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 readme

Style.css de la page

Code pour la partie contentEditable

Proppo pour les icon :

Certains icons comme bold on utilise une class

Pour le h1 il nous faut plusieurs attributs (donc on fait un data-cmd sur les li pour affecter un attribut)

Aller voir des infos sur le repo ignore sur google,

Sources :

<https://developer.mozilla.org/fr/docs/Web/API/Document/execCommand>

Recherche de Lionel sur le Get Ranch et le content Editable ainsi que sur le WYSIWIG et [www.tinymce.com](http://www.tinymce.com) (lien vers les recherches [**ici**](Senseo.doc.Jquery.docx)

Vincent grâce à Jean a pu tester et approuvé la fonction execCommand

Vincent a bien avancé sur le js de la partie « texte area » ainsi que sur les boutons.

« Petit point de situation de mon avancement :

J'ai pu tester toutes les fonctionalités sauf le heading h1 h2 h3 ...

J'ai pu intégrer le colorpicker (merci pour vos recherches Julie / Lionnel)

j'essaierai de voir ve qui ne va pas demain dans les headings

Ah non il y a encore l'affichage du mode "Text", permettant de voir les tags html. Mais je vois a peu prêt comment faire. Je ne suis pas trop inquiet

Pour vous expliquer un peu le fonctionnement :

J'ai indiqué à Jean de mettre sur ses boutons la classe correspondant exactement au nom de la commande "execCommand" qui est néccéssaire

Tous ces boutons contiennent également la classe icone. du coup le simple code suivant permet d'activer la fonctionnalité voulue :

$(".icone").on("click", function() {

var action = $(this).attr("class").split(" ")[1];

applyStyle(action);

});

var action = $(this).attr("class").split(" ")[1] permet de récupérer la deuxième classe

function applyStyle(style) {

var val = "";

if (style == "createLink")

val = prompt('Saississez un lien :');

if (style == "insertImage")

val = prompt('Saississez une url d\'image:');

// document.execCommand(style);

document.execCommand(style, false,val);

}

Un petit rappel sur l’utilisation de Git Hub par ici proposé par Vincent :

<https://www.youtube.com/watch?v=V6Zo68uQPqE&t=1372s>