

Il y a cinq ans les tfinaghges étaient codés dans Unicode, que de chemin parcouru !

Patrick Andries

Conseils Hapax, Québec, Canada
Membre du consortium Unicode
patrick@hapax.qc.ca

Résumé. Lors de cette communication, nous nous pencherons sur le chemin parcouru depuis le codage des caractères tfinaghges dans l'ISO 10646 et Unicode en 2005. Il y a cinq ans, il était impossible d'envoyer des documents en tfinaghges sans se référer à un codage de police non normalisé. Aujourd'hui, on peut créer des pages HTML, XML en tfinaghges, envoyer des courriels. Il existe un clavier normalisé pour saisir des textes tfinaghges, une norme de tri, Microsoft fournit une police qui prend en charge les tfinaghges dans Windows 7. L'utilisateur peut désormais voir des pages HTML sans qu'il n'ait à explicitement installer de polices tfinaghges sur son système. Des bibliothèques logicielles comme ICU prennent également en charge les tfinaghges et il est possible, en théorie, d'avoir des noms de domaine Internet en tfinaghges.

Mots-clés. Unicode, tfinaghe, ISO 10646, amazighe, informatique, réseaux, jeux de caractères, normalisation, polices embarquées, polices incorporées, XML, noms de domaine, IDNA 2008, NDI, courriel, messagerie, touareg, navigateurs, internet, WOFF, Foucauld, biconsonnes, ligatures, diacritiques.

1 Introduction

Le 31 mars 2005, Unicode 4.1 était publié. Il comprenait tous les tfinaghges normalisés de l'IRCAM ainsi que les principaux caractères touaregs et kabyles.

Avant cette date, il n'existait aucune façon de coder de manière normalisée des textes tfinaghges. Chaque producteur utilisait un codage, le plus souvent de son cru, lié à une police particulière. Il était donc virtuellement impossible d'échanger des documents produits par des personnes qui utilisaient des polices tfinaghges différentes. C'était vrai

pour les courriels, les documents Word, les pages HTML en général et, bien sûr, celles d'une encyclopédie en ligne naissante comme Wikipédia.

Depuis cinq ans bien des choses ont changé : les tifinaghes normalisés sont désormais de plus en plus présents sur Internet et dans d'autres produits et normes informatiques. Nous allons brièvement passer en revue ci-dessous ces heureuses améliorations.

2 Les caractères Unicode

Le lecteur de cette communication connaît certainement les premiers caractères tifinaghes qui ont été codés dans Unicode 4.1 en mars 2005, car ils ont fait l'objet de plusieurs communications de l'IRCAM et de ses chercheurs¹. Il ignore peut-être certaines précisions apportées depuis 2005 qui se retrouveront dans la version 6.0 d'Unicode à paraître à la fin 2010.


2.1 Les diacritiques

Plusieurs variantes tifinaghes modernes utilisent des diacritiques pour compléter les lettres du bloc tifinaghe. C'est ainsi que la notation Hawad utilise des diacritiques du bloc [U+0300-U+036F]. Ces signes s'utilisent pour représenter des voyelles ou des consonnes étrangères. Dans cette notation, <U+2D35, U+0307> $\dot{\text{X}}$ représente un « a » court, <U+2D49, U+0304> $\bar{\text{X}}$ un « i » long /i:/ et <U+2D31, U+0302> $\hat{\text{X}}$ permet d'écrire un « p ». On indique certaines voyelles longues à l'aide de deux signes diacritiques, un « é » long /e:/ s'écrit <U+2D49, U+0307, U+0304> $\bar{\text{X}}\dot{\text{X}}$. Ces signes sont affichés côte à côte, et non empilés, au-dessus de la lettre de base dans l'ordre d'apparition dans la chaîne codée.

2.2 Deux caractères ajoutés dans Unicode 6.0

Deux caractères tifinaghes ont été ajoutés dans Unicode 6.0 (et bien sûr dans la version correspondante de l'ISO/CEI 10646) :

- U+2D70 \int SÉPARATEUR TIFINAGHE
= tazaraste

- U+2D7F  LIANT DE CONSONNES TIFINAGHE

¹ Voir, par exemple, Lahbib ZENKOUAR, *L'Écriture amazighe tifinagh et Unicode*, revue *Études et Documents Berbères* vol. 22, 2004, pp. 175-173.

Le caractère U+2D70 ⵓ est ce signe targui dont Prasse dit² « Au Hoggar on nous a donné le séparateur ⵓ , à l'intérieur duquel s'écrivait la dernière lettre de chaque mot phonétique. »

Les deux figures ci-dessous illustrent l'utilisation de ce séparateur. On remarquera que le signe est réfléchi quand il s'écrit dans un contexte de droite à gauche.

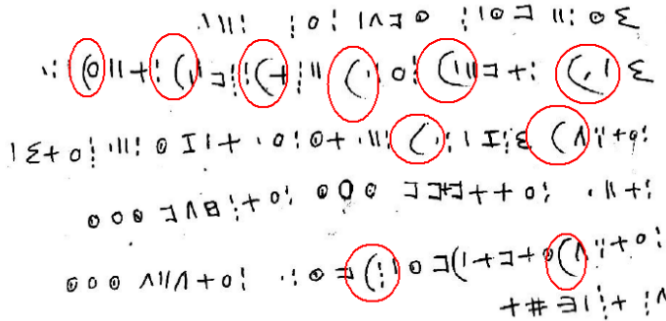


Figure 1 – Exemple d'utilisation du séparateur de mot, Sud de l'Algérie

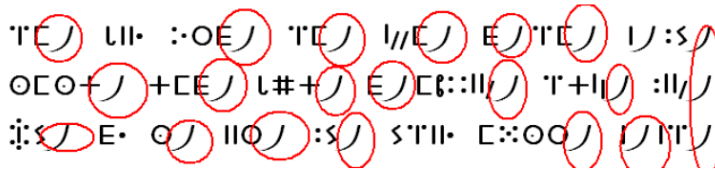



Figure 2 – Utilisation au Niger dans un contexte gauche à droite

Le *liant de consonnes* U+2D7F ⵟ est un caractère de commande qui permet de forcer la formation d'une biconsonne. Nous étudierons son utilisation dans la section suivante.



2.3 Ligatures

Dès le codage des tfinaghges dans Unicode 4.1, il était possible de préciser que l'on préférerait obtenir une biconsonne (ou une triconsonne) en insérant un U+200D ⵓ *liant sans chasse* entre les consonnes qui constituent la ligature souhaitée. On peut également utiliser U+200C *antiliant sans chasse* entre deux consonnes pour décourager la formation d'une ligature à partir de ces deux lettres.

² Karl-G. PRASSE, Manuel de Grammaire touarègue, Éditions de l'Université de Copenhague, 1972, p. 152.

C'est ainsi que l'on peut demander la formation de la biconsonne « rt » en écrivant <U+2D54, U+U200D, U+2D5C> (ⵔ  ⵔ). Les polices pourvues d'une telle ligature devraient en présence de ces caractères afficher le glyphe de celle-ci, par exemple ⊕, ⊞, +ⵔ, ou +□ selon la police. Il existe également quelques cas de triconsonnes : parmi celles-ci notons le groupe consonantique « nft » |ⵏⵑⵔ dont la ligature est parfois ⵏⵑⵔ et « knk » |ⵏⵑⵏ représenté dans certaines régions par ⵏⵑⵏ. Pour bien fixer les idées, on demandera la formation optionnelle de la ligature « nkn » à l'aide de la suite suivante de caractères : <U+2D4F, U+200D, U+2D3E, U+200D, U+2D4F>.

Certaines polices pourront être dépourvues de ligatures, d'autres n'en inclure que pour certaines variantes géographiques. Si une police venait à ne pas avoir de ligature correspondant à la suite de caractères liée par un liant sans chasse, la police devrait simplement afficher les deux consonnes de base, à savoir ⵔⵔ dans notre exemple ci-dessus.




Toutefois, aux yeux d'aucuns, il est apparu que le liant sans chasse qui n'indique que la formation facultative d'une biconsonne ne suffisait pas. Il fallait pouvoir préciser qu'une biconsonne devait impérativement être formée car la présence de cette ligature, dans une graphie non voyellée, indique l'absence d'une voyelle implicite entre les consonnes qui forment la ligature. C'est à cet effet qu'a été introduit dans Unicode 6.0 U+2D7F  *liant de consonnes tifinaghe*. Ce caractère de commande impose la formation de la ligature. À ce titre, il joue un rôle similaire au U+0652  *soukoûn*, le signe de quiescence arabe. Le tableau ci-dessous illustre l'utilité d'une telle convention, les exemples sont tirés du dictionnaire de Foucauld³.

Lettres de base	Graphie touarègue	Translittération	Glose en français
:ⵔ	ⵏⵑⵔ	ənkər	se lever (inhabituel)
	:ⵔ	nāk:ər	se lever (habituel)
ⵔ:ⵔ	ⵔ:⊕	teyert	marmite en terre
	ⵔ:ⵔⵔ	teyərīt	cri strident, très perçant

³ Charles Eugène de FOUCAULD, *Dictionnaire touareg-français, dialecte de l'Ahaggar*. 4 volumes, Imprimerie nationale, Paris, 1951.

⊕:	⊕: ⁴	istəy	chasser, pousser devant soi
	⊕:	ɔsatay	chasse, poursuite
†ⵏⵔ	†ⵏⵔ	təfert	mot, proposition, vers
	†ⵏⵔ	tefərit	petite aiguille rocheuse
IVO	∨O	əndər	excéder les forces
	IVO	ənadar	fait d'être en chaleur, en rut

Dans les exemples ci-dessus, comme les biconsonnes sont considérées comme obligatoires, on n'utilisera pas de U+200D *liant sans chasse*, mais bien un U+2D7F *liant de consonnes tiffinaghe*. Le verbe *chasser* ⊕: se codera donc <U+2D59, U+2D7F, U+2D5C, U+2D57>.

Unicode prescrit que, si une biconsonne obligatoire ne peut être affichée à l'aide d'une ligature correcte, le système de rendu devra faire apparaître *le liant de consonnes tiffinaghe (LCT)* de manière visible afin de bien indiquer l'absence de voyelle entre les deux consonnes. Cela pourra prendre la forme d'un losange ◆, d'un  ou de toute autre glyphe. Unicode ne prescrit pas ce glyphe de repli. La forme de repli de <S, LCT, T> sera donc <gS, , gT> où gS est un glyphe de S, gT un glyphe de T et  est un glyphe qui indique la présence du *liant de consonnes tiffinaghe*.

Rappelons que si un système de rendu ne peut afficher une biconsonne optionnelle, c'est-à-dire dont les consonnes de base sont séparées par un U+200D *liant sans chasse (LSC)*, la forme de repli consiste simplement dans l'affichage des consonnes constitutives de cette ligature. La forme de repli de <S, LSC, T> est donc <gS, gT>.

3 Clavier national, norme de tri

Depuis la normalisation des tiffinaghes dans l'ISO 10646 et Unicode, un clavier normalisé marocain pour la saisie de ces caractères a été homologué par le SNIMA. Ce clavier respecte la norme internationale en la matière, l'ISO/CEI 9995. Il a été repris par plusieurs fabricants de Linux et soumis à Microsoft pour considération et inclusion potentielle dans une version ultérieure de Windows. Normalisé il y a plusieurs années, il ne comprend pas les caractères introduits par Unicode 6.0.

⁴ On aurait aussi pu mettre ⊕ à la place de ⊕:.

le grand changement introduit par cette édition est de permettre la quasi-totalité des caractères Unicode et notamment les tifinaghes dans les noms d'élément et d'attribut ainsi que les valeurs d'attribut.

En théorie, la pratique est identique à la théorie, mais en pratique ça peut bien sûr être différent. Pour qu'un nom d'élément en tifinaghes soit accepté par un logiciel de traitement de documents XML, il faut tout de même que les analyseurs (parseurs) XML soient modifiés pour mettre en œuvre la nouvelle règle de formation des noms. Il faudra donc attendre quelques années avant que les analyseurs en place soient mis à jour pour que l'échange de documents XML/XHTML avec des tifinaghes dans les noms d'éléments et d'attributs puisse se faire sans encombre. Par contre, si vos documents XML ne sont utilisés qu'en interne où vous contrôlez les analyseurs XML, il se peut que, en mettant à jour ceux-ci pour choisir une version qui prend en charge la cinquième édition de XML 1.0, vous puissiez dès aujourd'hui utiliser des documents XML dont les noms d'élément, d'attribut et les valeurs d'attributs contiennent des tifinaghes.

5 Les polices

5.1 Ebrima sur Windows

Depuis Windows 7, Microsoft inclut en série dans son système d'exploitation une police « panafricaine », la police Ebrima. Elle est fournie en deux graisses : normale et grasse. Ebrima a été conçue pour prendre en charge un grand nombre de langues africaines. Elle contient des glyphes pour le n'ko, les tifinaghes, le vaï et l'osmanya. Les glyphes latins de la police sont accompagnés des diacritiques et autres signes utilisés pour représenter les langues africaines.

Les œils tifinaghes d'Ebrima sont directement inspirés de ceux utilisés dans les tableaux Unicode (c'est-à-dire Hapax Berbère). La police ne permet pas pour l'instant l'ajout de diacritiques – ceux-ci sont mal placés – ni la composition de biconsonnes touarègues. Microsoft mettra à jour la police pour remédier à ces deux lacunes.

5.2 Les polices incorporées dans les pages HTML

Avec une écriture aussi récente – en termes de normalisation informatique – que le tifinaghe, il n'est pas rare qu'un utilisateur qui désire afficher une page Internet qui contient des caractères tifinaghes n'ait pas de police tifinaghe à sa disposition.

Malgré la diffusion récente d'Ebrima, le problème demeure entier aujourd'hui, car il existe encore de nombreux utilisateurs qui n'utilisent pas Windows et *a fortiori* Windows 7. En outre, que faire quand on veut être sûr qu'une page s'affiche dans un style tifinaghe particulier ? On risque donc dans ces cas-là d'être confronté à des pages remplies de petits rectangles blancs comme dans l'illustration ci-dessous, chaque rectangle y représente un caractère (tifinaghe ici) qui ne peut être représenté par manque de police adéquate.

Extrait du dictionnaire touareg-français de Charles de Foucauld (page 339)

- éferi □□□ sm. φ (pl. iferân □□□), dar_ăferi (ĕferi), dar_ferân || aiguille rocheuse (quelconque) || diffère d'ădaouđa « aiguille rocheuse très mince (ressemblant à un doigt) ». Tout ădaouđa est un éferi, mais non inversement.
- téferit □□□□ sf. φ (pl. tiferâtîn □□□□□), dar_tăferit (tĕferit), dar_tferâtîn || dim. du pr.
- tăfirt □□□ sf. φ (pl. tifir □□□), dar_tfir || mot (une syllable ou plusieurs syllabes réunies exprimant une idée) || p. ext. « quelques paroles (paroles en petit nombre) » || p. ext. « vers (assemblage de mots rythmés d'après des règles déterminées en poésie) » Le sing. tăfirt signifie un vers, le pl. tifir signifie 2 ou plusieurs vers.
- aferra || v. □□□ afri .

Figure 3 – Police manquante dans une page HTML

Il existe plusieurs remèdes possibles à ce désagrément. Une solution consiste à prévenir les lecteurs des pages en tifinaghe qu'ils doivent installer une ou plusieurs polices en fournissant un lien permettant de télécharger ces polices. Il existe cependant une autre solution : les polices dynamiquement téléchargeables. Cette solution consiste à envoyer les glyphes nécessaires à l'affichage d'une page HTML avec la page en question. On parle alors de polices embarquées ou incorporées.

Il y a près de deux ans, en 2008, nous avons déjà abordé les techniques alors disponibles⁸. Rappelons que les polices incorporables dans les pages HTML ne sont pas une idée neuve. Dès 1998, CSS 2 permettait de préciser un lien vers une police que les fureteurs internet pouvaient télécharger. Microsoft et Netscape prirent en charge cette syntaxe et permettaient de télécharger des polices. Malheureusement,

⁸ Patrick ANDRIES, *Demain encore plus de tifinaghes sur Internet*, Actes du colloque du CEISIC, Rabat, 2008.

aucun des deux fabricants ne prenait en charge directement le format de police le plus populaire : TrueType. Microsoft choisit le format de police EOT (un format propriétaire) dans Internet Explorer 4.0 alors que Netscape 4.0 jeta son dévolu sur un format rival TrueDoc qu'il abandonna deux versions plus tard, car Mozilla ne pouvait rendre public le code source de TrueDoc, propriété de Bitstream.

Depuis 2008, cette question a connu un vif regain d'intérêt. En 2010, on a assisté à un véritablement engouement pour les polices internet. Subitement, coup sur coup, on a assisté à l'élaboration d'un format de police conçue pour le téléchargement sur Internet (WOFF), la prise en charge de ce standard naissant par les quatre grands concepteurs de moteurs de composition HTML⁹ et enfin la mise en place de nombreux sites de partage ou de vente de polices incorporables dans des pages HTML¹⁰.

5.2.1 WOFF

WOFF est un format de police embarquée qui a été proposé au W3C par Microsoft, Mozilla et Opera. Il a été conçu pendant l'année 2009. Le 27 juillet 2010, le W3C publiait un « projet de travail » de ce standard, document qui indique un travail sérieux sur ce standard à venir.

Fondamentalement, WOFF est une enveloppe qui recouvre les polices de type *sfnt* (à savoir TrueType, OpenType ou Open Font Format) qui ont été comprimées à l'aide d'un outil qui leur permet d'être incorporées dans des pages HTML. Ce format utilise une compression *zlib* qui assure habituellement une réduction de plus de 40 % par rapport à la police TTF équivalente.

WOFF est pris en charge par Firefox depuis sa version 3.6 et par Chrome de Google depuis la version 5.0. Microsoft, pour sa part, la prend en charge dans la future version 9 de son navigateur. Enfin Opera, le permet depuis sa version 10.5.

WOFF en tant que tel ne contient aucun mécanisme de sécurité qui empêcherait la copie de la police. Les navigateurs peuvent, toutefois interroger les serveurs et ne télécharger par défaut que les polices qui sont hébergées dans le même domaine que la page qui y fait référence.

⁹ Les quatre grands moteurs de composition sont *Trident* de Microsoft, *Gecko* de Mozilla, *WebKit* utilisé notamment par Google et Safari et *Presto* d'Opera. Chacun de ces moteurs de rendu est utilisé dans plusieurs applications : Presto se retrouve ainsi, non seulement dans le fureteur Opera, mais aussi dans des produits Nintendo, Nokia, Sony, Adobe et Macromedia.

¹⁰ Parmi ces sites, on peut citer : <typekit.com>, <fontsquairrel.com>, <typotheque.com>, <openfontlibrary.org>, <fontshop.com> et <fontfont.com>.

De même, la sélection d'un sous-ensemble de glyphes correspondant aux caractères utilisés dans une page HTML donnée (ou un ensemble de pages) est laissée à des outils tiers qui ne font pas partie de la norme.

5.2.2 Autres formats

Malgré le fait que les nouvelles versions des grands navigateurs prendront toutes en charge le format WOFF, il est important de considérer le parc actuel des navigateurs internet, trois autres types de polices téléchargeables y sont utilisés :

- | | |
|-----|---|
| TTF | Fonctionne bien avec la majorité des fureteurs (mais pas IE ni l'iPhone), volumineux car non comprimé. |
| EOT | Uniquement sur IE, nécessaire pour IE 5 à 8, comprimé, permet de ne sélectionner qu'un sous-ensemble de glyphes qu'on retrouve dans une page HTML donnée. |
| SVG | Format XML nécessaire pour l'iPhone et l'iPad avant la version 4.2, volumineux. |

5.2.3 Hapax Berbère et Hapax Tifinar Carrée

Les polices Hapax Berbère allégée et Hapax Tifinar Carrée sont disponibles sous quatre formats (TTF, EOT, SVG et WOFF) respectivement aux adresses suivantes hapax.qc.ca/essai-polices-incor/hapaxber-sousensemble-webfont.zip et hapax.qc.ca/extrait-foucault/hapaxtifinarcarrée.zip.

La police Hapax Berbère allégée ne comprend qu'un sous-ensemble restreint de la police Hapax Berbère alors que la police Hapax Tifinar Carrée est complète (et donc assez volumineuse). On peut les copier à loisir. Deux pages de démonstration permettent de voir ces polices Web en action : hapax.qc.ca/essai-polices-incor/HapaxBerbereRegular-demo-hex.html et hapax.qc.ca/extrait-foucault/foucault-p339.html.

5.2.4 @font-face et CSS

Revenons à notre extrait du dictionnaire touareg de Charles de Foucauld. Comment s'assurer que la page s'affiche correctement tant sur les anciennes versions de MSIE, les iPhone, les iPad que les nouvelles versions de Firefox ?

La première étape est de créer un ensemble de polices TTF, SVG, EOT et WOFF. Un site comme Font Squirrel¹¹ permet de fournir sa police en entrée et de créer des versions SVG, EOT, TTF et WOFF de celle-ci ainsi que de ne sélectionner qu'un sous-ensemble des glyphes de la police d'origine afin de réduire encore la taille des fichiers produits.

Dans notre cas, la police Hapax Tifinar Carrée utilisée dans la page du dictionnaire de Foucauld a déjà été transformée en ces quatre formats. Il suffit de télécharger l'ensemble¹², de les dézipper et de les placer dans un répertoire de votre serveur. Le code fourni ci-dessous assume que les polices sont présentes dans le même répertoire que les feuilles de style CSS.

Il faut ensuite définir la police Hapax Tifinar Carrée dans CSS à l'aide d'une déclaration `@font-face`. C'est ce que fait le code ci-dessous. Il définit cette même police dans deux déclarations que les principaux navigateurs comprendront. Si les polices ne sont pas stockées dans le même répertoire que le code CSS, n'oubliez pas de changer les `url()` pour y indiquer l'URL qui correspond à leur position.

```
@font-face {
    /* Déclaration destinée à Internet Explorer */

    font-family: 'HapaxTifinarCarree';
    src: url('hapaxtifinarcree.eot');
}

@font-face {
    /* Déclaration destinée à tous les autres */

    font-family: 'HapaxTifinarCarree';
    src: url(//:) format('pasde404'),
        url('hapaxtifinarcree.woff') format('woff'),
        url('hapaxtifinarcree.ttf') format('truetype'),
        url('hapaxtifinarcree.svg#webfontrYQ4E2jU')
            format('svg');

    font-weight: normal;
    font-style: normal;
}
```

¹¹ Voir <fontquirrel.com/fontface/generator>. Font Squirrel fournit également les `@font-face` à inclure dans vos feuilles de style.

¹² Disponible ici : <hapax.qc.ca/extrait-foucauld/hapaxtifinarcree.zip>.

Nous reviendrons sur ces déclarations par la suite. Pour l’instant, il suffit de savoir que ces déclarations permettent aux navigateurs de savoir où trouver la définition d’une police nommée HapaxTifinarCarree.

Pour employer la police dans des documents HTML, il nous reste à préciser en CSS quand l'utiliser. Le code ci-dessous précise, par exemple, que les paragraphes (<p>) devront d'abord utiliser la police HapaxTifinarCarree, puis si des glyphes manquent d'employer la police Arial et enfin, par défaut, n'importe quelle police sans empattements.

```
p {font-family: 'HapaxTifinarCarree', Arial, sans-serif;}
```

Dans le cas de l'extrait du dictionnaire touareg, nous désirons d'abord utiliser des polices à empattements hautement optimisées comme Times New Roman pour le texte latin avant d'utiliser les glyphes de la police Hapax Tifinar Carree dont les glyphes latins sont moins esthétiques. Le code ci-dessous fera l'affaire :

```
body {  
    font-family : Times New Roman, HapaxTifinarCarree, serif;  
    width: 650px;  
    text-align: justify;  
    margin-left: auto; margin-right: auto;  
}
```

Ensuite dans le corps de la page HTML, il suffit d'écrire par exemple

```
<p>- <u>téferit</u> ⵜⴰⴳⴰⵔⴰⵏⵜ ⵜⴰⴳⴰⵔⴰⵏⵜ ⵜⴰⴳⴰⵔⴰⵏⵜ (pl. <u>tiferâtîn</u> ⵜⴰⴳⴰⵔⴰⵏⵜ),  
<u>daṛ tăferit</u> (<u>tëferit</u>), <u>daṛ tferâtîn</u> ||  
dim. du pr.</p>
```

pour que les caractères latins s'écrivent à l'aide de la police Times New Roman et les caractères tifinaghes avec la police Hapax Tifinar Carrée comme dans la figure ci-après.

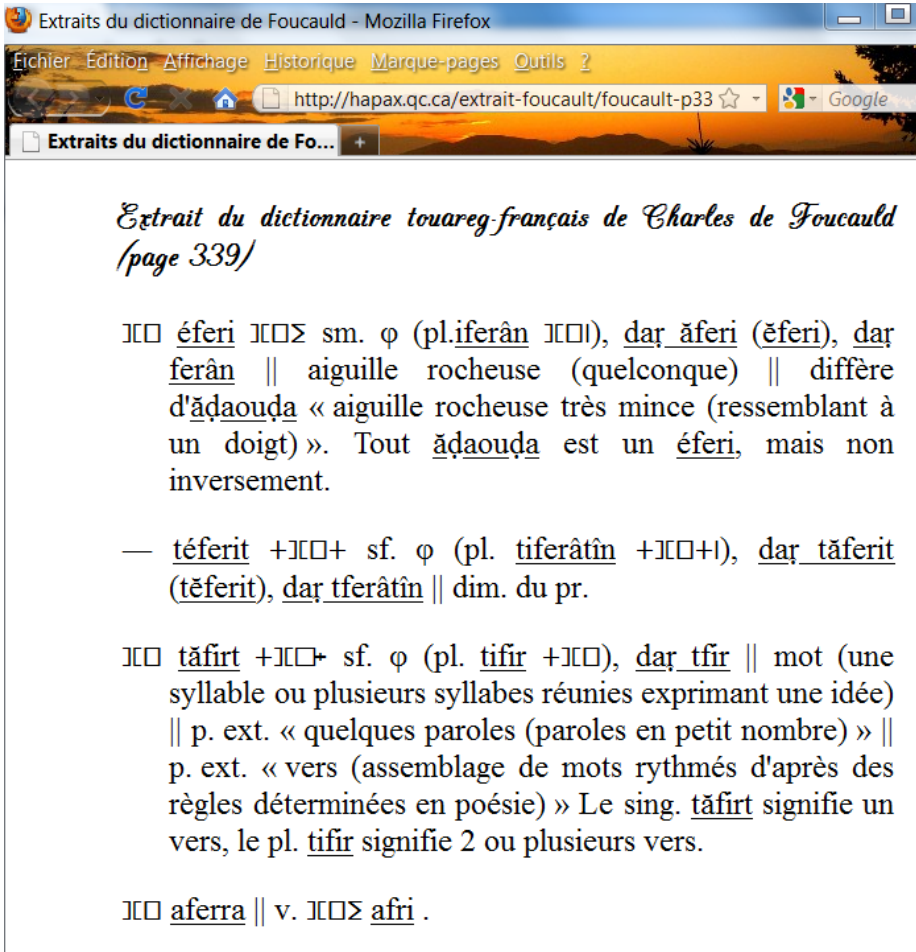


Figure 4 – Page avec des polices Internet incorporées

5.2.5 Retour sur les différents @font-face

Chaque @font-face permet de déclarer plusieurs fichiers alternatifs pour un même nom de police, en séparant chacun des emplacements, les url(), par une virgule :

```
@font-face{
  font-family : 'maPolice';
  src : url('maPolice.woff') format('woff'),
        url('maPolice.svg#abcd') format('svg'),
        url('maPolice.ttf') format('truetype');
}
```

La propriété src est suivie de l'url d'un fichier de police puis d'un format optionnel. Déclarer le format permet de préciser explicitement aux navigateurs le type de police associé à l'url qui précède et d'éviter

ainsi au fureteur d'avoir à télécharger le fichier en question pour en vérifier le format.

L'url de la police SVG se termine par un nom de fragment après le croisillon (« # »). Ce nom précise où, dans le fichier SVG, se trouve la déclaration de police. À première vue, ça peut paraître superflu, il faut toutefois se rappeler qu'un fichier SVG peut contenir bien d'autres choses qu'une police de caractères ou même en contenir plusieurs.

L'ordre de déclaration des formats est également important : d'abord WOFF, puis selon la taille des fichiers, la version SVG et enfin TrueType. Il est important de considérer l'ordre de déclaration, car certains navigateurs prennent en charge plusieurs de ces formats et utiliseront le premier qu'ils comprennent dans la liste fournie. Or, dans cette liste, WOFF est – comme nous l'avons vu – nettement plus léger que TTF. Le SVG étant un format XML, les fichiers peuvent vite devenir volumineux. Pour palier cet inconvénient, il en existe une version comprimée qui porte le suffixe `.svgz`. Elle est toutefois peu prise en charge par les fureteurs. En outre, si votre serveur gère la compression des requêtes HTTP, cet artifice n'est guère utile. La version compressée d'une police SVG est, en règle générale, plus légère que sa version TrueType.

Pourquoi, dans notre code ne trouve-t-on pas de déclaration comme ci-dessous ?

```
|      src : url('maPolice.eot') format('eot');
```

La raison en est simple : Internet Explorer ignore la propriété `format`¹³ et ne prend en charge qu'une valeur d'`url()`. En outre, confronté à la ligne

```
|      src : url('maPolice.woff') format('woff');
```

Internet Explorer cherche à télécharger une police du nom suivant : « `maPolice.woff` » `format('woff')`. MSIE envoie une requête dans ce sens au serveur et attend que le serveur lui réponde. Il le fera à l'aide d'un 404 (pas trouvé), car bien évidemment il n'existe pas de telle police sur le serveur. Cet appel au serveur est inutile et inefficace.

¹³ Ceci est corrigé à partir de la future version 9 d'Internet Explorer.

C'est pourquoi¹⁴, pour les versions 4 à 8 de MSIE, on ajoute une déclaration supplémentaire :

```
@font-face {  
    /* Déclaration destinée à Internet Explorer */  
  
    font-family: 'HapaxTifinarCarree';  
    src: url('hapaxtifinarcree.eot');  
}
```

et l'on ajoute cette première ligne dans la deuxième déclaration¹⁵ :

```
src: url(//:) format('pasde404'),
```

afin que la requête inutile d'IE échoue avant l'envoi de celle-ci, car aucun protocole ni aucun serveur ne correspond à l'url fourni.

6 Les bibliothèques logicielles

Grâce à la normalisation des tfinaghes il y a 5 ans et à leur inclusion dans Unicode, on trouve aujourd'hui des bibliothèques logicielles qui permettent de traiter les textes écrits en tfinaghes. Si Java 6 ne connaît pas les propriétés des caractères tfinaghes, ce n'est pas le cas d'ICU¹⁶, une bibliothèque Unicode¹⁷ qui sert de terrain d'essai à Java et dont les fonctionnalités sont habituellement intégrées avec un certain retard dans Java.

On trouvera ci-dessous un exemple de programme Java qui permet de découper un texte Unicode en « mots ». On remarquera que la syntaxe du `BreakIterator` ICU utilisée ici est identique à celle de Java. Il suffit souvent pour accéder aux fonctionnalités d'ICU de remplacer l'import de Java (ici `java.text.BreakIterator`) par son équivalent ICU (`com.ibm.icu.text.BreakIterator`) :

```
import com.ibm.icu.text.BreakIterator;  
  
// import java.text.BreakIterator;
```

¹⁴ Pour plus de détails sur les bogues d'IE en la matière, voir <http://covertprestige.info/css/font-face/>.

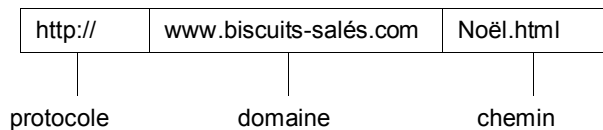
¹⁵ On préférera cette astuce à celle qui consiste à ajouter une propriété `local()` qui plante le navigateur du système d'exploitation Android utilisé sur des téléphones intelligents : readableweb.com/best-practice-for-font-face-css-takes-a-turn.

¹⁶ On peut télécharger ICU pour Java ici : [ici <download.icu-project.org/files/icu4j/>](http://download.icu-project.org/files/icu4j/)

¹⁷ Pour une introduction à ICU, voir le chapitre 12 d'*Unicode 5.0 en pratique*, Dunod, Paris, 2008.

7 Adresses internet, courriel et noms de domaine internationalisés (NDI)

Lors de notre colloque de 2008¹⁹, nous avons déjà abordé le sujet des adresses internet internationalisées du type <http://www.biscuits-salés.com/Noël.html>. Techniquement on nomme ce type d'adresse des identificateurs de ressource internationalisée (IRI). Ces adresses se divisent en trois parties principales, la première avant le « :// » indique le protocole à utiliser (ici http), ensuite vient le nom de domaine proprement dit puis, après un « / », le chemin de la ressource sur le serveur identifié par le nom de domaine.



Tous les domaines de tête (le *.ma*, le *.fr* ou *.com* final du nom de domaine) n'acceptent pas ces adresses internationalisées. Certains domaines de tête comme la Suisse (*.ch*) acceptent des accents français (et plus généralement les lettres accentuées de ses langues nationales) alors que la France n'accepte pas de tels accents dans les domaines qui se terminent par *.fr*. La politique d'attribution de ces noms est décidée par l'autorité d'enregistrement responsable d'un domaine de tête particulier. Au Canada, il s'agit d'une agence sans but lucratif (l'ACEI) régie par les lois canadiennes. Les domaines de tête génériques (*.com*, *.org*, *.biz*) sont administrés aux États-Unis en vertu des lois américaines.

Comme nous le mentionnions déjà en 2008, les navigateurs modernes transformeront de manière transparente les noms de domaine comme *biscuits-salés* en des noms de domaine compatibles avec l'infrastructure actuelle prévue pour des adresses ASCII.

¹⁹ *Demain encore plus de tiffnaghes sur Internet*, Actes du colloque du CEISIC, Rabat, 2008.

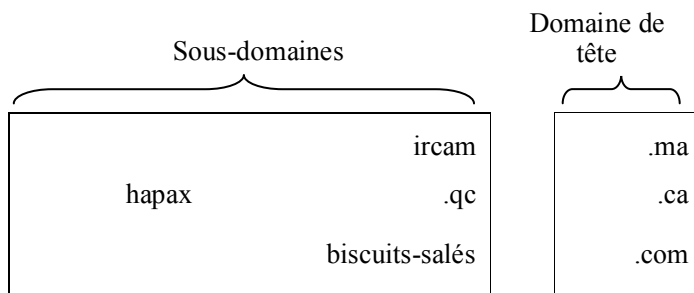


Figure 5. – Parties d'un nom de domaine

Ce qui a changé depuis ce colloque, c’est l’approbation d’IDNA 2008 qui a mis à jour le répertoire des caractères permis, passant d’Unicode 3.2 à Unicode 5.2. Par ce fait même, tous les caractères tifinaghes marocains peuvent, en théorie, être utilisés dans les adresses Internet. Il est important d’insister sur la dimension théorique de cette avancée, car l’acceptation de ces caractères dans les noms de domaine dépend des autorités d’enregistrement de noms de domaine. Pour que des sous-domaines en tifinaghes soient acceptés dans des adresses du domaine de tête .ma, il faudra donc que l’organisme responsable de l’attribution des noms de domaine au Maroc le permette.

8 Conclusion

Depuis 2005 et la normalisation du tifinaghe dans Unicode et l’ISO/CEI 10646, beaucoup de choses ont changé. Il est de plus en plus facile d’utiliser des tifinaghes sur internet et dans les applications informatiques.

Il reste encore, bien sûr, dans ce domaine quelques améliorations possibles. On peut citer des pistes comme l’ajout de nouveaux caractères tifinaghes²⁰, l’introduction d’un code de langue ISO 639 pour l’amazighe marocain, la diffusion et l’emploi de CSS 3 qui permettra une typographie fine des polices, embarquées ou non, la mise en place concrète d’IDNA 2008, la création et la diffusion de ressources de localisation en tifinaghes et amazighe, la mise à jour du clavier tifinaghe pour inclure les nouveaux caractères tifinaghes Unicode, etc. Dans chacun de ces domaines, les choses avancent et des solutions techniques

²⁰ Deux caractères tifinaghes supplémentaires (YE < et YO --) seront ajoutés à ceux d’Unicode 6.0 dans une version ultérieure d’Unicode.

se dessinent pour que l'utilisation des tfinaghes et de l'amazighe dans tous ces domaines se fasse désormais sans entraves. En cinq ans, que de chemin parcouru !

9 Remerciements

Nous tenons à vivement remercier l'IRCAM et plus particulièrement le directeur du CEISIC, Youssef Aït Ouguengay, pour leur accueil chaleureux et l'organisation du colloque international à Rabat au cours duquel cette communication a été présentée. Nous voulons également ici rendre hommage au prédécesseur de M. Aït Ouguengay, le professeur Lahbib Zenkouar, sans lequel la normalisation informatique du tfinaghe n'aurait été ni aussi rapide ni aussi complète.