Note du Challenge collectif des 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 répository (bouton vert)

**Ajouter un nouveau fichier :**

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

-Git add espace nom complet du fichier ex : README.md

-Git Commit –m « message » si nouveau fichier // Git commit –a –m « message » si mise à jour d’un fichier déjà présent

-Git pull

-Git push –u origin master ou origin (nom de la branche) (préférer la deuxième option)

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

***Info sur challenge donné par Robin :***

**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,**

*Reste à faire pour l’équipe : Trouver un nom, assurer la bonne communication et cohésion.*

**ACTIONS REALISEES CE JOUR :**

Répartition des premières tâches via le planning gestion de projet

Utilisation de [https://glyphicons.bootstrapcheatsheets.com/#](https://glyphicons.bootstrapcheatsheets.com/) et <https://fontawesome.com/> pour les icons. Pb avec le glyphicons boostraps

Création en amont des fichiers htm,l scss et js et un dossier css où on classe le dossier scss. Intégrer le dossier le plugin jquery

Installation et utilisation de SaSS :

Se référer à ce site pour la procédure : https://la-cascade.io/gulp-pour-les-debutants/

Idée pour la partie texte éditable (Vincent) :

Attribut contentEditable 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, voir ce que cela signifie ici : <https://openclassrooms.com/courses/1916641-dynamisez-vos-sites-web-avec-javascript/1918968-les-evenements> )

**[NEXT](#VEND29JUIN)**

**VENDREDI 29 JUIN 2018**

Propositions pour démarrer la journée :

-S’assurer que tous ont bien Gulp, saass, Node.modules et accès au Git

-Faire un Débrief des étapes réalisées hier :

Consultation de l’ébauche de Jean,

Retour de Vincent Julie et Lionel sur leur recherche

-Un point sur les prochaines répartitions des tâches pour la matinée (cf planning de gestion de projet)

**ACTIONS REALISEES CE JOUR :**

Test de pusher les fichiers sur nos branches via Git hub

Draft read me

Test et intégration de color picker