

<http://irem.univ-reunion.fr/spip.php?article382>



TP en JavaScript sans CaRMetal

- Algorithmique et programmation
- TP d'algorithmique en Seconde

Publication date: vendredi 28 mai 2010

Copyright © IREM de la Réunion - Tous droits réservés

Le sujet du TP était le suivant : Écrire un fichier html contenant une [table trigonométrique](#) au degré près (seuls les cosinus étaient demandés). La compétence testée était la capacité que les élèves aient éventuellement acquise, de faire du JavaScript directement dans un fichier [html](#).

Faire du JavaScript sans la facilité de programmation de [CaRMetal](#) (clic sur les instructions, apparition du message d'erreur en bas de la fenêtre de l'éditeur) c'est un peu comme faire du vélo sans les petites roulettes de chaque côté : La première fois, on a une certaine appréhension ! La présence d'une console d'erreur dans le navigateur [Firefox](#) a donc fait choisir ce dernier.

Pour l'éditeur de texte,

il en fallait un qui

- offre la coloration syntaxique ;
- gère facilement le format html ;
- ne soit pas installé mais seulement placé (provisoirement) dans le dossier de travail des élèves ;
- tourne sous Windows XP ;
- et si possible, soit libre !

Le choix s'est donc porté sur [Notepad++](#), placé sous license [GNU GPL](#).

Préparation

Après avoir lancé le logiciel, les élèves ont eu à le configurer, tout d'abord en le mettant en Français (dans « Préférences ») :

[<http://irem.univ-reunion.fr/local/cache-vignettes/L400xH230/np1-073fb.png>]

puis en mettant son langage en [html](#) (pour cela il a fallu qu'ils pensent que « html » commençant par un « h », il fallait le chercher à la lettre « h »...) :

[<http://irem.univ-reunion.fr/local/cache-vignettes/L400xH357/np2-ec5b9.png>]

Enfin il a fallu sauvegarder le fichier vierge sous le nom **trigo.html**, en effet

- Seuls les fichiers enregistrés sont testables par *Mozilla Firefox*,
- Si le nom du fichier contient un espace, le fichier n'est pas ouvert correctement.

Ces manip, relativement rapides, ont permis de valider l'item 1.4 du B2i !

Il ne restait alors plus qu'à taper le code html (langage qu'*a priori*, les élèves ne connaissaient pas du tout).

Gabarit

Pour les balises *html* indispensables, un gabarit a été distribué sous forme papier (donner un fichier html tout fait aurait fait gagner du temps mais aurait privé les élèves de la satisfaction d'avoir créé leur propre fichier html). Une fois recopié dans *Notepad++*, on obtient ceci :

[<http://irem.univ-reunion.fr/local/cache-vignettes/L400xH244/np3-12db2.png>]

(la première ligne n'est pas indispensable, elle annonce une volonté de se plier aux revendications du [w3c](#), visibles sur les lignes 6 et 10 ; le corps (Â« body Â») est vide parce que le script a été placé dans la tête (Â« head Â») pour s'exécuter le plus tôt possible).

Il ne reste maintenant plus qu'à se lancer en JavaScript, mais cette fois-ci de mémoire, sans autre aide que celle consultée sur le web par certains élèves.

Pour commencer, une boucle affichant les 10 premiers entiers a été placée dans la zone réservée au script. Comme les fonctionnalités de [CaRMetal](#) ne sont plus là, en particulier le *Println* tant utilisé dans l'année, il faut maintenant utiliser *document.write(i)* pour afficher *i*.

Premier essai

La boucle d'affichage des 10 premiers entiers ressemble à ceci :

[<http://irem.univ-reunion.fr/local/cache-vignettes/L310xH114/np4-d8ed0.png>]

Mais elle imprime

12345678910

sous *Firefox* ; en effet le passage à la ligne doit se coder en html par une balise Â« br Â», écrite en JavaScript sous

forme d'une [chaîne de caractères](#) :

[<http://irem.univ-reunion.fr/local/cache-vignettes/L300xH148/np5-de46e.png>]

Avec ça on a bien les entiers l'un sous l'autre :

1
2
3
4
5
6
7
8
9
10

Tableau simple

L'expérience des élèves a alors rapidement mené à ce tableau :

[<http://irem.univ-reunion.fr/local/cache-vignettes/L312xH218/np6-32bd4.png>]

Seulement les cosinus ne sont pas bons. Plusieurs élèves ont alors dit "Bien sûr ! JavaScript calcule toujours les [cosinus](#) en [radians](#) ! Ce qui a naturellement amené à la question autour de laquelle était tacitement articulé tout ce TP :

Au fait, comment est-ce qu'on convertit les angles des degrés en radians ?

Les élèves qui savaient comment faire (il y en avait tout de même quelques-uns) avaient utilisé

- soit leur connaissance du cours,
- soit Internet ; mais dans ce cas, ils ne savaient pas comment traduire les formules trouvées en JavaScript ; selon toute vraisemblance, il est plus facile d'interpréter une phrase comme « on multiplie par $\frac{\pi}{180}$ »

qu'une formule...

Le script finalement obtenu par ce travail collectif des élèves est le suivant :

[<http://irem.univ-reunion.fr/local/cache-vignettes/L383xH216/np7-ee8f8.png>]

et en « live » :

```
Table trigo <!-- for(t=0;t<=90;t++){ document.write("|"+t+"|"+Math.cos(t*Math.PI/180)+"|"); } -->
```

Le résultat est améliorable : Les cosinus de 60° et de 90° sont visiblement mal arrondis par JavaScript. Et un tableau fabriqué en html, c'est tout de même mieux que ce bricolage avec des traits verticaux.

Tableau amélioré

Pour améliorer l'aspect du tableau, il faut écrire des balises html « table » pour le tableau, « tr » pour les lignes (« row ») du tableau, « th » pour les entêtes du tableau, « td » pour les cases. Ce qui complique nettement les choses :

[<http://irem.univ-reunion.fr/local/cache-vignettes/L400xH141/np8-64de4.png>]

D'ailleurs les élèves n'ont pas eu le temps de compléter cette partie.

Pour afficher mieux les cosinus (et mieux arrondis) il faut

1. les multiplier par 1000 (par exemple),
2. arrondir à l'entier le plus proche, puis
3. diviser le résultat par 1000

(tiens, un *algorithme* !)

Combinée à l'amélioration précédente, celle-ci donne

[<http://irem.univ-reunion.fr/local/cache-vignettes/L400xH232/np9-6bd08.png>]

et

en Â« live Â» :

Angles	Cosinus
"); document.write(i); document.write(")	"); document.write(Math.round(1000000*Math.cos(i*Math.PI/180))/1000000); document.write(")

"); --Â»

Conclusions :

- Faire du html à petites doses, c'est possible en classe de Seconde ; certes, les TPs précédents ont énormément aidé à ceci !
- Faire du JavaScript sans CaRMetal est tout de même plus difficile qu'avec, même avec la console d'erreur (certaines erreurs ne sont pas signalées par la console de Firefox et longues à détecter car peu visibles ; cas typiques :
 - le remplacement d'un point-virgule par une virgule
 - l'ajout d'un espace entre la parenthèse fermante et l'accolade ouvrante)
- L'existence d'élèves déjà au fait du langage html, et même de l'outil, soupçonnée, fut confirmée. Cela occasionne une inversion du sens de transfert des connaissances (pour une fois, c'est le prof qui demande aux élèves de lui expliquer comment ça marche)
- L'attention de **tous** les élèves a été soutenue pendant **toute l'heure** : Difficile d'imaginer que le TP n'ait pas été jugé intéressant !
- Outre les révisions de trigonométrie, ce TP a été le prétexte à quelques éclaircissements historiques sur le sujet : Écriture de la plus vieille table trigonométrique connue par [Âryabhata](#), avec un pas de 3,75°, réalisation de la première table au degré près par l'équipe d'[Al Kashi](#) à [Samarkande](#)... Pour en savoir plus, [voir ici](#).