ETHERPIN



Geste de savoir

# On voit un seul caractère, mais la machine en contient plusieurs

mercredi 31 juillet 2024

# Table des matières



## Introduction

Avant toute chose, une explication du sous-titre et une petite définition :

Un « *grapheme cluster* », en bon français, c'est un groupe de graphèmes. Cette terminologie est utilisée pour caractériser un fait curieux : **on voit parfois sur l'écran un seul caractère alors qu'il y en a fait plusieurs dans la machine**.
Les termes « *grapheme cluster* » sont mal choisis, car tant en anglais qu'en français, le mot graphème correspond à une lettre ou un ensemble de lettres pour former un phonème, ce qui n'a pas grand-chose à voir avec le fait curieux cité ci-dessus.
C'est pourtant la terminologie utilisée par Unicode.

**Unicode**

Unicode permet de coder tous les caractères du « jeu universel de caractères » dont l'acronyme anglais est **UCS**. Chaque caractère de l' UCS est représenté par un nombre hexadécimal entre `0x0000` et `0x10FFFF`. Ce nombre est appelé « **point de code** ». La valeur de ce nombre peut être codée avec un des trois codages suivants : UTF-32, UTF-16 et UTF-8.
Un point de code Unicode est désigné par l'écriture « U+ » suivi de son numéro hexadécimal de `U`+`0000` à `U`+`10FFFF`.
L' UCS fait partie de la norme ISO 10646 (qui a une version en français). Cette norme fournit un nom pour les caractères de l' UCS.

Beaucoup de caractères sont codés de manière unique. Mais pour des raisons historiques, certains peuvent avoir un codage alternatif. C'est le cas les **caractères composés** (par exemple les caractères coréens ou les lettres accentuées).

**Démonstration**

Sélectionnez le `é` ci-contre, et puis le recopier dans un document vide, par exemple avec Notepad++.
Faire un retour arrière `⌫ back pace`. Surprise, l'accent a disparu, il reste juste un `e` sans accents !

Explication

Il y a des représentations graphiques qui ne sont pas dans l' UCS. Ces représentations sont faites en associant plusieurs caractères de l' UCS.
Le présent article s'intéresse en particulier aux drapeaux nationaux et à certains émojis. Avec les caractères composés, les drapeaux nationaux et certains émojis, la longueur de ce qui s'affiche à l'écran ne correspond plus au nombre de caractères de l' UCS utilisés.

Cela pose des problèmes difficiles. On aimerait bien pouvoir trier, appliquer des expressions régulières, ou pouvoir gérer le curseur lors du développement d'applications dans lesquelles on peut éditer des textes.

Pour compliquer le tout, les règles applicables dépendent de la langue de rédaction. Unicode a écrit un document à ce sujet qui détaille ces problèmes.

Pour en savoir plus sur Unicode et UTF-8, voir L'encodage UTF-8 à la main

On notera qu'Unicode est une norme qui évolue (typiquement, une fois par an), sachant que les nouvelles versions restent compatibles avec les anciennes versions. C'est tout à fait souhaitable, car il s'agit ici d'écrits basés sur des conventions graphiques qui ont une certaine stabilité temporelle. Ce n'est pas tous les jours qu'on change l'alphabet.

**Affichage incorrect ?**

Pour obtenir un affichage correct des drapeaux et des émojis sur un PC, il faut utiliser une police d'émojis à jour.
En cas de problèmes, on peut utiliser le navigateur Firefox avec son extension Twemojify.

# 1. Les caractères composés

## 1.1. Les caractères composés

### 1.1.1. Caractères accentués

L'écriture du français nécessite l'utilisation d'accents, de la cédille (sans compter les ligatures). C'est aussi le cas d'autres langues européennes et de diverses transcriptions latines. Les caractères concernés sont dans le supplément Latin-1 d'Unicode.

Mais Unicode prévoit aussi des « **caractères diacritiques** ». Ce sont des caractères sans chasse destinés à se superposer au caractère précédent. Ces caractères sont dans le bloc U+300 – U+36F « Diacritiques ». On y trouve entre autres les accents pour le français et aussi la cédille. Par exemple [é] « lettre minuscule latine e accent aigu » (U+00E9) peut aussi être codé [e] (U+0065) - [◌́] (U+0301).

[◌́] (U+0301) est un des caractères diacritiques .

### 1.1.2. Le cas du hangûl (alphabet coréen)

La langue écrite coréenne est phonétique. Il y a des symboles qui représentent chacun une syllabe. Chaque symbole est construit à partir d'un ensemble de 67 signes élémentaires. Chacun de ces signes est appelé **Jamo**. Par exemple, la syllabe Gan [가](U+AC04) peut aussi se coder avec 3 jamos : [ᄀ] (U+1100) - [ᅡ] (U+1161) - [ᆫ] (U+11AB).

Chaque syllabe est composée de 2,3 ou 4 jamos. Face à une succession de jamos, il n'est pas évident de délimiter chaque syllabe. Traiter des textes en coréen devient compliqué en présence de caractères composés. Unicode recommande d'utiliser les points de code associés aux syllabes et d'éviter les formes composées.

De nombreuses autres langues utilisent des caractères qui peuvent être écrits sous la forme composée.

**Site pour voir les informations sur un caractère**

1. Hapax : entrez la valeur d'un point de code dans la barre de recherche (par exemple AC04). On obtient un fichier PDF qui contient à minima le tableau de caractères du bloc de codes concerné. Il y a souvent des informations complémentaires (il n'y en a pas pour le hangûl) avec une entrée par point de code, le nom normalisé en français et la forme composée éventuelle.
2. Compart : entrez un point de code (sous la forme U+XXXX) ou un caractère dans la barre de recherche. Vous obtiendrez les informations associées, y compris le nom normalisé en anglais et la forme composée éventuelle.

## 2. Les drapeaux nationaux

Pour coder un drapeau national, il faut connaître son code ISO 3166 sur deux lettres (ISO 3166 alpha-2). Ces codes sont décidés par les Nations Unies.

La Plateforme de consultation en ligne (OBP) de l'ISO permet de vérifier ou de trouver un code alpha-2 à jour. On représente chacune de ces deux lettres en utilisant le *Regional Indicator Symbol Letter* correspondant. Les 26 lettres de l'alphabet sont codées de U+1F1E6 à U+1F1FF. Par exemple, pour l'Afghanistan, le code alpha-2 est AF, lettres codées U+1F1E6 et U+1F1EB .
Ces lettres apparaissent en blanc sur un carré bleu clair.

A noter que la représentation n'est pas nécessairement une image du drapeau. Ainsi, sous Windows, on voit AF plutôt que le drapeau de l'Afghanistan.

On peut aussi faire apparaître le drapeau européen (EU), et celui des Nations Unies (UN).

#### 2.0.0.1. Un problème de stabilité

La norme ISO 3166 a été révisée en 1974, 1981, 1988 et 1994 et 2013. Les codes alpha-2 ne sont pas nécessairement stables dans le temps. On a vu des ensembles fédéraux se disloquer, par exemple l'URSS ou la Yougoslavie, ce qui a engrainé l'attribution de nouveaux codes.

Concernant Unicode, le principe de compatibilité ascendante interdit de supprimer quoi que ce soit aux codes nationaux / régionaux déjà présents. Si un code Alpha-2 vient à disparaître, Unicode doit faire une exception au principe de compatibilité.

Les subdivisions nationales sont encore moins stables. C'est ainsi que le drapeau du Québec n'a jamais été accepté, pas plus que ceux des états d'Amérique du Nord.

Depuis 2017, il y a quelques drapeaux régionaux dans la norme : l'Angleterre , l'Écosse et le Pays de Galle

Les risques d'instabilité n'incitent pas Unicode à introduire d'autres drapeaux régionaux.

## 3. Les émojis

Les émojis sont des représentations qui ont été créées dans le cadre de diverses applications ou implémentations propriétaires. Quand il a été question de normaliser ce qui existait de part et d'autre, Unicode s'est trouvé confronté à des approches très diverses. Il en résulte un manque de cohérence.

Dans Unicode 15.1, il y a 3782 émojis recensés. 1374 d'entre eux sont des **émojis de base**. Les émojis de base sont dans l'UCS, et font donc partie de la norme ISO 10646.
Unicode fournit une liste partielle des émojis . La liste complète est ici.

De nombreux sites présentent des listes d'émojis, dont le codage peut facilement être copié dans le presse-papier :

— openmoji.org
— emojiterra.com
— getemoji.com
— emojipedia.org

## 3.1. Séquences avec liant sans chasse

### 3.1.1. Le liant sans chasse, c'est quoi ?

Le Liant Sans Chasse (LSC) ou *Zero-Width Joiner* (ZWJ) en anglais, est un caractère spécial utilisé pour fusionner deux caractères. C'est le point de code U+200D « liant sans chasse ».

Il est en particulier utilisé pour coder des émojis relatifs aux personnes ou aux parties du corps humain au moyen de **séquences LSC**.

Il n'y a pas de moyen performant pour savoir où s'arrête une séquence.
Actuellement, les plus longues séquences ont **10 composants**.
Unicode prévoit un maximum de **32 points de code** au total pour un émoji.

Les séquences LSC sont regroupées par type :

— familles
— rôles
— genré
— cheveux
— autres

Le liant sans chasse est également utilisé pour écrire de l'arabe, du sanskrit ou d'autres langues de la région himalayenne.

vocabulaire

Les points de code sont nommés à partir de la version en français de la norme ISO 10646, tels qu'ils apparaissent sur le site https://hapax.qc.ca/ ? . Les noms sont donnés en lettres majuscules. Les mentions en lettre minuscules ne font pas partie de la norme.

## 3.2. Les modificateurs de couleur de peau

Ces modificateurs peuvent s'appliquer dans les cas où la peau humaine est visible.
Il y a 5 couleurs de peau proposées, basées sur la classification de Fitzpatrick . À savoir :

— U+1F3FB `MODIFICATEUR D'ÉMOJI PHOTOTYPE-1-2 (Peau Claire)`

- U+1F3FC MODIFICATEUR D'ÉMOJI PHOTOTYPE-3 (Peau Moyennement Claire)
- U+1F3FD MODIFICATEUR D'ÉMOJI PHOTOTYPE-4 (Peau Légèrement Mate)
- U+1F3FE MODIFICATEUR D'ÉMOJI PHOTOTYPE-5 (Peau Mate)
- U+1F3FF MODIFICATEUR D'ÉMOJI PHOTOTYPE-6 (Peau Foncée)

À partir d'un personnage ou d'une partie du corps humain ou d'une séquence avec liant sans chasse, il est possible d'ajouter un modificateur de peau pour voir apparaître l'émoji souhaité dans le cas où la plateforme le supporte. Sinon, le modificateur de peau est ignoré.

adulte peau foncée
U+1F9D1 ADULTE
U+1F3FF MODIFICATEUR D'ÉMOJI PHOTOTYPE-6 (Peau Foncée)

## 3.3. Familles

Ce type de séquences LSC regroupe une grande variété de couples et de familles avec ou sans enfants, éventuellement agrémentées de baisers ou de cœurs. Y figure également une combinaison de poignées de mains.

gens se tenant la main
U+1F9D1 ADULTE
U+200D LSC
U+1F91D POIGNÉE DE MAIN
U+200D LSC
U+1F9D1 ADULTE

gens se tenant la main, peaux moyennement claire
U+1F9D1 ADULTE
U+1F3FC MODIFICATEUR D'ÉMOJI PHOTOTYPE-3 (Peau Moyennement Claire)
U+200D LSC
U+1F91D POIGNÉE DE MAIN
U+200D LSC
U+1F9D1 ADULTE
U+1F3FC MODIFICATEUR D'ÉMOJI PHOTOTYPE-3 (Peau Moyennement Claire)

poignée de main: peau claire, peau foncée
U+1FAF1 MAIN VERS LA DROITE
U+1F3FB MODIFICATEUR D'ÉMOJI PHOTOTYPE-1-2 (Peau Claire)
U+200D LSC
U+1FAF2 MAIN VERS LA GAUCHE
U+1F3FF MODIFICATEUR D'ÉMOJI PHOTOTYPE-6 (Peau Foncée)

Dans le monde réel, une poignée de main implique deux mains droites ou deux mains gauches. Dans le monde des émojis, c'est un peu différent ! 🫵

## 3.4. Rôles

Ce sont des personnages illustrant une activité.

étudiant

U+1F468 HOMME
U+200D LSC
U+1F393 CHAPEAU DE DIPLÔMÉ

## 3.5. Composant de genre

Le genre des personnes est souvent indécis par exemple pour les personnages évoquant un métier ou pour les têtes. Il est parfois possible de donner un genre à un personnage.

Femme qui court

U+1F3C3 COUREUR
U+200D LSC
U+2640 SIGNE FEMELLE
U+FE0F SÉLECTEUR DE VARIANTE-16

U+2640 est utilisé en botanique et en zoologie, il ne s'agit pas seulement de l'espèce humaine. C'est pourquoi le texte de la version en français de la norme ISO 10646 est SIGNE FEMELLE et non pas SIGNE FÉMININ.

Le sélecteur de variante sert à indiquer qu'il faut une présentation spéciale ; il fait suite à un composant de genre.

## 3.6. Chevelures

Il y a quatre type de chevelure proposées :

U+1F9B0 ÉLÉMENT D'ÉMOJI CHEVEUX ROUX
U+1F9B1 ÉLÉMENT D'ÉMOJI CHEVEUX BOUCLÉS
U+1F9B2 ÉLÉMENT D'ÉMOJI CHAUVE
U+1F9B3 ÉLÉMENT D'ÉMOJI CHEVEUX BLANCS

homme à cheveux roux

U+1F468 HOMME
U+200D LSC
U+1F9B0 ÉLÉMENT D'ÉMOJI CHEVEUX ROUX

## 3.7. Autres séquences LSC

Les séquences LSC ne sont pas réservées aux corps humains. Il y a peu d'autres utilisations, par exemple :

`chien d'assistance`
U+1F415 CHIEN
U+200D LSC
U+1F9BA GILET DE SÉCURITÉ

### 3.7.1. Encore des drapeaux !

`drapeau de pirate`
U+2691 DRAPEAU NOIR
U+200D LSC
U+2620 TÊTE DE MORT

`drapeau arc-en-ciel`
U+2691 DRAPEAU NOIR
U+200D LSC
U+1F308 ARC-EN-CIEL

## Conclusion

Pour pallier les difficultés liées à l'incohérence entre ce qu'on voit et ce que contient la machine, le langage Swift définit un caractère comme étant un « *grapheme cluster* », conformément à ce que l'on voit à l'écran. En Swift, un drapeau national est compté comme un seul caractère dans la machine alors qu'il y a 2 points de code. De même un émoji codé avec 10 points de code est vu comme un seul caractère.

C'est une solution radicale, mais qui nécessite une puissance de calcul importante en cas d'utilisation intensive des chaînes de caractères. De plus, le style de programmation résultant est contestable.

> Un petit exemple trouvé sur un forum

Malheureusement, Swift ne prévoit pas un mode « *legacy* » **avec des performances acceptables**.

Une tentative d'introduire le concept en Python a été abandonnée, car il y avait un déficit de performances rédhibitoires (et ce n'était pas codé avec les pieds). Du coup, les concepteurs ont pensé que le concept n'était pas très pertinent en Python, qui après tout, n'est pas destiné à coder des interfaces pour les smartphones et les tablettes[1].

Il semble qu'avec `unicode-segmentation V1.11.0` , Rust ait une solution viable. Cela permet de traiter les « *grapheme clusters* » sans pour autant impacter les traitements plus classiques.

1. Voir ce package python et ce rapport de bug

À noter que ces implémentations sont tributaires des évolutions d'Unicode, ce qui implique un travail de maintenance obligatoire pour rester conforme à la norme en vigueur, tout en préservant l'existant.

# Contenu masqué

## Contenu masqué n°1 : Explication

Au lieu de le coder `0xC3 0xA8` (standard Unicode, en UTF-8), il est codé `0x65 oxCC 0x81`, c'est à dire un `e` suivi d'un `accent aigu`. On voit un seul caractère : `é` alors qu'il y en a deux dans la machine. Retourner au texte.

## Contenu masqué n°2 : Un petit exemple trouvé sur un forum

comment obtenir les 10 premiers caractères d'un texte, qqc comme substring(text, 0, 10) en langage courant :

```
let nsRange = NSRange(location:0, length:10)
let r = Range(nsRange)
var start = text.index(text.startIndex, offsetBy: r!.lowerBound)
var end = text.index(text.startIndex, offsetBy: r!.upperBound)
text = String(text[start..<end])
```

*Pourquoi faire simple quand on peut faire compliqué* (proverbe Shadock) Retourner au texte.

# Liste des abréviations

UCS Universal Coded Character Set. 1, 2