

# Cours 17 : Chaînes en PHP

Introduction à la programmation Web



# Encodages

Avant d'étudier les fonctions de traitement de chaînes de caractères et les autres techniques qui s'y rattachent, il faut comprendre les deux types d'encodage utilisés pour les caractères.

1. Le type sur un seul octet, comme par exemple l'encodage ASCII, qui peut représenter tous les caractères en anglais, mais pas les caractères accentués, chinois, etc.
2. Le type sur deux octets, comme par exemple l'encodage UTF-8, qui permet de représenter les caractères accentués.

En effet, plusieurs fonctions et fonctionnalités ont été conçues seulement pour les chaînes à un octet. D'autres fonctions ont été ensuite conçues pour fournir une alternative à ceux qui utilisent les chaînes à deux octets, comme nous, les francophones!

# Traverser une chaîne (chaînes à un octet)

- En PHP, il est possible de traverser une chaîne de caractères de la même façon qu'on traverserait un tableau, caractère par caractère.
- En effet, il est possible d'accéder à un caractère dans une chaîne à l'aide de sa position ainsi :
  - `$caractere = $chaine[4]`
  - Ceci accèdera au 5ème caractère de la chaîne (n'oubliez pas le **zéro**!)
- Pour boucler à travers la chaîne, par contre, il faut connaître sa longueur... Il existe donc la fonction **strlen()**, qui prend en paramètres une chaîne de caractères et qui retourne sa longueur.
- Par contre, cette façon de traverser la chaîne est seulement valable pour les chaînes dont les caractères sont encodés sur **un seul octet**. Pour des chaînes contenant des accents, cela ne donnera pas le bon résultat!

# Traverser une chaîne (chaînes à deux octets)

Le principe est le même pour les chaînes à deux octets, mais on ne peut pas utiliser les brackets [ ].

Il faut combiner `mb_strlen()` (la version avec encodage de `strlen`) avec l'utilisation de la fonction `mb_substr()`, qui nous permet d'accéder à une sous-chaîne.

N.B. La plupart des fonctions de traitement de chaînes de caractères possèdent un équivalent débutant par `mb_`, qui permet de spécifier un encodage et donc de fonctionner avec les caractères à deux octets.

# Changement de majuscule à minuscule (et vice-versa)

Les fonctions `strtoupper()` et `strtolower()` servent respectivement à faire passer tous les caractères d'une chaîne de minuscule à majuscule, et de majuscule à minuscule.

Les équivalents `mb_strtoupper` et `mb_strtolower` existent.

# Fonctions de recherche de sous-chaînes

Il est possible de confirmer l'existence d'une sous-chaine dans une chaîne, pour fins de recherche, avec les fonctions `strpos` et `strstr`.

La fonction `strpos()` reçoit deux paramètres, le premier étant la chaîne **dans laquelle** il faut chercher (on l'appelle **haystack**, pour botte de foin), le second étant la **chaîne recherchée** (on l'appelle **needle**, pour l'aiguille!). Le retour de cette fonction sera la position de la sous-chaîne à l'intérieur de la chaîne, ou `false` si la sous-chaîne est introuvable.

La fonction `strstr()` prend les mêmes paramètres, mais retourne le reste de la chaîne après la sous-chaîne trouvée (incluant la chaîne trouvée).

Les équivalents avec `mb_` existent aussi!

# Récupération d'une sous-chaîne

Pour obtenir une partie d'une chaîne contenue entre deux positions, on utilise la fonction `substr()`, qui prend en paramètres la chaîne sujet, une position et une longueur, et elle retourne la sous-chaîne contenue entre ces deux positions.

L'équivalent `mb_substr()` existe.

# Élagage d'une chaîne

Pour enlever les caractères autour d'une chaîne qui pourrait contenir des espaces, il existe la fonction `trim()`, qui prend en paramètres la chaîne de caractères et qui enlève les espaces qui précèdent et qui suivent le mot.

La fonction `trim()` n'a pas d'équivalent pour les caractères à deux octets, mais elle fonctionne bien avec l'encodage UTF-8. Avec UTF-16, elle peut causer des problèmes.



# Remplacement de sous-chaîne

Il est possible de remplacer un morceau d'une chaîne par une autre chaîne.

On le fait avec la fonction `str_replace`, qui prend 3 paramètres. Le premier est la chaîne que l'on cherche, le second est la chaîne que nous allons utiliser comme remplacement, et le troisième est la chaîne sujet.

Il n'y a pas d'équivalent `mb_`, mais il n'y a généralement pas de problème d'encodage avec `str_replace` si les chaînes sujet et remplacement ont le même encodage.

# Diviser une chaîne : `str_split`, `explode`, `implode`

Il est possible de transformer une chaîne en un tableau de caractères avec la fonction `str_split()` (ou `mb_str_split`), qui prend une chaîne en paramètres et retourne un tableau de caractères.

Il est aussi possible de transformer une chaîne en tableau en spécifiant un caractère ou une chaîne de séparation. La chaîne sujet sera séparée à chaque fois que l'on rencontre une chaîne de séparation. La fonction utilisée pour le faire est **`explode()`**, qui prend en paramètres le délimiteur et la chaîne à séparer, et qui retourne un tableau de chaînes. Le contraire de `explode`, qui consiste à prendre un tableau de chaînes et à former une seule chaîne avec ces dernières s'appelle **`implode()`**.