Note Challenge COll dev-front

**Jeudi 28 juin 2018 :**

Création d’un git hub : <https://github.com/fvince25/JJVL_wysiwyg>

Pour le git :

-Git clone pour récupérer le lien du repository (bouton vert)

**Ajouter un nouveau fichier :**

-Git status pour vérifier les fichiers de notre dossier que nous avons modifié

-Git add

-Git Commit

-Git pull

-Git push

Test sur le Readme ! A refaire car ne fonctionne pas.

***Info sur challenge :***

**Ne pas se servir du text area (il ne permet pas de faire des mises en forme), chercher du côté de HTML5 « comment on rend une balise éditable ».**

**Objet JS à trouver sur le net avec ses méthodes (position curseur, sélection de l’utilisateur,)**

**A faire un Plugin Jquery, pour cela il faut aller chercher un tuto pour en créer un**

**A créer : une interface, chercher une doc sur l’objet JS (2 pers), création du plugin Js, aller chercher le color Picker,**

*Trouver un nom d’équipe, animation équipe,*

Répartition des premières tâches via le planning gestion de projet

Utilisation de <https://glyphicons.bootstrapcheatsheets.com/#> et <https://fontawesome.com/> pour les icons.

NB : créer en amont les fichiers html scss et js et créer un dossier css ou on class le dossier scss. Intégrer le fichier plugin jquery

SaSS :

Création d’un dossier css aller ds la console, aller dans le dossier de git hub avec cd

1 : npm init ; juste saisir (gulp file.js) « dans entry point » puis saisir ok

un nouveau fichier package.json se crée dans le dossier

2 : récupérer le fichier gulpfile.js déjà existant dans un ancien dossier refaire les npm installs des nouveaux plugins du gulpfile.js

3 : installer Gulp dans le dossier projet avec **npm install –save–dev gulp**

rajoute un nouveau dossier node.module ainsi qu’un nouvel élément dans le package.json dans le devDependencies.

4 : créer une task ds le fichier gulpfile pour tester que cela fonctione. Puis faire console log du nom de la tache dans la console.

5 : pour installer SaSS : ds la console faire **npm install –save-dev sass**

6 : vérifier ds le fichier package Json qu’il a rajouter l’élément sass dans devdependencies

7 : Créer une variable qui contient le plugin (var sass = require(‘sass’) et créer la task « sass : watch » et « sass »

aller chercher en src le fichier css/style.scss que l’on a créé. (voir ancien exemple)

Idée pour la partie texte éditable (Vincent) :

Attribu content éditable qui permet à une div de se comporter comme un text area

Création d’une Div en html attribut contentditable <div id= ‘textarea’ contenteditable>

Pour interagir avec le contenu, sur l’objet Window il existe la méthode Get selection

Cela affiche ce que tu sélectionne on met toString() permet de changer un objet js en élément string. Donc on n’affiche que la partie text de l’objet js.

En JS on met un addListener au get élémentByid et lui rajoute l’événement mouseup qui permet de récupérer le contenu sélectionner.

Finir la fonction par ,false (notion de bouillonnement : <https://openclassrooms.com/courses/1916641-dynamisez-vos-sites-web-avec-javascript/1918968-les-evenements)>