

Intégration II

582-204-SF

2h théorie
4h pratique
3h temps personnel

Filières : 0001 et 0002
Préalables : Intro à la programmation (582-104-SF)
Intégration I (582-101-SF)

David Lapointe Gilbert
dlapointegilbert@csfoy.ca

Teams @David Lapointe Gilbert
GitHub <https://github.com/dlapointegilbert>

<input type="checkbox"/>	<input checked="" type="checkbox"/>	Problème lors de la minification des CSS bug good first issue #89 opened on 20 May 2021 by ferlandalex
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Ajout d'une petite ligne sous les titres css #82 opened on 18 May 2021 by Camillemarceau07
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Comment faire "bleeder" correctement mon background sur 1440px? css #80 opened on 17 May 2021 by jeffduval
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Intégration d'un nav bar non rectangulaire css enhancement good first issue #76 opened on 17 May 2021 by Camillemarceau07
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Quelle est meilleure manière d'intégrer le visuel de l'expérience immersive? good first issue #67 opened on 13 May 2021 by jeffduval
<input type="checkbox"/>	<input checked="" type="checkbox"/>	comment est ce que je peux intégrer la rose des vents pour le format mobile ? css #59 opened on 10 May 2021 by lounaparron
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Design Camille Proposition Intégration collaboration good first issue #56 opened on 6 May 2021 by lounaparron
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Changement de couleur au survol d'un svg css good first issue #47 opened on 5 May 2021 by OlivierLemay
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Bouton bascule FullScreen bug JavaScript #45 opened on 4 May 2021 by evefevrier

Thématique

Le cours Intégration II est le second et **dernier** cours de cette série.

Il vise à approfondir la connaissance et la compréhension des normes, des meilleures pratiques et des méthodes de travail permettant l'intégration de projets Web. Au fil du cours, un *Guide de développement Web*¹ sera élaboré et mis à jour pour définir un ensemble de bonnes pratiques et méthodes de travail en respect du style de codage du programme ainsi qu'une liste de ressources et un carnet de notes personnelles au format Markdown.

Le bloc 1 traite

- des normes reliées à la description et la mise en forme des contenus : Markdown, HTML, CSS et WCAG² ;
- des outils et extensions pour améliorer l'efficacité et contrôler la qualité du travail ;
- des concepts, notions et méthodes de travail avec les CSS incluant l'approche *Mobile First*, le découpage modulaire des fichiers et l'utilisation de variables CSS ;
- d'interactivité avec JavaScript;
- des techniques pour implémenter des grilles et des conceptions fluides et réactives : les flexbox, puis, les CSS *Grids* et enfin en combinant les différentes stratégies disponibles ainsi que les possibilités graphiques des CSS selon l'approche du *Design Web intrinsèque*³.

Le bloc 2 s'articule autour de la réalisation complète de projets Web incluant une mise en ligne sous https.

Un de ces projets est un travail d'équipe qui est réalisé successivement ou parallèlement dans tous les cours de la formation spécifique.

¹ <https://github.com/dlapointegilbert/guide-developpement-web>

² Les règles et techniques pour l'accessibilité. *Web Content Accessibility Guidelines*.

³ Ce terme a été proposé par Jen Simmons.

Attitudes socioprofessionnelles (soft skills)

Les attitudes socioprofessionnelles sont des savoir être, des savoir-faire relationnels, collaboratifs et organisationnels qui complètent le profil professionnel du domaine d'emploi. Voici un exemple tiré d'une offre d'emploi en décembre 2021 :

Débrouillard et autonome;

Efficace et rigoureux;

Équipier de pointe, aimant partager et apprendre en équipe;

Ouvert et en mode recherche de solutions;

Polyvalent, navigue aisément à travers différents types de mandats

Bonne écoute et esprit d'analyse.

Dans le cours d'Intégration II et plus généralement, dans le programme TIM, une participation constructive à la communauté (la classe, la cohorte, ...) est la voie privilégiée pour le développement des attitudes socioprofessionnelles.

Stratégies d'enseignement

Les notions et techniques sont présentés à l'aide d'exposés ponctués de démonstrations et de petits exercices pour alterner les modes passif-actif autant que possible. Les périodes de laboratoire sont utilisés pour avancer en présence les travaux pratiques afin de permettre à l'étudiant(e) de s'approprier la matière au fur et à la mesure. Le travail est souvent exécuté en binôme avec des retours en plénière. La présence en classe est obligatoire (voir l'article 7 du règlement numéro 6 sur la présence aux cours⁴).

Le cours d'Intégration II est porteur d'un projet pédagogique de « communauté d'apprentissage ». Ce projet s'appuie sur la théorie socioconstructiviste qui avance que la cognition, les apprentissages se réalisent le mieux lors d'interactions sociales positives et structurantes.

Dans cet esprit, la déclaration d'inclusivité (<https://github.com/integration2>), est l'assise de la communauté d'apprentissage aussi chaque personne étudiante est appelée à signer cette déclaration pour s'engager à la respecter.

La communauté d'apprentissage se forme et s'élève en s'ouvrant sur les communautés de pratique tels que FrontEndMentors et **GitHub**. concrètement, l'ouverture vers les milieux professionnels se fait par :

1. la présence physique dans la communauté d'apprentissage
la présence numérique sur certains médias sociaux :
 - a. LinkedIn du programme : <https://www.linkedin.com/groups/2211970/>
 - b. **Git, GitHub et GitHub Classroom seront utilisés en classe**
2. l'utilisation des communautés de pratique :
 - a. StackOverFlow
 - b. Salons de développeurs sur Discord [en anglais seulement]
(Ces salons ont des codes de conduite très stricts auxquels il est impératif d'adhérer pour le bien-être de tous)
 - **Frontend Mentors** : <https://discord.com/invite/frontend-mentor-824970620529279006>,
<https://www.frontendmentor.io>
 - **Learn Build Teach**: <https://www.learnbuildteach.com/>

Stratégies d'évaluation

Évaluation formative

Toutes les remises sont formatives et visent l'acquisition progressive des compétences.

Évaluation sommative par entrevue à la fin de la session

Tous les livrables, examen, exercices et travaux constituent le portfolio de la personne étudiante.

⁴ https://www.csfoyc.ca/fileadmin/documents/notre_cegep/politiques_et_reglements/Reglement_6_reussite_scolaire.pdf

L'entrevue consiste à identifier le niveau d'atteinte de chacun des objectifs d'apprentissage ciblés pour ce cours (voir la liste des objectifs en page suivante).

Notation

En accord avec les principes de la dénotation, la note finale sera obtenue non pas par addition des points (puisque les livrables ne sont pas notés) mais en composant la note finale d'après l'évaluation faite en entrevue du niveau atteint pour chacun des 9 objectifs d'apprentissage listés ci-bas.

Quatre notes sont possibles : 50%, 60%, 85%, 100%.

Objectifs d'apprentissage

1. Maîtriser le balisage HTML sémantique, valide et accessible

- Utiliser les éléments HTML de manière appropriée pour structurer le contenu de manière claire et significative (titres, paragraphes, listes, sections, formulaires, etc.).
- Valider le code HTML à l'aide d'outils comme l'inspecteur de DOM, les validateurs en ligne et en collectant les feedbacks des pairs.
- Implémenter des attributs ARIA (Accessible Rich Internet Applications) pour enrichir l'accessibilité des interfaces dynamiques.
- Ajouter des alternatives textuelles pertinentes aux images, vidéos et éléments multimédias.
- Corriger les erreurs de balisage identifiées lors des validations.

2. Organiser et structurer les styles CSS de manière modulaire

- Structurer le CSS en utilisant une approche modulaire.
- Compiler des fichiers SCSS en CSS grâce à un préprocesseur comme SASS pour centraliser et optimiser les styles.
- Utiliser des variables, mixins et fonctions SCSS pour assurer la réutilisabilité et la maintenabilité du code.
- Implémenter des conventions de nommage cohérentes (BEM) et documenter le code.
- Intégrer des outils d'analyse (comme StyleLint) pour vérifier et corriger la cohérence du code CSS.

3. Comprendre et appliquer les concepts fondamentaux des CSS

- Expliquer et appliquer les principes de la cascade CSS (priorité des règles de style).
- Utiliser des sélecteurs efficaces en maîtrisant la spécificité, en appliquant la méthodologie BEM (Block, Element, Modifier) et en exploitant l'imbrication de manière réfléchie et intentionnelle.
- Gérer l'héritage et réinitialiser les styles de manière appropriée pour éviter des conflits.
- Exploiter les pseudo-éléments et pseudo-classes (`::before`, `::after`, `:hover`, `:nth-child()`, etc.) pour enrichir les styles.
- Résoudre des conflits CSS réels en expliquant les causes et les solutions.

4. Maîtriser les stratégies de mise en page en CSS

- Utiliser les boîtes flexibles (flexbox) pour créer des dispositions dynamiques et adaptatives.
- Créer des mises en page complexes en utilisant les grilles CSS (grid layout).
- Implémenter des positionnements relatifs et absolus pour des placements précis d'éléments.
- Personnaliser les dispositions en fonction des préférences des utilisateurs grâce aux container queries et prefers-color-scheme.
- Créer des micro-layouts réutilisables pour des sections répétées comme des cartes de contenu ou des composants hero.

5. Intégrer des transitions et des animations pour améliorer l'expérience utilisateur

- Créer des transitions fluides pour les interactions utilisateur (hover, focus, clic).
- Développer des animations CSS simples à l'aide de keyframes.
- Utiliser des propriétés optimisées pour la performance (transform, opacity).
- Intégrer des animations conditionnelles en fonction des interactions des utilisateurs sur différents supports (touch, souris, clavier).
- Implémenter des animations adaptées aux utilisateurs sensibles en respectant les règles prefers-reduced-motion.

6. Intégrer des pratiques favorisant l'accessibilité web

- Ajouter des alternatives textuelles pertinentes pour les images, vidéos et contenus multimédias.
- Créer des formulaires accessibles avec des labels associés, des champs obligatoires bien indiqués et des messages d'erreur clairs.
- Utiliser des contrastes adéquats pour les textes et les éléments interactifs selon les critères WCAG.

- Tester l'accessibilité à l'aide d'outils automatisés et de vérifications manuelles avec des lecteurs d'écran comme NVDA ou VoiceOver.
- Intégrer des attributs ARIA dynamiques (aria-live, role, etc.) pour améliorer l'accessibilité des interfaces interactives.

7. Utiliser un système de versionnage avec Git et GitHub

- Créer et gérer des dépôts locaux et distants via Git et GitHub.
- Utiliser les commandes de base (init, clone, add, commit, push, pull) et des commandes avancées (merge, rebase) pour résoudre des conflits.
- Créer et gérer des branches pour isoler les modifications du code.
- Documenter l'historique des modifications avec des messages de commit explicites et descriptifs.
- Utiliser les pull requests pour demander des révisions de code et intégrer les suggestions.

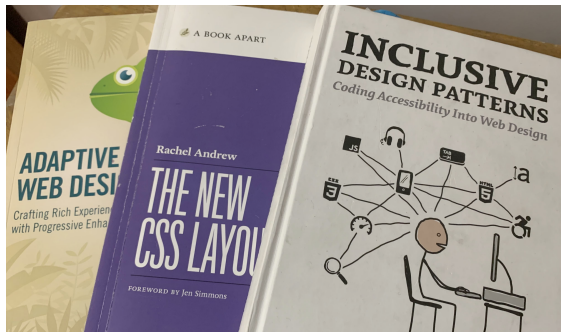
8. Optimiser les médias pour le Web et comprendre les images réactives

- Choisir des formats de fichiers adaptés (JPEG, PNG, WebP, SVG, AVIF) selon leurs cas d'utilisation.
- Utiliser des outils de compression pour réduire le poids des images sans perte de qualité notable.
- Implémenter des images réactives (srcset, picture) pour adapter les visuels aux différentes tailles d'écran.
- Gérer le chargement différé des images (lazy loading) pour optimiser les temps de chargement.
- Documenter les choix de formats et les optimisations appliquées.

9. Utiliser des outils d'intelligence artificielle pour l'apprentissage et la productivité

- Interroger des modèles d'intelligence artificielle pour obtenir des explications sur le code et des suggestions de refactorisation.
- Vérifier les suggestions de l'IA en fonction des standards de développement web et corriger si nécessaire.
- Utiliser l'IA pour générer du code récurrent, automatiser la documentation et accélérer les tâches répétitives.
- Créer des prompts efficaces et spécifiques pour obtenir des résultats précis.
- Documenter les prompts, les adaptations et modifications faites aux réponses de l'IA.

Médiagraphie



Quelques sites de références

Apprendre le développement Web
<https://developer.mozilla.org/fr/docs/Learn> [FR]

HTML 5
 Référence des éléments HTML
<https://developer.mozilla.org/fr/docs/Web/HTML/Element>

HTML (une norme vivante)
<https://html.spec.whatwg.org/> [EN]

Accessibilité Web
 Making the Web Accessible
<https://www.w3.org/WAI/> [EN]
<https://www.w3.org/standards/webdesign/accessibility> [EN]

Directives d'accessibilité du W3C, 2.0
<http://www.w3.org/Translations/WCAG20-fr/> [FR]

Feuilles de style en cascade CSS 3 [EN]
<http://www.w3.org/Style/CSS/>

Matériel requis

Éditeurs de code

- Extensions pour VS Code
- **Live Server**,
- **GitLens, Git supercharged**,
- **Markdown Preview Enhanced**

- Extensions pour PhpStorm
- Live Edit

Outils

- Navigateurs (**Chrome**, Safari, Firefox, Edge)
- Émulateurs (*Toggle Device* de DevTools)
- Responsively (Tests de sites adaptatifs)

Barre d'outils de développement de Chris Pederick
 Terminal (Mac) ou invite de commande (Windows)

FileZilla

Validateurs (HTML, CSS)

- Outils et services en ligne:
- pour la compression des images, des CSS;
 - pour le partage de code;
 - pour le partage de fichiers;

Git et GitHub

Logiciels ou Apps

- Adobe CC (Photoshop, Illustrator, Audition...)
- Figma

Inspiration

- Visuel / design : [Dribbble](#), [Behance](#)
- Web : [Awwwards](#), [CSS Design Awards](#), [One Page Love](#), [Hover States](#), [Site Inspire](#)
- Actualités : [Grenier aux nouvelles](#)

Calendrier

Le contenu des périodes de cours est sujet à ajustement durant la session.

Sem.	Période	Contenu	Travaux pratiques
BLOC 1 – Environnements, méthodes de travail et bonnes pratiques			
1	Jeudi 22-01	1 Plan de cours. Communauté d'apprentissage et réseau de communautés de pratique. GitHub. Markdown.	Configuration de l'environnement de travail. Accueil des élèves de 3 ^e année pour les entrevues utilisateur du site de la TIM
	Lundi 26-01	2 HTML, XML, Règles d'écriture pour la conformité, doctypes et validation Emmet. Extensions pour VS Code et les navigateurs. Validateur HTML du W3C.	Exercice HTML + validateurs
2	Jeudi 29-01	3 Amélioration progressive Entités HTML Test HTML	Jalon de validation – Objectif d'apprentissage 1 (HTML sémantique et accessible)
	Lundi 02-01	4 <i>Laboratoire & préparation au versionnage</i>	<i>Premiers contacts avec les communautés de pratique GitHub et FrontEndMentors</i>
3	Jeudi 05-02	5 Initiation au versionnage avec Git et GitHub.	Guide de développement Web
	Lundi 09-02	6 CSS : cascade et héritage Typographie et images réactives (Mobile First) Ressources : les API de services media Google Fonts, Placeholder.com...	

Sem.	Période	Contenu		Travaux pratiques
4	Jeudi 12-02	7	Sélecteurs CSS. Nomenclature BEM.	Exercices (katas) de sélecteurs CSS
	Lundi 16-02	8	CSS : spécificité, modèles de boîte, positionnement, flexbox	Exercices (katas) de flexbox
5	Jeudi 19-02	9	Grilles CSS. Penser la grille fluide. Générateurs, mode mise en page de l'inspecteur de code.	Exercices (katas) de grille css
	Lundi 23-02	10	Grilles CSS (fonctions CSS, patterns...) Transitionner et Animer	
6	Jeudi 26-02	11	Introduction à SASS. Variables, modularité des fichiers, table des matières navigable. Test CSS	Jalon de validation – Objectifs d'apprentissage 2 à 5 (organisation et fondamentaux CSS)
	Lundi 02-03	12	WCAG : techniques pour rendre accessibles les contenus	Exercices (katas) d'accessibilité
	Mercredi 04-03	14	Conférence-atelier sur l'accessibilité avec le RAAMM / de 12h à 14h	
7	Jeudi 05-03	13	WCAG : balisage html pour l'accessibilité des formulaires	
	Lundi 09-03	14	WCAG : styler pour l'accessibilité des formulaires + utiliser le lecteur vocal pour tester Cours réduit à la dernière heure : 10h à 11h / 12h à 13h (bimodale au besoin)	
SEMAINE R : Journées d'activités périscolaires le 9 mars et 14 avril, journées de relâche du 10 au 13 mars				
8	Jeudi 12-03	15	Introduction aux méthodes agiles. [Projet d'équipe] Énoncé. Contrat d'équipe. Rôles. Poker planning pour Projet synthèse.	Contrat d'équipe
	Lundi 16-03	16	WCAG : navigation accessible (menu javascript accessible imposé dans le projet MAIS à personnaliser)	
9	Jeudi 19-03	17	Travail d'équipe – Guide de survie Utiliser Git et GitHub de manière collaborative	Activité : Escarmouche (régler des conflits de versionnage)
	Lundi 23-03	18	Intégrer des SVG Test A11Y	Jalon de validation – Objectif d'apprentissage 6 (accessibilité)

Calendrier (suite)

Sem.	Période	Contenu	Travaux / Préparations
BLOC 2 – Projets			Dans le bloc des projets, les remises seront précisées au fur et à la mesure
10	Jeudi 26-03	19 [Projets individuels]	
	Lundi 30-03	20 [Projets individuels]	
11	Jeudi 02-04	21 [Projets individuels]	
	Lundi 06-04	22 [Projets individuels]	
Congés de Pâques : du vendredi 3 avril au lundi 6 avril inclusivement			
12	Jeudi 9-04	23 [Projet d'équipe] Prévoir les tâches d'intégration et les ajouter avec des post-its Activité découverte et préparation des medias pour la fiche Scrum / Laboratoire intégrer le HTML Création du répertoire d'équipe sur Timunix3	
	Lundi 13-04	24 [Projet d'équipe] Mise au point de la procédure de versionnage Scrum / Discuter les stratégies d'intégration (grids, flexbox...) Laboratoire intégrer le HTML	
13	Lundi 16-04	25 [Projet d'équipe] Laboratoire intégrer le HTML du projet synthèse Schématiser les stratégies d'intégration	
	Jeudi 20-04	26 [Projet d'équipe] Infos pour le mandat D : formulaire et API Google Maps Test GIT, Optimisation et IA	Jalon de validation – Objectifs d'apprentissage 7 à 9 (GIT, Optimisation et IA)
14	Lundi 01-05	27 [Projet d'équipe] Schématiser les stratégies d'intégration Scrum / Laboratoire intégrer les styles CSS	
	Jeudi 05-05	28 [Projet d'équipe] Scrum / Laboratoire intégrer les styles CSS Contrôle qualité / Mise en ligne	
15	Lundi 08-05	29 [Projet d'équipe] Scrum / Laboratoire intégrer les styles CSS Infos et préparation pour les entrevues finales	
	Jeudi 12-05	30 Finalisation des projets et début des entrevues finales	
16		Entrevues finales	Autoévaluation à compléter avant la rencontre. Présentation du portfolio constitué des Travaux pratiques et des projets.

Règles départementales

Ce plan de cours est complété par le document rassemblant les règles départementales.
Ce document est disponible dans la section *Documents distribués* du cours sur Léa.