TP 13: Composants Web

Objectifs:

- Développer un composant web simple: Accordéon / Carousel / Galerie vidéo

Slides du TP

Création d'un composant Web simple

Sur le Web, un composant est un programme qui permet de fournir une interface intégrable, conçue pour un usage précis, mais personnalisable par chaque développeur.

> Par exemple, si je développe et diffuse un composant de galerie d'images, d'autre développeurs pourront intégrer simplement mon composant sur leur site, et y ajouter les images de leur choix.

Les composants permettent donc d'enrichir le contenu, l'esthétique, et/ou les interactions proposées par un site Web, en ré-utilisant du code qui n'a pas été écrit spécifiquement pour ce site.

Il existe de très nombreux composants publiés sur Internet et utilisables gratuitement. Pour la plupart, ils sont basés sur la librairie "jQuery" (dont nous parlerons plus tard dans ce cours). Mais nombreux ont été conçus en JavaScript/DOM natif (aussi appelé "Vanilla JavaScript"), et peuvent donc fonctionner sans jQuery.

Note: Dans le cadre de ce cours, l'usage (direct ou pas) de jQuery ne sera pas accepté.

Voici quelques exemples de composants natifs intégrables librement:

- Sweet Alert
- Dialog Modal
- Animate on scroll
- Tranglify

Vous pourrez trouver d'autres composants natifs sur le site plainjs.com.

À ce stade, nous allons apprendre à réaliser un composant *simple*, dans le sens où celui-ci ne sera pas suffisamment modulaire pour être intégré plusieurs fois sur une même page. Nous verrons plus tard comment faire cela.

Exercice 1: intégrer un composant simple

Pour comprendre le principe d'usage des composants web:

1. intégrer le composant Sweet Alert (cf liste ci-dessus) sur une page web,
2. puis personnaliser son aspect à l'aide de sa documentation.

Généricité et instructions d'intégration

Afin que notre composant puisse être intégré de manière personnalisée par chaque développeur, quelque soit le site Web en question et son contenu, il est important de définir quelques règles et abstractions.

D'abord, un composant doit être simple à intégrer, et un utilisateur de composant ne devrait jamais avoir à consulter ni modifier le code source du composant. Il va falloir donc que le composant soit suffisamment générique et configurable depuis le site Web qui l'intégrera.

Par exemple, un composant de galerie d'images doit s'adapter à une liste d'images fournie par un développeur désirant l'intégrer, et l'intégrateur ne devrait en aucun cas avoir à modifier la liste d'images dans le code source du composant.

Pour cela, un composant:

- ne doit pas contenir de valeurs et références littérales; (ex: nombre d'images et/ou URLs des images stockés "en dur" dans le code du composant)
- doit définir et documenter les règles et contraintes éventuelles que devront respecter les intégrateurs du composant.

Exemples d'instructions d'intégration fournies par la documentation d'un composant:

- *toutes les images à afficher dans la galerie doivent être des balises `<img>` portant la classe `carousel-img`*;
- ou encore: *appeler la fonction creerGallerie() (définie par le composant) en passant en paramètres l'identifiant du DIV devant contenir la galerie, et un tableau d'URLS d'images*.

Prenez le temps d'analyser les instructions d'intégration des composants listés plus haut.

Exercice 2: développer un "carousel"

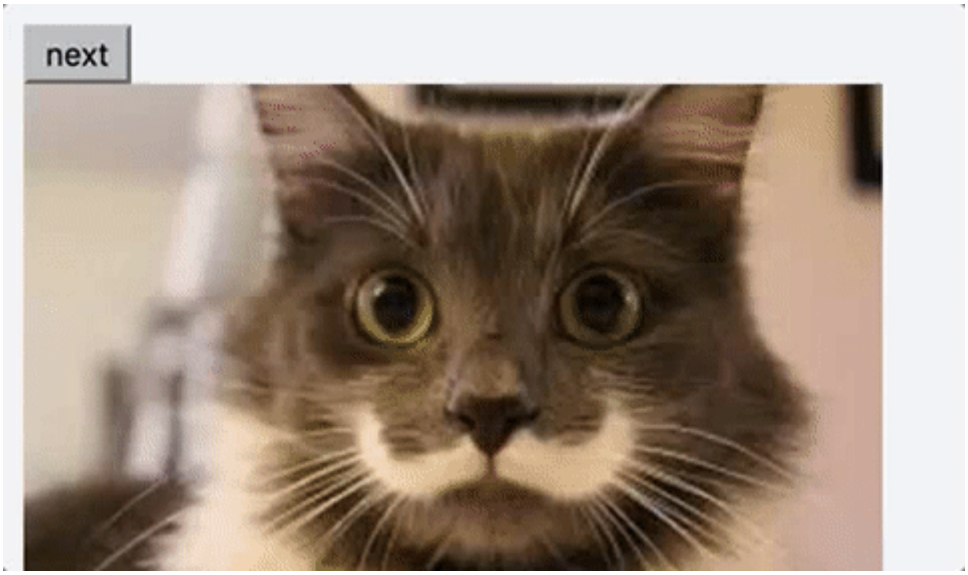
Avant de développer un composant réutilisable, il est prudent de prototyper une version d'essai de ce futur composant.

Pour prototyper notre futur composant "carousel":

1. Créer une page HTML contenant un élément `<img>` (qui contiendra l'image courante) et un `<button>` (qui servira à afficher l'image suivante).
2. Créer un tableau `images` contenant les URLs de trois images trouvées sur le web, sous forme de chaînes de caractères.
3. Toujours dans le code JavaScript associé à notre page, écrire l'instruction permettant d'afficher la première image du tableau `images` dans l'élément `<img>`.
4. Faire en sorte que l'utilisateur puisse afficher les images suivantes en cliquant sur le bouton.

Composant: Carousel

Un "carousel" est un composant de galerie d'images permettant à l'utilisateur de passer d'une image à l'autre, et éventuellement d'agrandir une image en plein-écran.



Certains carousels passent automatiquement d'une image à l'autre, à l'aide d'un minuteur (`setTimeout()` ou `setInterval()`).

Pour développer un carousel, il faut maîtriser:

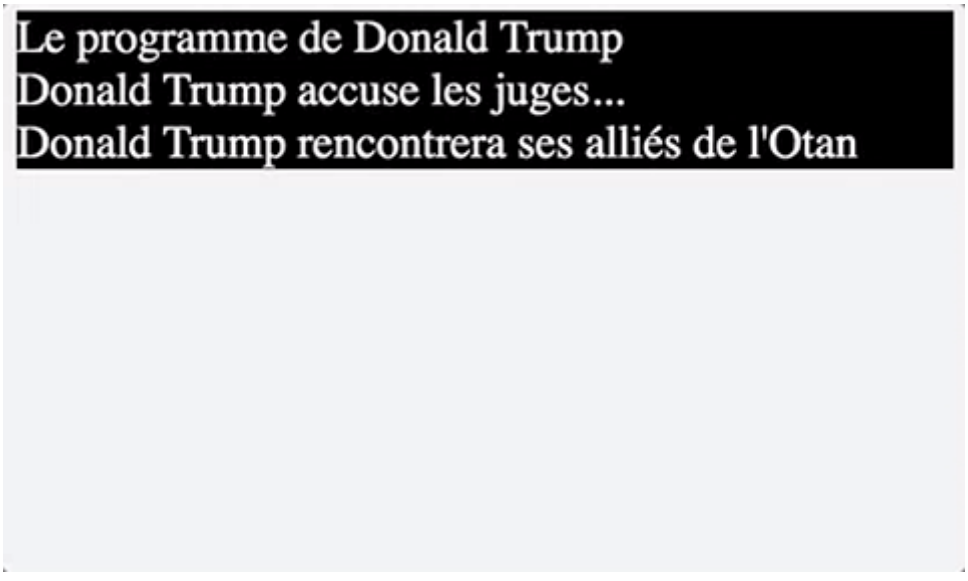
- les sélecteurs DOM (`getElementById()`, `getElementsByClassName()` et/ou `getElementsByTagName()`),
- la manipulation de styles et/ou classes CSS en JavaScript,
- la capture d'événements `onclick`,
- et l'opérateur modulo (`%`).

<!--

> Solution:
>
> - codepen
> - jsfiddle (ancienne version)
>
> -->

Composant: Accordéon

Un "accordéon" est un composant proposant plusieurs rubriques à l'utilisateur, et lui permettant d'afficher le contenu d'une rubrique à la fois, en cliquant sur son titre.

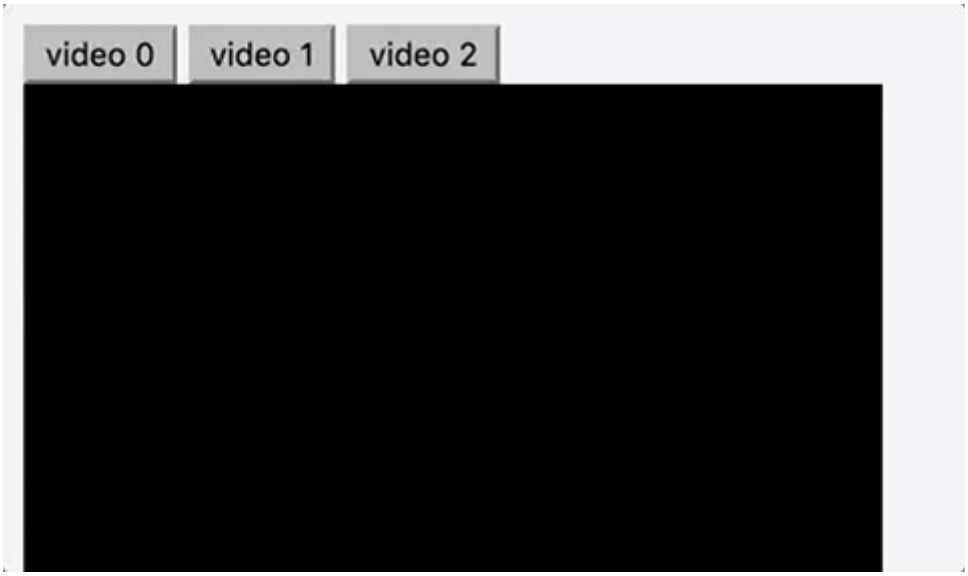


Pour développer un accordéon, il faut maîtriser:

- les sélecteurs DOM (`getElementById()`, `getElementsByClassName()` et/ou `getElementsByTagName()`),
- la manipulation de styles et/ou classes CSS en JavaScript,
- la capture d'événements `onclick`,
- et la gestion du scope.

Composant: Galerie vidéo

Une galerie vidéo permet à l'utilisateur de visualiser une vidéo, en cliquant parmi une sélection fournie.



Pour développer une galerie vidéo, il faut maîtriser:

- les sélecteurs DOM (`getElementById()`, `getElementsByClassName()` et/ou `getElementsByTagName()`),
- la capture d'événements `onclick`,
- l'intégration d'`<iframe>` (ex: à l'aide de `innerHTML`),
- et la gestion du scope.

Exercice 3: développer et documenter un composant réutilisable

Développer un des trois composants ci-dessus, de manière à ce qu'il soit facilement intégrable sur n'importe quelle page web, et fournir la page de documentation expliquant comment l'intégrer et le paramétrer.

Pour vérifier que votre composant est bien réutilisable, intégrez-en plusieurs sur une même page. Chacun des composants devra fonctionner indépendamment des autres.

La page de documentation doit contenir:

- une description concise du composant: à quoi il sert, quelles sont ses fonctionnalités et avantages éventuels;
- une démonstration du composant, pour le tester depuis la page;
- les instructions permettant d'intégrer simplement ce composant à son site;
- la description des fonctions et/ou paramètres éventuellement fournis par le composant;
- BONUS: la liste des navigateurs (et leur versions) sur lesquels le composant fonctionne;
- BONUS: le composant et sa documentation publiés sur GitHub.

S'inspirer de la documentation des composants fournis plus haut en exemple.

> Conseil pratique: pour afficher du code source sur une page HTML sans que celui-ci ne soit interprété par le navigateur, utiliser la balise `<xmp>`.

<!--

> Solutions:
>
> - Carousel réutilisable + documentation
> - Accordéon
> - Galerie
>
> -->