Laboratoire: Introduction Javascript

# Objectif(s)

* Apprendre à utiliser JavaScript
  + Obtenir des éléments
  + Ajouter des éléments
  + Modifier des éléments
  + Réagir à des évènements

# Créer la branche de la fonctionnalité

Clonage du repository

1. Dans Gitkraken, clonez le repository
   * Soit via URL
   * Soit via GitHub.com
2. Ouvrez le repository (OK)

Création d’une branche pour la fonctionnalité

1. Créez une branche appelée **FCT \_MODIFICATIONS\_DOM\_JS**
2. Assurez-vous d'être dans la branche que vous avez créée dans LOCAL

# LES BASES DE JAVASCRIPT

## GÉNÉRER DU CONTENU

ATTENTION : Ce laboratoire est très différent des autre. Pour simplifier l’apprentissage de **JavaScript**, il contient uniquement un fichier **index.html,** un fichier **style.css** et un fichier **script.js**. Vous pouvez utiliser **VS Code** ou **Visual Studio** pour travailler. C’est probablement une bonne idée de mettre vos fichiers script.js et index.html côte à côte pour travailler.

1. Ouvrez les pages **index.html** et **script.js** dans un éditeur de votre choix.
2. Ouvrez la page **index.html** dans un navigateur de votre choix.
3. **Attention, vous allez devoir recharger la page chaque fois que vous allez tester une modification.**
4. Le premier objectif c’est de pouvoir **ajouter des éléments** à cette page HTML en manipulant le **DOM** avec du code JavaScript. Vous allez ajouter les éléments à l’intérieur du div avec l’id **« demo ».**

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

Une image contenant texte, Police, capture d’écran, nombre

Description générée automatiquement

1. Vous pouvez voir sur la page qu’il y a un premier formulaire avec 3 inputs et un bouton. Lorsque vous cliquez sur le bouton, la fonction JS **genererContenu**() est appelé. Il faut modifier le code de cette fonction pour ajouter des éléments html.
2. Ajoutez un élément **h2** et un élément **p** N fois. Où N est la valeur entrée dans le champ Nombre de sections.
3. Le texte du **h2** doit être le texte de l’input Titre + son index. Donc le premier devrait afficher « Titre par défaut **0 »**, avec la valeur par défaut de l’input.
4. Après le **h2**, il faut ajouter un élément **p** avec le texte de l’input Texte. Encore une fois, ajoutez l’index au bout du texte.

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

1. Si vous avez bien travaillé vous allez obtenir avec les valeurs par défaut :

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

## MODIFIER DU CONTENU

On veut maintenant modifier la couleur du texte de nos paragraphes.

1. Lorsque vous cliquez sur le bouton du 2e formulaire, obtenez les éléments **p** que vous avez générés (et pas les autres de la page) et modifiez leur couleur pour la couleur spécifiée.

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

1. Vous pouvez regarder dans **index.html** pour voir quelle fonction JS est appelé lorsque vous cliquez sur le bouton.
2. Si vous avez réussi, vous allez obtenir le résultat suivant lorsque vous cliquez sur le bouton et avez modifié la couleur pour **blue**.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

## MODIFIER LES CLASSES

On veut maintenant modifier la class des boutons de nos formulaires

1. Lorsque vous cliquez sur le bouton du 3e formulaire, obtenez les éléments **bouton** de la page et modifiez leur **className** par celle spécifiée.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

1. Si vous avez réussi, votre page a maintenant cette apparence avec la valeur par défaut de Class. Vous pouvez également essayer « btn btn-danger » :

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

## Commentaires et validation (Commit) des changements dans le code

1. Dans Gitkraken, assurez-vous d'être dans le bon dossier/repo et sur la bonne branche
2. Cliquez sur **View changes**
3. Validez les modifications: cliquez sur **Stage all changes**
4. Commentez en respectant les standards proposés:
   * **Summary** FCT Modifications du DOM avec JavaScript
   * **Description**: Implémenter la logique de 3 formulaires pour ajouter et modifier du contenu à la page index.html
5. Cliquez sur **Commit**

## Gérer des éVènements

On veut maintenant réagir à différents évènements à l’aide de code JS.

1. **Premièrement**, vous devez modifier le code de la première fonction, **genererContenu**() pour ajouter un attribut **« monId »** avec la valeur **"Id" + l’index** de l’élément aux éléments **h2** que vous générez. Dans bien des cas, c’est très utile d’ajouter un attribut de ce genre aux éléments que l’on génère. On peut ensuite les utiliser dans d’autres scripts pour en savoir plus sur l’élément avec lequel on interagit!
2. Ensuite, vous devez ajouter une fonction à l’évènement **onmousedown** des **h2** que vous générez. Faites les choses suivantes dans cette fonction :
   1. Utilisez **console.log** pour afficher le contenu de l’event reçu par votre fonction
   2. Utilisez encore **console.log** pour afficher event.target (vous devriez voir dans l’information de l’élément h2 sur lequel vous avez cliqué)
   3. Utilisez encore une fois **console.log** pour afficher l’attribut **« monId »** de l’élément sur lequel vous avez cliqué
   4. Finalement, modifiez la couleur de backgroud de l’élément. Changez la pour « gray » sauf si elle est déjà « gray », auquel cas vous la changez pour « white ».
3. Voici le résultat attendu lorsque vous cliquez sur le premier titre :

Une image contenant texte, capture d’écran, Police, logiciel

Description générée automatiquement

On veut maintenant réagir aux évènements **onmouseleave** et **onmouseenter** et faire apparaître un élément.

1. Vous devez à nouveau modifier le code de la première fonction, genererContenu() pour ajouter un élément de type **aside** avec le texte "Plus de détails à propos de: " + i. Ajoutez le **tout de suite après le h2** (entre le **h2** et le **p**)
2. Ensuite, vous devez ajouter une fonction pour chacun des évènements onmouseleave et onmouseenter des **h2** qui va simplement ajouter la class **« show »** à l’élément **aside** sur le enter et la retirer lors du leave.
3. Vous devriez maintenant voir apparaitre la chose suivante lorsque vous faites un **onmouseenter** sur un titre :

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

1. Et simplement disparaitre avec le **onmouseleave**

## Commentaires et validation (Commit) des changements dans le code

1. Dans Gitkraken, assurez-vous d'être dans le bon dossier/repo et sur la bonne branche
2. Cliquez sur **View changes**
3. Validez les modifications: cliquez sur **Stage all changes**
4. Commentez en respectant les standards proposés:
   * **Summary** FCT Gestion d’évènements avec JavaScript
   * **Description**: Implémenter la logique en JS pour réagir aux évènements onmousedown, onmouseover et onmouseenter sur les éléments h2 générés
5. Cliquez sur **Commit**
6. Une fois que tout est terminé, n’oubliez pas de faire un push.
7. Merge la branche **FCT \_MODIFICATIONS\_DOM\_JS** dans la branche **Main** et faites un **dernier push.**

**Fin du laboratoire!**