

Introduction

Votre travail consiste à concevoir une librairie de *Modal* accessible et simple à utiliser qui supporte toutes les spécifications et fonctionnalités demandées. Il y aura également une page Web de style documentation qui montrera comment utiliser les différentes options de votre librairie.

Le but du travail est de se concentrer sur la partie qualité du code au niveau de la conception d'une librairie. Le travail est assez « libre » puisque que vous devez vous concentrer à bien faire les choses, ce n'est pas « juste » le fonctionnement qui est requis ici, mais comment bien utiliser tous les outils du langage pour faire une librairie qui est simple à utiliser et qui peut être réutiliser pour n'importe quel site Web pour des *Modal*. Mettez-vous à la place d'un utilisateur qui veut utiliser votre librairie.

Prenons, par exemple, la librairie *Bootstrap* très complète qui vous donne accès à plusieurs fonctionnalités et composants déjà fait incluant les *Modal*. Dans votre cas, vous devez seulement faire une librairie qui offre un component de *Modal* pouvant être utilisé dans n'importe quelle page Web.

Spécifications

- **Vous n'avez le droit à aucune librairie pour faire votre librairie** à l'exception de :
 - Highlight.js (<https://github.com/highlightjs/highlight.js/>) qui peut être utiliser pour vous aider à faire votre page de documentation.
- Vous devez respecter les bonnes pratiques du développement Web (HTML, CSS, JavaScript). Vos modals devront également respecter les normes d'accessibilité : <https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA>
- Vous devrez fournir tout le code source de votre travail. Pour simplifier, vous allez remettre tout le code de votre librairie et de votre page de documentation dans le même projet. En outre, vous allez devoir remettre :
 - Une page HTML de démonstration et de documentation sur l'utilisation de votre librairie;
 - Le JavaScript et le CSS de votre page de documentation;
 - Le JavaScript et le CSS de votre librairie.
- Votre page de **Documentation** montrera toutes les fonctionnalités de votre librairie avec des exemples de code et d'utilisation de votre librairie. Elle doit être simple, efficace et fonctionner autant pour version bureau que version mobile.
 - Le UI de cette page (comme n'importe quelle UI) est important!
 - N'oubliez pas d'indiquer comme faire l'installation!
- Le CSS de votre librairie doit inclure tout pour donner le style visuel de vos *Modal* (version bureau et version mobile) incluant:
 - Toutes les classes CSS pour le visuel de vos *Modal*;
 - Toutes les classes CSS pour les animations de vos *Modal*;
 - Vous devez supporter minimalement 2 animations CSS (e.g. slide-in, fade-in).
 - Toutes les variables CSS pour personnaliser facilement vos *Modal*;
 - Le style visuel par défaut devrait être « neutre » autant que possible pour permettre la réutilisation de votre librairie dans n'importe quel site.
- Précisions sur les *Modal* :
 - Il peut y avoir plusieurs modals pour une même page. On veut aussi pouvoir réutiliser le même modal pour plusieurs boutons différents.
 - Vous devez permettre des modals avec n'importe quel contenu.
 - Vous devez permettre une façon rapide de faire modal d'alerte et de confirmation sans avoir à mettre de code HTML dans la page.
 - Vous devez permettre de mettre des valeurs dans les champs du modal à son ouverture.

Spécifications pour la librairie

- Affichage des *Modal* :
 - Le modal a la priorité et le fond de la page devient foncé;
 - Le modal devrait un style par défaut (e.g. centré horizontalement et dans le haut de la page), mais on devrait pouvoir contrôler son visuel (positionnement, dimension, animations, etc.) à l'aide de classes CSS.
 - Le modal devrait permettre n'importe quel contenu (le contenu doit s'adapter automatiquement). Si le contenu dépasse la fenêtre, il faudrait que le modal aille des barres de défilement.
- Support de différents types de *Modal* :
 - **Alert** : Affiche un *Modal* d'alerte contenant un titre, un message et un bouton « Ok »
 - Nous devons pouvoir personnaliser le titre, le message et le texte du bouton;
 - Il faut également pouvoir passer un callback qui sera exécuter sur le clic du bouton « Ok ». Le *callback* n'a pas de paramètres.
 - **Confirm** : Affiche un *Modal* de confirmation contenant un titre, un message, un bouton « Annuler » et un bouton « Ok »
 - Nous devons pouvoir personnaliser le titre, le message et le texte des boutons;
 - Il faut également pouvoir passer un callback qui sera exécuter sur le clic du bouton « Ok » ou du bouton « Annuler ». Le *callback* aura un booléen comme paramètre pour indiquer le choix de l'utilisateur (OK = true, Annuler = false).
 - **AJAX** : Affiche un modal dont le contenu sera chargé à partir d'AJAX.
 - L'implémentation est laissée à votre discrétion.
 - **Custom** : Affiche un *Modal* déjà existant dans le HTML.
- Permettre la fermeture des *Modal* :
 - Le modal devrait avoir un bouton « X » pour le fermer;
 - Le modal devrait se fermer si on appuie sur la touche « esc »;
 - Le modal devrait se fermer si on clique en dehors du modal;
 - Les boutons du modal devraient fermer le modal.
- Support des événements *custom* :
 - Le modal devrait minimalement avoir deux événements *custom* : un pour l'ouverture du modal et l'autre pour la fermeture du modal.

- Accessibilité
 - Votre librairie doit utiliser les attributs ARIA correctement (aria-hidden, etc.);
 - Vous devez conserver la position du focus avant et après l'ouverture/fermeture du modal.
 - À l'ouverture du modal, le premier champ qui peut avoir un focus devrait avoir le focus.
 - La navigation en utilisant « Tab » devrait se faire à l'intérieur du modal seulement lorsqu'il est ouvert.

Évaluation

La remise du travail sera le **10 novembre avant 15h20** dans un dépôt sous LÉA en créant une archive compressée ZIP qui contient le répertoire du projet complet (fichiers HTML, CSS et JS).

Le travail est principalement évalué sur la qualité du code et du UI. Il est toutefois obligatoire d'avoir une majorité des fonctionnalités demandées pour avoir tous les points possibles pour la qualité du code et pour le UI.

Le travail compte pour **20% de la session**.