qu'à positionner votre menu de la même façon.
 - c. Il reste à faire apparaître et disparaître votre menu.
 - Au départ, votre menu doit être invisible.
 - Il ne doit apparaître qu'après avoir été positionné suite à un clic droit de la souris.
 - Il doit disparaître si l'on clique en dehors du menu.
 - Il doit aussi disparaître lorsqu'on clique sur l'un des éléments du menu contextuel (ce code sera réalisé à l'étape suivante).

Élément 4 - Rendre le menu contextuel fonctionnel (pré requis: Élément 3) :

Cette partie consiste à rendre le menu contextuel pleinement fonctionnel en ajoutant des fonctions événementielles à chacun des éléments du menu.

1. À l'aide du DOM 2 Events, ajoutez des écouteurs d'événement sur chacun des items du menu pour capturer le « click ».
 - a. Chaque fonction écouteur liée sera chargée de manipuler le contenu du carnet en adéquation avec l'action annoncé par le texte de l'item.
 - b. Chacune des fonctions écouteurs devra faire disparaître le menu contextuel.
 - c. La gestion du « click » doit se faire sur la cellule et non sur le texte.



- d. Attention: Assurez-vous d'avoir des fonctions écouteurs bien formées même si elles sont vides au départ. Entre autres, n'oubliez pas de mettre un paramètre (et un seul) à vos fonctions écouteurs pour recevoir l'événement.
2. Suppression d'un individu (code lié à l'écoute du clic sur l'item « Enlever l'individu courant » du menu)
 - a. Cette procédure est relativement simple, vous n'avez qu'à utiliser le lien que vous aviez conservé dans une variable globale sur la ligne courante pour éliminer la ligne de l'arborescence à l'aide du DOM 2 Core.
 3. Ajout d'un individu au-dessus de la ligne courante (code lié à l'écoute du clic sur l'item « Ajouter un individu AVANT » du menu)
 - a. Cette procédure est un peu plus compliquée. Vous devez tout d'abord créer une nouvelle ligne (DOM2 Core) avec toutes les cellules nécessaires pour être conforme aux autres lignes du carnet (le numéro de la ligne doit apparaître dans la première cellule, les autres cellules seront vides à l'exception de Ville et Province qui devront avoir "? ? ?" comme valeur initiale). Ensuite, vous n'avez qu'à utiliser le lien que vous aviez conservé dans une variable globale sur la ligne courante pour ajouter la nouvelle ligne à l'arborescence à l'aide du DOM 2 Core.
 - Attention : le nouveau numéro de ligne correspond à 1 de plus que le plus grand numéro de ligne actuellement utilisé dans le tableau.
 - Vous devez parcourir le tableau DOM2 Core pour déterminer le chiffre maximal utilisé (cherchez ce nombre qu'une seule fois à l'ouverture de la page et incrémentez-le par la suite). Ceci veut dire qu'en aucun cas, lors d'ajout et de suppression, un numéro d'un élément supprimé précédemment ne sera réutilisé.
 - Rappel: Souvenez-vous qu'une cellule contient un nœud texte.
 4. Ajout d'un individu en dessous de la ligne courante (code lié à l'écoute du clic sur l'item « Ajouter un individu APRÈS » du menu)
 - a. Cette procédure est la plus compliquée. Vous devez, comme pour le précédent item, créer une nouvelle ligne avec toutes les cellules nécessaires pour être conforme aux autres lignes du carnet. Ensuite, vous devez ajouter cette ligne en dessous de la ligne courante. Le problème est que le DOM ne vous offre pas de méthode pour ajouter en dessous. De plus, l'ajout d'une ligne à la fin du carnet est un cas particulier à gérer. Le plus gros problème est que Firefox introduit des nœuds « TEXT_NODE » entre les nœuds des lignes de notre tableau ce qui génère beaucoup de problèmes lorsqu'on veut se promener dans l'arborescence.
 - Conseil: Pour éviter le problème des « TEXT_NODE » de Firefox, vous pouvez utiliser la liste des lignes « getElementByTagName » que vous récupérerez sur le corps de votre objet « carnet ».

NOTE : Lors de l'ajout d'une ligne, le « focus » doit se retrouver sur la cellule nom de cette nouvelle ligne. Puisque la nouvelle ligne est la ligne active, elle doit avoir l'apparence qui convient (mise en évidence).



Élément 5 - Rendre les cellules modifiables:

Cette partie consiste à rendre le contenu des cellules modifiables. Toutes les cellules de contenu seront modifiables à l'exception des cellules de numéros. Bien entendu, les cellules de titres ne seront pas modifiables. De plus, les cellules contenant les villes et les provinces devront être transformées en contrôle de liste déroulante.

1. Modification des contenus des cellules (sauf: No, Ville et Province)
 - a. Le contenu de ces cellules doit devenir complètement modifiable lorsqu'on clique dans la cellule. La façon de faire est laissée à votre discrétion, mais vous devez le faire en utilisant le DOM.
 - b. Bien entendu, la ligne et les cellules sont sélectionnées (ceci veut dire qu'ils ont la couleur d'arrière-plan des lignes sélectionnées).
2. Modification des contenus des cellules Ville et Province
 - a. Pour ces cellules, lorsque l'internaute clique dedans pour les modifier, vous devez remplacer le contenu de la cellule par une liste déroulante. Bien entendu, cette liste sera créée par le DOM Core et sa présentation sera assurée à l'aide du DOM Style. De plus, la valeur qui était affichée dans la cellule sera l'élément sélectionné dans la liste. Lorsque l'internaute quitte la cellule, vous devez enlever la liste déroulante et la remplacer par un nœud texte qui contiendra le texte sélectionné dans la liste. Donc, en résumé, ces cellules affichent toujours un texte directement dans la cellule, mais lorsque l'internaute clic pour modifier le contenu, vous lui présenter une liste déroulante pour faire son choix. Ci-dessous, le contenu des listes déroulantes :
 - Villes : "? ? ?", "Bromptonville", "Fleurimont", "Lennoxville", "Magog", "Montréal", "Québec", "Rock Forest", "Sherbrooke", "Stoke", "Windsor"
 - Provinces : "? ? ?", "Alberta", "Colombie-Britannique", "Île-du-Prince-Édouard", "Manitoba", "Nouvelle-Écosse", "Nouveau-Brunswick", "Nunavut", "Ontario", "Québec", "Saskatchewan", "Terre-Neuve et Labrador", "Territoires du Nord-Ouest", "Yukon");
 - Note : Les points d'interrogation sont les valeurs par défaut des listes lorsque la cellule n'a pas de contenu initial (par exemple, une nouvelle ligne).
 - b. Créez un style CSS pour ce contrôle et ajoutez-le à la feuille de style. La création doit se faire à l'aide du DOM2 Style. Bien entendu, les informations nécessaires pour ces styles sont dans un tableau au début du programme.
 - Notez que la couleur d'arrière-plan de la liste déroulante est la même que celle utilisée pour une ligne sélectionnée (cliquée).

Élément 6 - Validation des modifications des valeurs des cellules (pré requis: Élément 5) :

Cette partie consiste à s'assurer que les valeurs entrées par l'internaute sont valides.

1. Premièrement, lorsque l'internaute quitte une cellule, vous devez vous assurer qu'il ne l'a pas laissée vide.
 - a. Dans le cas des cellules Ville et Province, si l'internaute a laissé ou sélectionné "? ? ?", vous devez lui afficher un message d'avertissement lui indiquant qu'il a laissé la valeur par défaut. À



ce moment, vous lui donnerez deux choix (utilisé la méthode "confirm" pour afficher le message).

- Soit, il veut remédier à cette situation et dans ce cas vous remettrez l'attention sur la liste déroulante.
 - Soit, il ne veut pas remédier à la situation et dans ce cas votre application poursuivra son déroulement normal.
 - Attention: Tant que l'internaute quitte la liste avec une valeur "? ? ?", vous devez lui afficher le message (même si ça fait 20 fois que vous remettez l'attention sur la liste). La seule façon pour l'internaute de ne plus avoir votre message lorsque la valeur est sur "? ? ?" est de vous indiquer qu'il ne désire pas remédier à la situation.
- b. Dans le cas des autres cellules modifiables, si l'internaute a laissé la cellule vide ou avec des espaces détectable par une expression rationnelle (toutes les sortes d'espaces), vous devez lui afficher un message d'avertissement lui indiquant qu'il a laissé la cellule vide. À ce moment, vous devez vider le contenu de la cellule (si elle contenait des espaces) et lui donner deux choix (semblable à ce que vous faites pour une liste modifiable).
2. Deuxièmement, lorsque l'internaute quitte les cellules "CP", "Telephone" et "Courriel", vous devez vous assurer que les valeurs entrées sont adéquates pour le type de contenu attendu. Dans le cas contraire, vous devez lui afficher un message lui indiquant le format attendu et sélectionner le contenu de la cellule en erreur. (utilisé la méthode "alert" pour afficher le message).
- c. En aucun cas, l'internaute ne peut laisser une valeur erronée. Soit il entre une valeur correcte, soit il laisse la cellule vide (dans ce dernier cas, il recevra l'avertissement lui indiquant qu'il a laissé la cellule vide avec les choix qu'offre cette situation).

REMISE

- Le fichier « 13.13_TP2_JsDOM.js » est à remettre sur LÉA (date de remise sur LÉA).
 - Assurez-vous que les noms de tous les coéquipiers apparaissent dans le cartouche de ce document.
 - N'oubliez pas de vous assurer que votre code est impeccable (lisibilité, modifiabilité, structure).



SAISIE D'ÉCRAN

Carnet de téléphone - Mozilla Firefox

Fichier Édition Affichage Historique Marque-pages Outils ?

Carnet de téléphone

| No | Nom | Prénom | Adresse | CP | Ville | Province | Téléphone | Courriel |
|----|------------|-----------|-------------------|---------|------------|-------------------|----------------|---|
| 1 | Côté | Réal | 475, rue du Cégep | J1E 4K1 | ??? | Québec | (819) 564-6586 | Real.Cote@cegepssherbrooke.qc.ca |
| 2 | Dumas | Roméo | 475, rue du Cégep | J1E 4K1 | Sherbrooke | Québec | (819) 564-6367 | Romeo.Dumas@cegepssherbrooke.qc.ca |
| 3 | Fontaine | Francine | 475, rue du Cégep | J1E 4K1 | Sherbrooke | Québec | (819) 564-6368 | Francine.Fontaine@cegepssherbrooke.qc.ca |
| 4 | Gagnon | Julie | 475, rue du Cégep | J1E 4K1 | Lenoxville | Québec | (819) 564-6346 | Julie.Gagnon@cegepssherbrooke.qc.ca |
| 5 | Lamontagne | Carole | 475, rue du Cégep | J1E 4K1 | Sherbrooke | Nouveau-Brunswick | (819) 564-6370 | Carole.Lamontagne@cegepssherbrooke.qc.ca |
| 6 | Lamothe | Régis | 475, rue du Cégep | J1E 4K1 | Sherbrooke | Québec | (819) 564-6365 | Regis.Lamothe@cegepssherbrooke.qc.ca |
| 7 | Lapalme | Jocelyn | 475, rue du Cégep | J1E 4K1 | Sherbrooke | Québec | (819) 564-6587 | Jocelyn.Lapalme@cegepssherbrooke.qc.ca |
| 15 | | | | | ??? | ??? | | |
| 8 | Leduc | Phillipe | 475, rue du Cégep | J1E 4K1 | Sherbrooke | Québec | (819) 564-6372 | Phillipe.Leduc@cegepssherbrooke.qc.ca |
| 9 | Marcellin | Jean-Luc | 475, rue du Cégep | J1E 4K1 | Sherbrooke | Québec | (819) 564-6374 | Jean-Luc.Marcellin@cegepssherbrooke.qc.ca |
| 10 | Tétrault | Louise | 475, rue du Cégep | J1E 4K1 | Sherbrooke | Québec | (819) 564-6375 | Louise.Tetrault@cegepssherbrooke.qc.ca |
| 11 | Tétrault | Claude | 475, rue du Cégep | J1E 4K1 | Sherbrooke | Québec | (819) 564-6376 | Claude.Tetrault@cegepssherbrooke.qc.ca |
| 12 | Tremblay | Micheline | 475, rue du Cégep | J1E 4K1 | Sherbrooke | Québec | (819) 564-6373 | Micheline.Tremblay@cegepssherbrooke.qc.ca |
| 13 | Turcotte | Bertrand | 475, rue du Cégep | J1E 4K1 | Sherbrooke | Québec | (819) 564-6379 | Bertrand.Turcotte@cegepssherbrooke.qc.ca |
| 14 | Vigneux | Richard | 475, rue du Cégep | J1E 4K1 | Sherbrooke | Québec | (819) 564-6369 | Richard.Vigneux@cegepssherbrooke.qc.ca |

W3C XHTML 1.0 W3C CSS

Cette ligne a été ajoutée précédemment avec le menu contextuel.

Carnet de téléphone - Mozilla Firefox

Fichier Édition Affichage Historique Marque-pages Outils ?

Carnet de téléphone

| No | Nom | Prénom | Adresse | CP | Ville | Province | Téléphone | Courriel |
|----|------------|-----------|-------------------|---------|---------------|-------------------|----------------|---|
| 1 | Côté | Réal | 475, rue du Cégep | J1E 4K1 | ??? | Québec | (819) 564-6586 | Real.Cote@cegepssherbrooke.qc.ca |
| 2 | Dumas | Roméo | 475, rue du Cégep | J1E 4K1 | Sherbrooke | Québec | (819) 564-6367 | Romeo.Dumas@cegepssherbrooke.qc.ca |
| 3 | Fontaine | Francine | 475, rue du Cégep | J1E 4K1 | ??? | Québec | (819) 564-6368 | Francine.Fontaine@cegepssherbrooke.qc.ca |
| 4 | Gagnon | Julie | 475, rue du Cégep | J1E 4K1 | Bromptonville | Québec | (819) 564-6346 | Julie.Gagnon@cegepssherbrooke.qc.ca |
| 5 | Lamontagne | Carole | 475, rue du Cégep | J1E 4K1 | Fleurimont | Nouveau-Brunswick | (819) 564-6370 | Carole.Lamontagne@cegepssherbrooke.qc.ca |
| 6 | Lamothe | Régis | 475, rue du Cégep | J1E 4K1 | Lenoxville | Québec | (819) 564-6365 | Regis.Lamothe@cegepssherbrooke.qc.ca |
| 7 | Lapalme | Jocelyn | 475, rue du Cégep | J1E 4K1 | Magog | Québec | (819) 564-6587 | Jocelyn.Lapalme@cegepssherbrooke.qc.ca |
| 15 | | | | | Montréal | Québec | | |
| 8 | Leduc | Phillipe | 475, rue du Cégep | J1E 4K1 | Québec | Québec | (819) 564-6372 | Phillipe.Leduc@cegepssherbrooke.qc.ca |
| 9 | Marcellin | Jean-Luc | 475, rue du Cégep | J1E 4K1 | RockForest | Québec | (819) 564-6374 | Jean-Luc.Marcellin@cegepssherbrooke.qc.ca |
| 10 | Tétrault | Louise | 475, rue du Cégep | J1E 4K1 | Sherbrooke | Québec | (819) 564-6375 | Louise.Tetrault@cegepssherbrooke.qc.ca |
| 11 | Tétrault | Claude | 475, rue du Cégep | J1E 4K1 | Stoke | Québec | (819) 564-6376 | Claude.Tetrault@cegepssherbrooke.qc.ca |
| 12 | Tremblay | Micheline | 475, rue du Cégep | J1E 4K1 | Windsor | Québec | (819) 564-6373 | Micheline.Tremblay@cegepssherbrooke.qc.ca |
| 13 | Turcotte | Bertrand | 475, rue du Cégep | J1E 4K1 | Sherbrooke | Québec | (819) 564-6379 | Bertrand.Turcotte@cegepssherbrooke.qc.ca |
| 14 | Vigneux | Richard | 475, rue du Cégep | J1E 4K1 | Sherbrooke | Québec | (819) 564-6369 | Richard.Vigneux@cegepssherbrooke.qc.ca |

W3C XHTML 1.0 W3C CSS