



Département d'informatique
IFT 359 – Programmation fonctionnelle
Plan d'activité pédagogique
 Automne 2026

Enseignant	Félix-Étienne Trépanier
Courriel :	felix-etienne.trepanier@usherbrooke.ca
Local :	
Téléphone :	
Disponibilités :	Disponible par courriel en dehors des heures de cours pour répondre aux questions. Possibilité de prendre rendez-vous.

Site web du cours : <https://moodle.usherbrooke.ca>

Horaire	Exposé magistral :	Mardi	8 h 30 à 10 h 20	salle L1-4665/L1-3645/L1-5665/L1-5620/L1-11615/L1-3667
		Vendredi	9 h 30 à 10 h 20	salle L1-6670
	Exercices/laboratoires :	Vendredi	8 h 30 à 9 h 20	salle L1-6670

Description officielle de l'activité pédagogique¹

Cibles de formation :	Formaliser les notions d'abstraction procédurale et d'abstraction de données dans le cadre de la programmation fonctionnelle.
Contenu :	Qualité, modularité, conception fonctionnelle. Processus récursifs et itératifs. Objets atomiques. Listes. Abstraction d'ordre supérieur. Curryfication. Fermeture. Appels terminaux. Modèle d'exécution d'un programme fonctionnel. Application de la programmation fonctionnelle (structure de donnée non mutable, programmation par flots, pattern matching...). Insistance sur la qualité de la solution.
Crédits	3
Organisation	3.0 heures d'exposé magistral par semaine 1.0 heure d'exercices par semaine 5.0 heures de travail personnel par semaine
Préalable	IFT159
Particularités	Aucune

¹<https://www.usherbrooke.ca/admission/fiches-cours/ift359>

1 Présentation

Cette section présente les cibles de formation spécifiques et le contenu détaillé de l'activité pédagogique. Cette section, non modifiable sans l'approbation du comité de programme du Département d'informatique, constitue la version officielle.

1.1 Mise en contexte

Le cours IFT 359 introduit la personne étudiante à la programmation fonctionnelle. La programmation fonctionnelle a été introduite avec le langage Lisp créé par John McCarthy en 1958, dans les mêmes années où les langages de haut niveau supportant la programmation impérative (procédurale/structurée) tels que Fortran (1957) et Algol (1958) en étaient à leurs débuts. Lisp est le tout premier langage de programmation fonctionnelle et jusqu'au début des années 1990, c'était le langage de prédilection pour les recherches en intelligence artificielle.

La programmation fonctionnelle se distingue des autres styles de programmation (impérative, logique, objets) par le fait que les algorithmes sont fondamentalement vus comme des fonctions mathématiques. La programmation fonctionnelle est basée sur un modèle mathématique (lambda-calcul) dans lequel une fonction, souvent récursive, prend une entrée et calcule une sortie. La notion d'état d'un programme n'est pas explicite dans ce style de programmation.

À l'opposé du fonctionnel, la programmation impérative, utilisée dans les langages tels C++ et Java, est basée sur un modèle où un état change en fonction des opérations effectuées. La notion d'état d'un programme est très présente dans ce style de programmation. La base théorique sous-jacente est la machine de Turing qui est très proche de l'implémentation physique (registres, mémoire, jeu d'instructions de l'UCT, etc.).

De même, le paradigme de programmation orienté-objet, qui est basé sur le concept d'objet pouvant contenir des données et des méthodes, reste une forme de programmation impérative dans laquelle la notion d'état demeure très présente.

Finalement, l'approche fonctionnelle se distingue aussi de la programmation logique car cette dernière est basée sur l'inférence déductive avec le calcul des prédicats. La programmation fonctionnelle est restée dans l'ombre de la programmation orientée objet, mais elle refait maintenant surface, car ses propriétés sont très utiles pour faire face à certains problèmes actuels (e.g. LINQ, MapReduce).

De nombreux langages utilisent ou sont basés sur l'approche fonctionnelle dont :

1. Le langage Erlang (1986), développé par Ericsson, qui se base sur la programmation fonctionnelle pour offrir un langage propre à la programmation concurrente, parallèle et temps réel ;
2. Le langage Scheme et ses dialectes (Racket, MIT Scheme), souvent utilisés dans le milieu académique, qui est un langage très bien supporté par de nombreux outils et des compilateurs efficaces ;
3. Le langage Haskell (1990 et 1998) qui a été développé pour consolider les acquis des dernières décennies et qui se démarque par son typage statique totalement inféré, ses effets de bord contrôlés et son évaluation paresseuse ;
4. Le langage JavaScript (1995) qui supporte de nombreux concepts de programmation fonctionnelle ;
5. le langage Autolisp (1986) qui est fréquemment utilisé par les ingénieurs à l'intérieur d'Autocad ;
6. Le EmacsLisp (1976) permet d'adapter l'éditeur Emacs à un grand nombre d'applications ou langage informatique ;
7. les langages Common Lisp (1984), OCaml (1996), Scala (2003) et F# (2008) qui combinent la programmation fonctionnelle et la programmation orientée objet ;
8. les langages C#, Python, Java, C++ qui intègrent maintenant plusieurs notions de programmation fonctionnelle.

Étant donné l'influence que le paradigme de programmation fonctionnelle a eue et continuera probablement d'avoir sur les langages de programmation, il est intéressant d'en connaître les principes et les pratiques sous-jacentes.

1.2 Cibles de formation spécifiques

À la fin de cette activité pédagogique, l'étudiante ou l'étudiant devrait :

1. Connaître les principes et les pratiques sous-jacents à la programmation fonctionnelle : abstractions de données, abstractions de fonctions, évaluation des fonctions en ordre normal et en ordre applicatif, transparence référentielle, fonction d'ordre supérieur, lambda-calcul, curryfication ;
2. Savoir implémenter un algorithme dans un langage fonctionnel choisi : définition, utilisation et création des identificateurs et des fonctions ; formes syntaxiques (opérateurs logiques, instructions de contrôle) ; portée lexicale, structures de données basiques (liste, paire, etc.) ;
3. Savoir utiliser les notions clés de la programmation fonctionnelle (récursivité, fonction d'ordre supérieur, fermeture, récursivité terminale, etc.) ;
4. Pouvoir expliquer le déroulement de l'exécution d'un programme fonctionnel : évaluation par environnement d'un programme fonctionnel ;
5. Appliquer des techniques avancées de programmation en s'appuyant sur les propriétés des langages fonctionnels, p. ex. évaluation paresseuse, création de nouvelles formes syntaxiques, création d'un langage embarqué (p. ex. langage à objet), appariement de formes.

1.3 Contenu détaillé

Thème	Contenu	Nbr. d'heures	Objectifs	Lectures ¹
1	Principes théoriques : lambda-calcul, fonction d'ordre supérieur, évaluation en ordre normal, évaluation en ordre applicatif, transparence référentielle, curryfication.	6	1	[1] section 1.3.2 [2] chapitre 20, p. 591-626
2	Introduction au langage fonctionnel choisi : définition des identificateurs, définition des fonctions, portée lexicale, conditions et opérateurs de logique, instructions de contrôle, typage, structures de données basiques (liste, paire, etc.).	5	2	[1] section 1.1 [2] chapitre 1, p. 3-23 ; chapitre 2, p. 25-58
3	Évaluation par environnement d'un programme	4	4	[1] section 3.2 [2] sections 2.5 et 2.6, p. 40-46
4	Fonctions récursives	4	3 et 4	[1] section 1.2 [2] section 1.7, p. 17-21 ; section 2.5, p. 40-45
5	Fonctions d'ordre supérieur	4	3 et 4	[1] sections 1.3 et 2.2 [2] sections 3.1 et 3.6, p. 65-85
6	Appels terminaux	4	3 et 4	[1] sections 1.2.1 à 1.2.4 [2] sections 3.7 à 3.11, p. 83-99

Thème	Contenu	Nbr. d'heures	Objectifs	Lectures ¹
7	Fermeture, affectation ou monade	3	3, 4 et 5	[1] sections 3.1 et 3.3 [2] chapitre 5, p. 125-159
8	Création de nouvelles formes syntaxiques	4	3 et 5	[1] sections 4.1.2 à 4.1.7 [2] chapitre 9, p. 269-304
9	Structure de données non mutable	4	3 et 5	[1] chapitre 2 [2] chapitre 7, p. 185-234
10	Évaluation : méta-évaluateur circulaire en ordre applicatif, évaluation par environnement.	4	3 et 5	[1] sections 3.5 et 4.1 [2] sections 10.1 à 10.4, p. 305-317
11	Évaluation (suite) : méta-évaluateur circulaire en ordre normal, programmation par flots, évaluation paresseuse.	4	3 et 5	[1] sections 3.5 et 4.2 [2] sections 10.5 à 10.9, p. 313-319
12	Création de langage embarqué : exemples	4	3 et 5	[2] chapitre 11, p. 331-378
13	Utilisation explicite de continuations	4	3 et 5	[2] chapitre 8, p. 235-268
14	Études des types : construction et preuves de programme.	4	3 et 5	[2] chapitre 16, p. 491-514
15	Appariement de formes	4	3 et 5	[1] section 4.4 [2] chapitre 17, p. 515-536

¹ Les lectures indiquées ne sont là qu'à titre indicatif. L'enseignant est libre de choisir un autre document de référence

2 Organisation

Cette section propre à l'approche pédagogique de chaque enseignante ou enseignant présente la méthode pédagogique, le calendrier, le barème et la procédure d'évaluation ainsi que l'échéancier des travaux. Cette section doit être cohérente avec le contenu de la section précédente.

2.1 Méthode pédagogique

Les éléments théoriques ainsi que des démonstrations de code seront présentés lors des cours magistraux. Ces concepts seront immédiatement mis en application lors des séances de laboratoire hebdomadaires. L'ensemble des notes de cours, énoncés de laboratoires et exemples de code présentés en classe seront mis à disposition sur l'espace Moodle du cours.

2.2 Calendrier

Semaine	Commençant le	Thème	Éléments principaux
1	2026-08-31	1 et 2	Principes de la programmation fonctionnelle. TypeScript.
2	2026-09-07	2	TypeScript. Types génériques.
3	2026-09-14	9	Abstraction et structures de données fonctionnelles.
4	2026-09-21	4, 6 et 10	Structure récursive. Liste. Récursivité. Récursivité terminale.
5	2026-09-28	5	Fonctions d'ordre supérieur. Applications.
6	2026-10-05	3 et 7	Lambda. Fermeture. Évaluation par environnement. Données mutables.
7	2026-10-12	4, 5 et 9	Séquences. Transformateurs.
8	2026-10-19	Semaine des examens périodiques	
9	2026-10-26	Relâche	
10	2026-11-02	15	Appariement de formes. Type algébrique de données.
11	2026-11-09	14	Utilisation avancée des types. Classes de types.
12	2026-11-16	11	Évaluation paresseuse. Programmation par flot.
13	2026-11-23	7 et 13	Monade. Option. Promesse. Continuation.
14	2026-11-30	12	Création d'un langage embarqué.
15	2026-12-07	8	Macro.
16	2026-12-14	Semaine des examens finals	
17	2026-12-21	Semaine des examens finals	

2.3 Évaluation

Type de l'évaluation	Pondération	Utilisation des IAG ¹
Travaux Pratiques (5)	35 %	Interdite 
Examen intra	25 %	Interdite 
Examen final	40 %	Interdite 

¹ Référez-vous à la page « Balises d'utilisation des outils d'intelligence artificielle générative » à la fin du document.

Les travaux pratiques sont réalisés individuellement et doivent être soumis via Turnin.

Pour réussir le cours, une note minimale de 50% à la moyenne combinée des deux examens ainsi qu'une note minimale de 60% pour l'ensemble des évaluations sont requises.

2.3.1 Qualité de la langue et de la présentation

Conformément à l'article 17 du Règlement facultaire d'évaluations des apprentissages² l'enseignante ou l'enseignant peut retourner à l'étudiante ou à l'étudiant tout travail non conforme aux exigences quant à la qualité de la langue et aux normes de présentation.

2.3.2 Plagiat

Le plagiat consiste à utiliser des résultats obtenus par d'autres personnes afin de les faire passer pour sien et dans le dessein de tromper l'enseignante ou l'enseignant. Vous trouverez en annexe un document d'information relatif à l'intégrité intellectuelle qui fait état de l'article 9.4.1 du Règlement des études³. Lors de la correction de tout travail individuel ou de groupe une attention spéciale sera portée au plagiat. Si une preuve de plagiat est attestée, elle sera traitée en conformité, entre autres, avec l'article 9.4.1 du Règlement des études de l'Université de Sherbrooke. L'étudiante ou l'étudiant peut s'exposer à de graves sanctions qui peuvent être soit l'attribution de la note E ou de la note zéro (0) pour un travail, un examen ou une activité évaluée, soit de reprendre un travail, un examen ou une activité pédagogique. Tout travail suspecté de plagiat sera transmis au Secrétaire de la Faculté des sciences. Ceci n'indique pas que vous n'avez pas le droit de coopérer entre deux équipes, tant que la rédaction finale des documents et la création du programme restent le fait de votre équipe. En cas de doute de plagiat, l'enseignante ou l'enseignant peut demander à l'équipe d'expliquer les notions ou le fonctionnement du code qu'elle ou qu'il considère comme étant plagié. En cas d'incertitude, ne pas hésiter à demander conseil et assistance à l'enseignante ou l'enseignant afin d'éviter toute situation délicate par la suite.

2.4 Échéancier des travaux

Les dates de remise des travaux seront indiquées sur les énoncés.

2.5 Utilisation d'appareils électroniques et du courriel

Selon le règlement complémentaire des études, section 4.2.3⁴, l'utilisation d'ordinateurs, de cellulaires ou de tablettes pendant une prestation est interdite à condition que leur usage soit explicitement permise dans le plan de cours.

Dans ce cours, l'usage de téléphones cellulaires, de tablettes ou d'ordinateurs est autorisé. Cette permission peut être retirée en tout temps si leur usage entraîne des abus.

Tel qu'indiqué dans le règlement universitaire des études, section 4.2.3⁵, toute utilisation d'appareils de captation de la voix ou de l'image exige la permission de la personne enseignante.

Note : Je réponds aux questions posées par courriel à l'extérieur des périodes de cours.

²https://www.usherbrooke.ca/sciences/fileadmin/sites/sciences/documents/Etudiants_actuels/Informations_academiques_et_reglements/2017-10-27_Reglement_facultaire_-_evaluation_des_apprentissages.pdf

³<https://www.usherbrooke.ca/registraire/droits-et-responsabilites/reglement-des-etudes/>

⁴https://www.usherbrooke.ca/sciences/fileadmin/sites/sciences/documents/Etudiants_actuels/Informations_academiques_et_reglements/Sciences_Reglement_complementaire.pdf

⁵<https://www.usherbrooke.ca/registraire/droits-et-responsabilites/reglement-des-etudes/>

3 Matériel nécessaire pour l'activité pédagogique

Aucun.

4 Références

- [1] ABELSON, HAROLD AND SUSSMAN, GERALD J. : *Structure and Interpretation of Computer Programs*. MIT Press, Cambridge, MA, USA, 2nd édition, 1996.
- [2] CHAZARAIN, J. : *Programmer avec Scheme : de la pratique à la théorie*. International Thomson Publ. France, 1996.

Délits relatifs aux études

Extrait du règlement des études (Règlement 2575-009)

Sont notamment considérés comme un délit relatif aux études les faits suivants :

- a) commettre un plagiat, soit faire passer ou tenter de faire passer pour sien, dans une production évaluée, le travail d'une autre personne, des passages ou idées tirés de l'œuvre d'autrui ou du contenu, de toute forme, généré par un système d'intelligence artificielle (ce qui inclut notamment le fait de ne pas indiquer la source et la référence adéquate);
- b) commettre un autoplagiat, soit soumettre, sans autorisation préalable, une même production, en tout ou en partie, à plus d'une activité pédagogique ou dans une même activité pédagogique (notamment en cas de reprise);
- c) usurper l'identité d'une autre personne ou procéder à une substitution de personne lors d'une production évaluée ou de toute autre prestation obligatoire;
- d) fournir ou obtenir toute forme d'aide non autorisée, qu'elle soit collective ou individuelle (incluant l'assistance provenant d'un système d'intelligence artificielle), pour une production faisant l'objet d'une évaluation;
- e) obtenir par vol ou toute autre manœuvre frauduleuse, posséder ou utiliser du matériel non autorisé de toute forme (incluant le matériel numérique et celui généré par un système d'intelligence artificielle) avant ou pendant une production faisant l'objet d'une évaluation;
- f) copier, contrefaire ou falsifier un document pour l'évaluation d'une activité pédagogique;
- k) posséder ou avoir à sa portée un appareil électronique ou numérique interdit durant une activité d'évaluation;

[...]

Un [guide sur l'intégrité intellectuelle](#) vous est rendu disponible par le service des bibliothèques et des archives de l'Université de Sherbrooke, afin de bien comprendre les différents délits et ainsi éviter d'être aux prises avec un dossier disciplinaire et une ou des sanctions.

Les mesures pouvant être imposées à titre de sanctions disciplinaires sont les suivantes :

- a) la réprimande simple ou sévère consignée au dossier étudiant pour la période fixée par l'autorité disciplinaire ou à défaut, définitivement. En cas de réprimande fixée pour une période déterminée, la décision rendue demeure au dossier de la personne aux seules fins d'attester de l'existence du délit en cas de récidive;
- b) l'obligation de reprendre une production ou une activité pédagogique, dont la note pourra être établie en tenant compte du délit survenu antérieurement;
- c) la diminution de la note ou l'attribution de la note E ou 0;

[...]

Balises d'utilisation des outils d'intelligence artificielle générative

Autorisés ou pas dans les situations d'apprentissage et d'évaluation ?

NIVEAU 0

NIVEAU 1

NIVEAU 2

NIVEAU 3

NIVEAU 4

L'utilisation des outils d'intelligence artificielle générative (IAg) est limitée, voire complètement interdite parce que la personne enseignante considère que l'usage de ces outils nuit au développement de compétences essentielles. Ces compétences peuvent être disciplinaires, comme elles peuvent être d'ordre méthodologique, rédactionnel ou informationnel. Considérant que l'utilisation des IAg requiert un esprit critique, il peut s'agir d'une situation d'apprentissage ou d'évaluation sans IAg qui vise à développer celui-ci.

Dans ces situations, **la personne étudiante produit le travail.**

L'utilisation prononcée des IAg est permise parce que la personne enseignante considère que les personnes étudiantes sont en mesure d'exercer un esprit critique et sont capables de juger de la qualité des contenus produits par les IAg. Ou encore, l'utilisation est encouragée parce que la situation d'apprentissage ou d'évaluation proposée contribue à développer leur esprit critique.

Dans ces situations, l'IAg produit le travail préliminaire, alors que **la personne étudiante s'assure de sa qualité en l'améliorant.**



Utilisation interdite

Le **NIVEAU 0** signifie que l'**utilisation est interdite**.

Ceci signifie que si la personne enseignante a un motif de croire qu'il y a eu l'utilisation d'une IAg dans une situation d'évaluation, elle doit dénoncer les faits auprès de la personne responsable des dossiers disciplinaires universitaires. Il s'agit d'un délit relatif aux études tel que stipulé dans le [Règlement des études](#).



Utilisation limitée

Le **NIVEAU 1 D'UTILISATION** signifie que l'**utilisation est autorisée uniquement pour assister l'apprentissage dans le domaine disciplinaire ou des langues**.

Dans ce contexte, la personne étudiante **est tenue de déclarer l'utilisation qu'elle en a faite** selon les consignes fournies par la personne enseignante sans quoi l'utilisation peut être considérée comme un délit. Par exemple :

Domaine disciplinaire :

- S'inspirer
- Générer des idées
- Explorer un sujet pour mieux le comprendre
- Générer du matériel pour apprendre

Domaine des langues :

- Identifier ses erreurs et se les faire expliquer
- Reformuler un texte
- Générer un plan pour aider à structurer un texte
- Traduire un texte



Utilisation guidée

Le **NIVEAU 2 D'UTILISATION** signifie que l'**utilisation est autorisée pour améliorer un travail produit par la personne étudiante**.

Dans ce contexte, la personne étudiante **est tenue de déclarer l'utilisation qu'elle en a faite** selon les consignes fournies par la personne enseignante sans quoi l'utilisation est considérée comme un délit. Par exemple :

- Analyser des contenus
- Obtenir une rétroaction
- Évaluer la qualité de son travail à partir de critères
- Demander à être confronté relativement à ses idées, à sa démarche
- Diriger les processus de résolution de problèmes



Utilisation balisée

Le **NIVEAU 3 D'UTILISATION** signifie que l'**utilisation est autorisée pour produire un travail qui sera amélioré**.

Dans ce contexte, la personne étudiante **est tenue de citer selon les normes¹ le contenu généré par l'IAg ou de déclarer l'utilisation qu'elle en a faite** selon les consignes fournies par la personne enseignante sans quoi l'utilisation est considérée comme un délit. Par exemple :

- Résumer ou rédiger des parties d'un texte
- Générer un texte ou un modèle d'une production et l'adapter
- Réaliser des calculs mathématiques
- Produire du code informatique
- Résoudre des problèmes complexes
- Répondre à une question
- Générer des images, ou autres contenus multimédias



Utilisation libre

Le **NIVEAU 4 D'UTILISATION** signifie qu'**aucune restriction spécifique n'est imposée**.

Dans ce contexte, la personne étudiante **est tenue de citer selon les normes¹ le contenu généré par l'IAg ou de déclarer l'utilisation qu'elle en a faite** selon les consignes fournies par la personne enseignante sans quoi l'utilisation est considérée comme un délit.

Ce niveau inclut tout ce qui précède, de l'exploration à la production, ainsi que toute autre tâche particulière jugée complexe.

À considérer avant l'utilisation d'outils d'intelligence artificielles génératives

Si, en tant que personne étudiante envisagez d'utiliser un outil d'intelligence artificielle générative (IAG) lorsque l'évaluation autorise les niveaux 1 à 4 d'utilisation mentionnés précédemment.

Dans ce cas, gardez à l'esprit les éléments clés suivants.

- Vous assumez la responsabilité de tout le contenu produit, avec ou sans IAG, et intégré à votre production.
- Les produits des outils d'IAG peuvent très souvent comporter **des erreurs ou des faussetés** (hallucinations) : on doit donc impérativement valider tout contenu généré par ces outils.
- Dans l'état actuel de la Loi sur le droit d'auteur du Canada, les **productions faites par l'IAG sont du domaine public**, puisque les outils d'IAG ne sont pas reconnus comme des auteurs au sens de la Loi et que les contenus générés ne répondent pas aux critères d'une œuvre protégée, notamment aux critères d'originalité.
- L'entreprise qui fournit le service pourrait émettre certaines exigences dans ses conditions d'utilisation. Comme l'algorithme et le code informatique appartiennent à l'entreprise qui les a développés, nous devons tenir compte de ces conditions. Celles-ci pourraient également fournir des précisions relatives à la **réutilisation des données soumises (confidentialité)**.

Comment déclarer l'utilisation d'outils d'intelligence artificielle générative

Dans l'esprit d'une conduite intègre et responsable, vous devez TOUJOURS mentionner de façon explicite toute utilisation de l'intelligence artificielle, conformément au Règlement des études (9.4.1 Délits relatifs aux études). De plus, à des fins pédagogiques, il est recommandé de toujours intégrer à la production les requêtes, de même que les réponses intégrales générées par les outils d'IAG. Celles-ci pourront être intégrées directement dans le corps du texte ou en note de bas de page. Les réponses longues pourraient être insérées en annexe de votre document ou dans des documents supplémentaires, selon les directives de la personne enseignante.

L'utilisation de ces deux documents s'avèrera utile, ils se trouvent sous licence libre, donc vous pouvez utiliser les tableaux et les adapter selon votre besoin:

1. Modèle de citation : Ce formulaire, à remplir par l'enseignant, donne un exemple aux étudiants de citation de l'IAG dans la réalisation d'un travail évalué ou non.
2. Déclaration d'usage : Ce formulaire, à remplir par les étudiants, doit être remis avec une réalisation afin de déclarer l'usage de l'IAG dans la réalisation, qu'elle soit évaluée ou non.

Référence

La Faculté des sciences tient à remercier le SSF pour la production des documents.

- Cabana, M. et Côté, J.-A. (2024). Balises d'utilisation des outils d'intelligence artificielle générative. Service de soutien à la formation, Université de Sherbrooke. Sous licence [CC BY 4.0](#).
- Cabana, M. et Beaudet, M. (2024). Directives de déclaration de l'utilisation de l'intelligence artificielle générative dans une production étudiante. Service de soutien à la formation, Université de Sherbrooke. Sous licence [CC BY 4.0](#).
- Cabana, M. (2024). Formulaire de déclaration de l'utilisation de l'intelligence artificielle générative dans une production étudiante. Service de soutien à la formation, Université de Sherbrooke. Sous licence [CC BY 4.0](#).