#10 : Amusettes cryptographiques

1) Objectifs

Arrivé en fin de formation initiale, on se trouve démuni quand il s'agit d'élaborer un programme car on ne maîtrise pas la **modélisation de données**. C'est-à-dire **comment choisir les variables à créer et comment les utiliser**.

Cette fiche d'exercices a pour but de travailler cette notion qui s'acquièrt par l'expérience. Nous allons principalement travailler autour de deux grands types de variables : les **chaînes de caractères** et les **tableaux**.

2) Préparer l'interface

- ❑ Notre programme se présentera sous forme d'une page web dans laquelle on proposera **2 boites de saisie de texte** (Message à coder et Clef de cryptage) et d'un **bouton** pour lancer le codage et afficher le résultat.
- ❑ Recherchez sur internet (cours #7) comment on peut créer des boites de saisie de texte et récupérer leur contenu en JavaScript.
- ❑ Créez une page HTML standard nommée **cesar.html** qui lance le programme JavaScript **cesar.js** et dans laquelle vous ajoutez **les 2 boites de saisie** et **le bouton « codage ».**
- ❑ Faites en sorte que lorsqu'on clique sur le bouton, une « alert » affiche le contenu des deux boites de saisie.

3) Code de César

- ❑ Recherchez sur internet comment fonctionne le code de césar.
- ❑ Pour vérifiez que vous avez compris codez le texte suivant avec la clef 5 : « Tu ne dois pas lire ça »

_ _

_ _

- ❑ Nous allons devoir résoudre 2 problèmes classiques liés aux types des données : premièrement, nous avons besoin de la clef sous forme de **nombre** alors que pour l'instant c'est une **chaîne de caractères**. Cherchez sur internet comment convertir une « **string** » en « **int** ». Mettez en œuvre dans votre programme cette conversion et vérifiez que la nouvelle variable **clefNum** est bien un **entier** (là encore cherchez sur internet).
- ❑ Deuxième problème, nous allons devoir **traiter tous les caractères 1 par 1** pour les décaler de la valeur de la clef. Cherchez sur internet comment accéder à un caractère particulier d'une chaîne. Et testez dans votre programme que vous arrivez à récupérer la $5^{ème}$ lettre du message à coder.
- ❑ Dernier problème, pour décaler les lettres nous allons modifier leur **code**. En effet, en informatique chaque lettre est codée par un nombre. C'est le fameux tableau **ASCII** qui laisse maintenant sa place à l'**UNICODE**. Recherchez sur internet le fonctionnement du tableau ASCII et faite en sorte que votre programme affiche une chaîne de caractères composée des 3 caractères qui ont les codes (en décimal) **65 72 33**.
- ❑ Vous avez maintenant tous les éléments en main pour finaliser le programme qui va encoder le message saisi avec la clef saisie quand on clique sur le bouton !
- ❑ Appelez le prof pour crâner un peu quand ça marche !
- ❑ Expliquez pourquoi ce code est très facile à casser :

_ _

_ _

4) Décodage

- ❑ Ajoutez un bouton décodage qui décode le message tapé dans la boîte de saisie du message avec la clef.
- ❑ Programmez le décodage et appelez le prof pour crâner encore plus quand ça marche !

5) Enigma polyalphabétique...

- ❑ Pour se mettre à l'abri de l'analyse des fréquences des lettres il faut utiliser un codage polyalphabétique, c'est-à-dire qu'une lettre du message à coder (par ex : A) pourra se retrouver codée par différentes lettres en fonction de l'état du système (par ex : G, Z ou P). C'est le système mis en œuvre dans la fameuse machine Enigma.
- ❑ Voyons comment procéder avec la clef : « coder »

Message	p	r	o	g	r	a	m	m	e
Clef	c	o	d	e	r	c	o	d	e
Résultat	S	G	S	L	J	D	B	Q	J

- Premier constat, c'est bien une substitution polyalphabétique car la lettre « r » est codée « G » une fois et « J » une autre fois.
- Il suffit donc de recopier en boucle la clef sous le message et d'appliquer une opération de codage entre chaque lettre du message et la lettre de la clef qui lui est associée.
- Ici, j'ai choisi de décaler la lettre du message par le rang de la lettre de la clef dans l'alphabet. Explication pour la 1ère lettre à coder : le « c » de la clef est la 3ème lettre de l'alphabet, je décale donc le « p » à coder de 3 lettres, ce qui donne « s ».
- Copiez la page HTML précédente et renommez la **polycode.html**. Elle doit lancer le programme JavaScript **polycode.js** et contenir **les 2 boites de saisie** et **le bouton « codage ».**
- Écrivez une fonction qui retourne un nombre de décalage si on lui donne en argument une lettre de l'alphabet : **calculeDecalage(lettre)**. Si j'appelle cette fonction calculeDecalage("c") elle doit renvoyer le nombre 3.
- Écrivez une fonction qui retourne la lettre codée si on lui donne la lettre du message et le nombre de décalage : **codeLettre(lettreMsg, decalage)**. Si j'appelle cette fonction codeLettre("p", 3) elle doit me retourner la lettre « s ».
- Écrivez une boucle qui parcourt chaque lettre du message et qui lui associe la lettre de la clef. Vérifiez votre boucle en faisant un alert à chaque tour pour afficher le couple des 2 lettres.
- Finalisez le programme qui code le message avec la clef et qui affiche le résultat.

6) Turing…

- Programmez le décodage et appelez le prof pour le scotcher quand ça marche !

7) Améliorez la clef…

- Pour que la clef change automatiquement tous les jours on peut en générer une partie avec des éléments qui changent, par exemple la date. Par exemple : « maclefdecodage**jeudideux** ». Cette astuce est utilisée notamment par le système de validation en deux étape les « Authentificateurs » (google, steam, etc.).
- On peut aussi y ajouter des chiffres, ce qui ajoute des possibilités et donc rend la clef plus difficile à casser. Par exemple : « maclef13du**jeudi02032017** ».
- Améliorez votre programme pour qu'il ajoute automatiquement un élément de la date à la clef et vérifiez le codage et le décodage. Montrer votre œuvre au prof.

8) Algorithme de cryptage : XOR, dans le ciel …

- L'opération de codage par décalage de l'alphabet est trop simple, on peut donc la remplacer par un autre algorithme de cryptage moins simpliste. Pour cela on peut imaginer n'importe quelles opérations (mathématiques, logiques, informatiques, etc.) qui vont transformer le message avec la clef pour obtenir le message codé.
- Je vous propose d'essayer d'appliquer une opération logique XOR entre le code de la lettre du message et le code de la lettre de la clef. Ceci produira un message codé. Pour décoder ce message il suffira d'appliquer à nouveau la même opération logique car elle permet de retrouver le message initial si on l'applique 2 fois.
- Recherchez sur internet comment appliquer un XOR entre deux nombres en JavaScript.
- Implémentez cette opération dans votre algorithme de cryptage en faisant en sorte que les codes obtenus soient affichables.
- Vérifiez que le codage fonctionne et appliquez à nouveau le codage sur le message codé pour revenir au message initial… Est-ce que ça marche ???

9) L'avenir t'appartient !

- Imaginez un algorithme de cryptage et implémentez le ! Montrez votre œuvre au prof !!!