Correction

*-Quel élément HTML permet d'intégrer du JavaScript dans une page ?*

<script>

*-Quelle est la syntaxe correcte pour référencer un script externe appelé monScript.js dans une page ?*

src="monScript.js"

*-Que renvoie l'exécution du code : message="bonjour" ; alert(typeof message) ; ?*

string // typeof donne le type d'une variable

*-Qu'affiche le code : alert("10"+"5") ; ?*

105 // concaténation de deux string

*-En JavaScript, comment exécuter une fonction déjà crée qui s'appelle maFonction ?*

MaFonction() ;

*-En JavaScript, quelle propriété me permet de modifier l'élément HTML bonjour ?*

document.getElementsByTagName("p").innerHTML

*-En JavaScript, comment modifier la couleur de fond de l'élément HTML initialisé par elem = document.getElementById("myElement")* ?

elem.style.backgroundColor="blue" ;

*-Où insère-t-on du JavaScript (code ou lien) dans une page ?*

Les scripts javascript doivent être placés dans un fichier externe à la page HTML. Habituellement on place la balise d'insertion des fichiers dans le <head> . Pour une question de performances on préférera la placer à la fin de la page, juste avant </body> À noter que les deux méthodes sont valide du point de vue de la spécification HTML et XHTML.

*-Que font les commandes Git suivantes :*

$git status
Affiche l'état des fichiers.

$git branch
La commande git branch fait plus que créer et effacer des branches. Si vous la lancez sans argument, vous obtenez la liste des branches courantes. « * » vous indique la branche qui est actuellement extraite.

$ git checkout nom_de_ma_branch
Change de branche.

$ git add .
$ git commit - m "initial commit"

Fait un premier commit.
Push pour ensuite le commit vers le dépôt distant.

$ git reset --hard md5_commit
$ git push –force

Si vous vous êtes embrouillé dans votre répertoire de travail, mais que vous n’avez pas encore committé vos erreurs, vous pouvez retrouver l’état dans lequel était votre répertoire après le dernier commit en utilisant :
$ git reset --hard md5_commit
Cela effacera toutes les modifications que vous avez ajouté à l’index git ainsi que les changements qui sont présents dans votre répertoire de travail mais qui n’ont pas été ajoutés à l’index. En d’autres termes, après cette commande, le résultat de git diff et git diff --cached sera vide.
Il est important de noter que cette option (--hard) est le seul moyen de rendre la commande reset dangereuse et est un des très rares cas où Git va réellement détruire de la donnée. Toute autre invocation de reset peut être défaite, mais l’option --hard ne le permet pas, car elle force l’écrasement des fichiers dans le répertoire de travail.
La commande push permet d'envoyer tous les commits d'une ou plusieurs branche au dépôt distant. Git ne permet pas de push si le dépôt distant est en décalage (pas de fast-forward possible) et dans ce cas là vous pouvez utiliser le drapreau --force
git push --force
Ce drapeau doit être utilisé en dernier recours (normalement vous n'en aurez jamais besoin) car il va modifier l'historique distant et peu ainsi affecter tous les collaborateurs. Mais ça peut être utile en cas de problème, si par exemple vous envoyez un commit en oubliant de retirer une clef d'API ou une information sensible.

*Avec flexbox, mettez l'élément au centre du container:*

```
#element {
  display: flex;

justify-content : center ; // on centre sur l'axe horizontale

align-items : center ; // on centre sur l'axe verticale

}
```

*Soit notre élément avec column comme flex direction, faut-il utiliser justify-content ou align-items pour déplacer notre élément horizontalement ?*

Justify-content permet de place notre élément dans l'axe de la direction (row par défaut, donc horizontalement), de fait si la direction est en colonne, justify-content déplacera notre élément verticalement. On utilisera donc align-items.