



TECHNIQUES D'INTÉGRATION MULTIMÉDIA  
COLLÈGE DE MAISONNEUVE

---

## PLAN DE COURS

---

### IDENTIFICATION DU COURS :

Numéro du cours :	582-4PA
Titre du cours :	<b>Interfaces Web réactives et animées</b>
Pondération :	3-2-2
Durée :	75 heures
Session :	Hiver 2024
Professeurs :	Camille Semaan (messageries <i>Mio</i> et <i>Teams</i> ), bureau B-1163
Préalables :	<b>Préalable absolu</b> : aucun <b>Préalable relatif</b> : 582-3W3 Création de sites Web dynamiques

**Collège de Maisonneuve**  
3800 rue Sherbrooke Est . Montréal, Québec, H1X 2A2  
514-254-7131

PRÉAMBULE : Dans ce document, le masculin est utilisé pour désigner autant le féminin que le masculin.  
Ce choix éditorial a pour unique but d'alléger la lecture du texte.

# 1. Présentation du cours

---

## 1.1 Description du cours

Dans ce cours, l'étudiant ayant un intérêt plus marqué pour la programmation a la possibilité d'approfondir les acquis des cours de la séquence Web en s'initiant à des technologies permettant de réaliser des applications riches du côté client.

Il apprendra à utiliser un cadre côté client pour produire un site Web dynamique dont l'interface utilisateur réactive répond en temps réel aux modifications d'états des données de l'application. L'intégration d'effets de transition et d'animation dans l'interface-utilisateur est également couverte.

## 1.2 Place du cours dans le programme

Le cours Interfaces Web réactives et animées (582-4PA) est un cours de quatrième session et a comme préalable relatif le cours Création de sites Web dynamiques (582-3W3).

Ce cours est relié aux champs de compétences Programmation et Création de médias. Il permet à l'étudiant d'intégrer les connaissances acquises en programmation Web et en traitement d'images animées pour créer des applications Web client-serveur à interactivité asynchrone et interface utilisateur riche et animée. Le cours introduit aussi les étudiants à de nouveaux outils essentiels dans la chaîne de production multimédia (environnement de développement).

Les notions de programmation asynchrone, de cadres applicatifs côté client, de développement d'une interface de programmation (API), et d'animations de l'interface utilisateur qui sont acquises dans ce cours peuvent servir au besoin dans les cours Projet Web en équipe (582-5W5) et Projet de fin d'études (582-6N2).

Ce cours vise à transmettre et évaluer les connaissances, habiletés et techniques essentielles à la programmation d'applications Web modernes, utilisant des cadres logiciels permettant une interactivité riche et animée côté-client et une intégration de la logique applicative côté serveur de façon réactive. L'étudiant apprend à utiliser un cadre de programmation côté client et à programmer les fonctionnalités côté serveur sous forme d'API HTTP, pour produire une application Web à interface riche sur différentes plateformes.

Le cours introduit le concept de programmation client-serveur asynchrone, les motifs d'architecture logicielle, les concepts de programmation d'une API côté serveur, et des techniques d'animation de l'interface utilisateur pour produire une application réactive, conviviale, et répondant à certains critères d'utilisabilité.

## 2. Démarche d'enseignement et d'apprentissage

---

L'acquisition de la matière se fait par l'exemple et par la pratique : de courtes périodes d'explications théoriques sont aussitôt suivies de démonstrations de mise en pratique des notions vues sur un projet de session intégrateur.

Des exercices pratiques proposés à chaque étape importante de l'apprentissage, permettent aux étudiants de confirmer l'acquisition de la matière.

## 3. Sommaire des évaluations

---

Échéance	Description de l'évaluation	Pondération
Tout au long de la session	Exercices : 6 exercices aux semaines 2, 4, 7, 10, 12 et 14	30 %
Semaine 8	TP #1 : Produire une application Web à partir de composants React modulaires intégrant des effets d'animation	25 %
Semaine 16	TP #2 : Produire un site web réactif monopage intégrant des données non-relationnelles nuagiques	25 %
Semaine 15	Épreuve finale du cours	20 %
	<b>TOTAL :</b>	<b>100%</b>

## 4. Calendrier des activités

SEMAINE	DESCRIPTION SOMMAIRE DU CONTENU
<b>ÉTAPE #1 : Programmer l'interactivité de l'interface utilisateur en intégrant les fonctionnalités d'un cadriciel Web côté client.</b>	
Semaine 1	<p><b><u>Notions théoriques</u></b></p> <ul style="list-style-type: none"><li>• Notions de base reliées aux cadriciels JavaScript côté client</li><li>• Introduction à <i>React</i></li></ul> <p><b><u>Pratique</u></b></p> <ul style="list-style-type: none"><li>• Préparation d'un environnement de développement local</li><li>• Création de projets <i>React</i> avec l'outil <i>Vite</i></li><li>• Rappels <i>Git/GitHub</i></li><li>• Compilation du projet (build)</li><li>• Déploiement sur <i>GitHub Pages</i></li></ul>
Semaine 2	<p><b><u>Notions théoriques</u></b></p> <ul style="list-style-type: none"><li>• Démarrage d'un projet d'application Web <i>React</i></li><li>• Rendu des éléments de base (<i>ReactDOM</i>)</li><li>• Composants et props (propriétés)</li><li>• Introduction à <i>JSX</i></li></ul> <p><b><u>Renforcement JavaScript</u></b></p> <ul style="list-style-type: none"><li>• Déclaration de variables (<i>let</i>) et de constantes (<i>const</i>)</li><li>• Les fonctions fléchées</li><li>• Valeurs de retour implicites</li></ul> <p><b><u>Pratique</u></b></p> <ul style="list-style-type: none"><li>• Différentes façons de démarrer/configurer un projet de site Web <i>React</i></li><li>• Importation/Exportation du code JS avec <i>Webpack</i></li></ul> <p><b><u>Exercice #1 (5 %)</u></b></p>
Semaine 3	<p><b><u>Notions théoriques</u></b></p> <ul style="list-style-type: none"><li>• Composants <i>React</i> dynamiques</li><li>• Intégration de données JSON</li><li>• Intégration CSS/<i>React</i></li></ul> <p><b><u>Renforcement JavaScript</u></b></p> <ul style="list-style-type: none"><li>• Méthode <i>map()</i> (programmation <i>déclarative</i> &lt;-&gt; <i>impérative</i>)</li></ul> <p><b><u>Pratique</u></b></p> <ul style="list-style-type: none"><li>• Tests sur le serveur de développement local</li><li>• Installation des modules du projet lorsqu'on collabore sur <i>GitHub</i></li></ul>
Semaine 4	<p><b><u>Notions théoriques</u></b></p> <ul style="list-style-type: none"><li>• Gestion de l'état et du cycle de vie</li><li>• Notion de "crochet" des composants <i>React</i> ("Hook")</li></ul>

	<p><b><u>Renforcement JavaScript</u></b></p> <ul style="list-style-type: none"> <li>• Différentes méthodes pour <i>cloner</i> un objet JS</li> <li>• <i>Déstructurer</i> un tableau ou un objet</li> <li>• Syntaxe d'<i>étendu de listes</i></li> </ul> <p><b><u>Pratique</u></b></p> <ul style="list-style-type: none"> <li>• Transfert d'états en amont</li> <li>• Publier une appli React monopage sur GitHub (suite)</li> </ul> <p><b><u>Exercice #2 (5 %)</u></b></p>
Semaine 5	<p><b><u>Théorique</u></b></p> <ul style="list-style-type: none"> <li>• Concept de <i>routage d'URL (routeur)</i> dans une application monopage</li> <li>• Le routeur <i>React Router</i></li> </ul> <p><b><u>Renforcement JavaScript</u></b></p> <ul style="list-style-type: none"> <li>• Les notions de <i>programmation fonctionnelle</i> et de <i>programmation déclarative</i></li> <li>• Méthodes <i>filter()</i>, <i>map()</i> et <i>reduce()</i></li> </ul> <p><b><u>Pratique</u></b></p> <ul style="list-style-type: none"> <li>• Les redirections</li> <li>• Les paramètres de routeurs</li> <li>• <i>Persistence</i> de l'état <i>React</i> : sauvegarde dans <i>localStorage</i></li> </ul>
<b>ÉTAPE #2 : Produire les effets de transitions et d'animations de l'interface utilisateur d'une application Web à l'aide du cadriceil utilisé et du langage de feuille de styles.</b>	
Semaine 6	<p><b><u>Notions théoriques</u></b></p> <ul style="list-style-type: none"> <li>• Rappel des notions de transitions CSS</li> <li>• Animations CSS simples</li> </ul> <p><b><u>Renforcement JavaScript</u></b></p> <ul style="list-style-type: none"> <li>• Les modules (<i>import/export</i>)</li> </ul> <p><b><u>Pratique</u></b></p> <ul style="list-style-type: none"> <li>• Intégration des effets d'animation et de transition dans les composants <i>React</i></li> <li>• Différentes stratégies pour inclure le CSS dans les composants <i>React</i></li> </ul>
Semaine 7	<p><b><u>Notions théoriques</u></b></p> <ul style="list-style-type: none"> <li>• Survol de quelques systèmes de design d'interface utilisateur adaptés à <i>React</i> (<i>Semantic-UI</i>, <i>Material-UI</i>, etc.)</li> <li>• Les bibliothèques d'animation de l'interface utilisateur <i>React</i> (incluant les composants de transition CSS officiels)</li> </ul> <p><b><u>Pratique</u></b></p> <ul style="list-style-type: none"> <li>• Choisir et intégrer une bibliothèque de composants d'interface utilisateur <i>React</i> (<i>Material-UI</i>)</li> <li>• Intégrer une bibliothèque d'animation <i>React</i> (<i>Framer Motion</i>)</li> </ul> <p><b><u>Exercice #3 (5 %)</u></b></p>
<b>ÉTAPE #3 : Produire l'interface de programmation applicative (API) côté serveur d'une application Web.</b>	

Semaine 8	<p><b><u>Notions théoriques</u></b></p> <ul style="list-style-type: none"> <li>• Notions d'API Web</li> <li>• Survol d'API classique de types "Rest"</li> </ul> <p><b><u>Pratique</u></b></p> <ul style="list-style-type: none"> <li>• Ouverture de compte <i>Firebase</i>, configuration de base</li> </ul> <p><b><u>Remise du TP #1 (25 %) :</u></b></p> <ul style="list-style-type: none"> <li>• Période de production en classe et remise</li> </ul>
Semaine 9	<p><b><u>Notions théoriques</u></b></p> <ul style="list-style-type: none"> <li>• Base de données <i>Firestore</i> de <i>Firebase</i></li> <li>• Les divers types d'authentification</li> <li>• Lecture et écriture des données</li> </ul> <p><b><u>Pratique</u></b></p> <ul style="list-style-type: none"> <li>• Création d'une application, préparation de l'environnement de travail local</li> <li>• Travail avec les utilitaires "Terminal" de <i>Firebase</i></li> <li>• La console de gestion <i>Firebase</i></li> </ul>
Semaine 10	<p><b><u>Notions théoriques</u></b></p> <ul style="list-style-type: none"> <li>• Les requêtes d'accès aux données</li> <li>• Les notifications</li> </ul> <p><b><u>Pratique</u></b></p> <ul style="list-style-type: none"> <li>• Utilisation de l'API native (JavaScript) de <i>Firebase</i></li> </ul> <p><b><u>Exercice #4 (5 %)</u></b></p>
<b>ÉTAPE #4 : Intégrer l'API serveur et l'interface utilisateur d'une application en appliquant les techniques de programmation client-serveur asynchrone, et en utilisant un cadriciel applicatif côté client.</b>	
Semaine 11	<p><b><u>Notions théoriques</u></b></p> <ul style="list-style-type: none"> <li>• Requêtes asynchrones avec l'API JavaScript <i>fetch</i></li> </ul> <p><b><u>Renforcement JavaScript</u></b></p> <ul style="list-style-type: none"> <li>• Les promesses (async/await)</li> </ul> <p><b><u>Pratique</u></b></p> <ul style="list-style-type: none"> <li>• Exemples de base avec <i>fetch</i></li> </ul>
Semaine 12	<p><b><u>Notions théoriques :</u></b></p> <ul style="list-style-type: none"> <li>• Paramètres de requête <i>fetch</i></li> <li>• Requêtes authentifiées</li> <li>• Gestion de l'état</li> </ul> <p><b><u>Pratique</u></b></p> <ul style="list-style-type: none"> <li>• Ajout des composants <i>Firebase</i> dans une application Web <i>React</i></li> <li>• Exploration des différents modes d'authentification proposés par <i>Firebase</i></li> </ul> <p><b><u>Exercice #5 (5 %)</u></b></p>
Semaine 13	<p><b><u>Notions théoriques</u></b></p>

	<ul style="list-style-type: none"> <li>• Intégration des données et mise à jour de l'interface (utilisation des "hooks" <i>React</i> adéquats)</li> </ul> <p><b><u>Pratique</u></b></p> <ul style="list-style-type: none"> <li>• Requêtes d'authentification</li> <li>• Requêtes de manipulation des données</li> <li>• Intégration des données <i>Firebase</i> avec <i>React</i></li> <li>• Les règles de sécurité <i>Firestore</i></li> </ul>
Semaine 14	<p><b><u>Notions théoriques</u></b></p> <ul style="list-style-type: none"> <li>• Test, profilage et optimisation d'une application réactive</li> </ul> <p><b><u>Pratique</u></b></p> <ul style="list-style-type: none"> <li>• Déploiement et test d'un site Web sur <i>Firebase</i></li> <li>• <i>React Developer Tools</i></li> </ul> <p><b><u>Exercice #6 (5 %)</u></b></p>
Semaine 15	<p><b><u>Épreuve finale (20 %)</u></b></p> <p>TP #2 : Période de production de 2 h</p>
Semaine 16	<p><b><u>Remise du TP #2 (25 %)</u></b></p>

## 5. Épreuve finale du cours

---

<b>Objectif terminal :</b>	Produire une application Web ayant une interface utilisateur réactive.
<b>Tâche ou production :</b>	Produire une application Web « monopage » en utilisant un cadriciel réactif côté client et une API Web côté serveur, et en programmant l'interactivité client-serveur en mode asynchrone.
<b>Contexte de réalisation :</b>	<ul style="list-style-type: none"><li>• À partir d'un devis de production et de médias fournis</li><li>• À partir d'un gabarit de base fourni</li><li>• Épreuve individuelle</li><li>• Sous la supervision directe d'un professeur</li><li>• En laboratoire</li><li>• Durée minimale de 3 heures en fin de session</li><li>• Documentation permise</li></ul>
<b>Critères d'évaluation et leur pondération :</b>	<ul style="list-style-type: none"><li>• Intégration complète des éléments de l'interface utilisateur requis dans le cadriciel côté client utilisé (20 %)</li><li>• Conception et implémentation adéquates des données de l'application dans le SGBD utilisé (10 %)</li><li>• Implémentation juste de l'interactivité client-serveur asynchrone (10 %)</li><li>• Programmation précise des effets d'animation de l'interface suivant le devis et en utilisant la librairie de code du cadriciel côté client utilisé (20 %)</li><li>• Programmation judicieuse des fonctionnalités en intégrant le cadriciel côté client et l'API côté serveur (15 %)</li><li>• Élaboration adéquate de la codification de façon modulaire (5 %)</li><li>• Fonctionnement correcte et optimisé de l'application réalisée (10 %)</li><li>• Test et validation correcte des fonctionnalités sur diverses plateformes (5 %)</li><li>• Qualité optimale de la programmation, vérification et débogage du code (5 %)</li></ul>
<b>Pourcentage de la note finale du cours :</b>	20 %

## 6. Matériel à se procurer

---

Aucun matériel spécifique n'est requis.

## 7. Disponibilité des enseignants

---

Les périodes de disponibilité des enseignants sont disponibles sur Omnivox.



## 8. Politiques d'évaluation des apprentissages

---

La Politique institutionnelle d'évaluation des apprentissages (PIEA) précise l'ensemble des objectifs, principes et règles relatives à l'évaluation des apprentissages de tous les étudiants inscrits au Collège de Maisonneuve. Cette politique est accessible à partir de la page d'accueil d'Omnivox dans la section « Documents de référence ».

La Politique départementale d'évaluation des apprentissages (PDEA) est complémentaire à la PIEA et précise les modalités, les particularités et les exigences départementales en regard de l'évaluation des apprentissages pour tous les cours de discipline 582. Pour consulter la PDEA du département de multimédia cliquer sur le lien suivant : <https://goo.gl/QpB728>

## 9. Médiagraphie

---

Reactjs.org. (Janvier 2024). Documentation et tutoriel disponibles en anglais, en français, et dans plusieurs autres langues. Repéré à : <https://react.dev/>

Vite. (Janvier 2024). Documentation de l'outillage (*toolchain*) React utilisé dans le cours ; disponible en anglais seulement. Repéré à : <https://vitejs.dev/guide/>

Firebase. (Janvier 2024). Documentation et tutoriel disponibles en anglais, en français, et dans plusieurs autres langues. Repéré à : <https://firebase.google.com/docs> (aussi disponible en français à la même adresse)

Banks, A., Porcello, E. (2020). *Learning React*, Sébastopol, États-Unis: O'Reilly Media.

Bevacqua, N. (2018). *Mastering Modular JavaScript*, Sébastopol, États-Unis: O'Reilly Media[i]. (Une version gratuite de ce livre est disponible à l'adresse suivante : <https://github.com/mjavascript/mastering-modular-javascript>)

Chinnathambi, K. (2017). *Creating Web Animations: Bringing Your UIs to Life*, Sébastopol, États-Unis: O'Reilly Media.

JavaScript.info. (Janvier 2023). The Modern Javascript Tutorial. Repéré à : <https://javascript.info> (en français : <https://fr.javascript.info/>)

Google Developers. (Septembre 2022). Responsive Web Design Basics. Repéré à : <https://web.dev/responsive-web-design-basics/>

DevDocs.io. (Décembre 2022). DevDocs API Documentation. Repéré à : <https://devdocs.io/>

Mozilla Developer Network. (Décembre 2022). JavaScript | MDN. Repéré à : <https://developer.mozilla.org/en-US/docs/Web/JavaScript>

Mozilla Developer Network. (Septembre 2022). CSS: Cascading Style Sheets | MDN. Repéré à : <https://developer.mozilla.org/en-US/docs/Web/CSS>

Mozilla Developer Network. (Décembre 2022). HTML: Hypertext Markup Language | MDN. Repéré à : <https://developer.mozilla.org/en-US/docs/Web/HTML>

Mozilla Developer Network. (Novembre 2022). Media queries - CSS: Cascading Style Sheets | MDN. Repéré à : [https://developer.mozilla.org/en-US/docs/Web/CSS/Media\\_Queries](https://developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries)