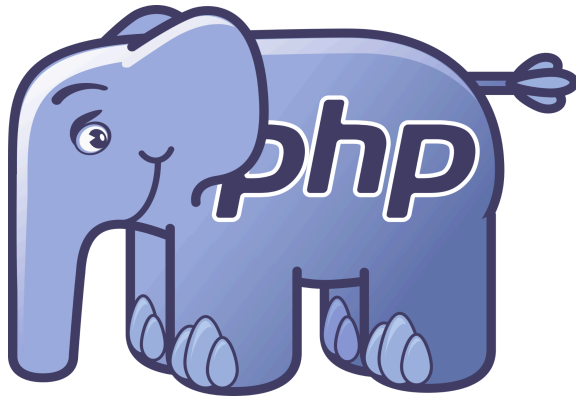


Web avancé – L2 MIASHS

TD 07 - Petits premiers pas en PHP



Dans ce TD nous allons faire les premiers en PHP. Ce TD part du principe que vous maîtrisez déjà Javascript et les concepts de programmation inhérents : logique, fonctions, etc. L'accent sera donc mis sur les différences sans pour autant répéter les éléments similaires.

Objectif : commencer PHP en douceur et dans la bonne humeur

Cible : les étudiants connaissant déjà Javascript et des concepts de programmation

Durée : très variable, en fonction de votre niveau de départ. Si vous avez déjà fait du PHP par le passé, n'hésitez pas à tout de même suivre ce TD, cela vous sert de révision.

Un conseil important : même si ce TD peut, à certains points, vous sembler facile, ne sautez pas d'étape !

Chaque exercice se distingue par ce logo  !

Dans ce TD, le code est intégré sous forme d'image afin que vous ne puissiez pas simplement le copier-coller.

Installation

Je vous invite à installer MAMP en version gratuite. Bien sûr vous pouvez toujours vouloir installer XAMPP mais MAMP vient sans Perl, et Perl est un autre langage de programmation dont nous avons pas besoin. L'objectif est de garder la simplicité d'installation pour les premiers pas en PHP.

Cas 1 : Vous avez XAMPP d'installé

Si vous avez un XAMP d'installé, je vous invite à le conserver. Vérifiez simplement que vous utilisez bien PHP8.2 (dernière version donc) et MySQL.

Cas 2 : Vous êtes sous Windows

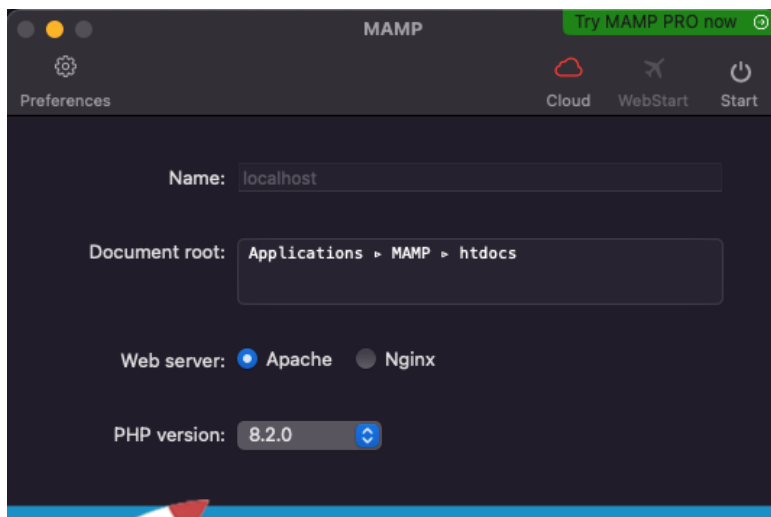
La procédure d'installation est visible ici (en anglais) : <https://youtu.be/sv0pKGyKYtU>

1. Rendez-vous sur <https://www.mamp.info/en/downloads/> et téléchargez la version Windows
2. Exécutez le fichier téléchargé, suivez les étapes mais ne choisissez pas MAMP Pro.
3. Normalement vous trouverez les fichiers dans C:\MAMP

Cas 3 : Vous êtes sous Mac

La procédure d'installation est visible ici (en anglais) : <https://youtu.be/NqaZ2ygpA4E>

1. Rendez-vous sur <https://www.mamp.info/en/downloads/> et téléchargez la version Mac qui convient (si vous avez un MAC M1 ou + alors prenez la version Apple Silicon)
2. Exécutez le fichier téléchargé, suivez les étapes
3. Une fois fini regardez bien dans le dossier Applications que vous bien un dossier MAMP installé
4. Lancez MAMP et constatez une petite fenêtre



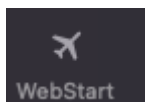
5. Nous utiliserons la version 8.2 de PHP avec un serveur Apache. Vérifiez bien qu'ils sont bien sélectionnés.

Cas 4 : Vous êtes sous Linux

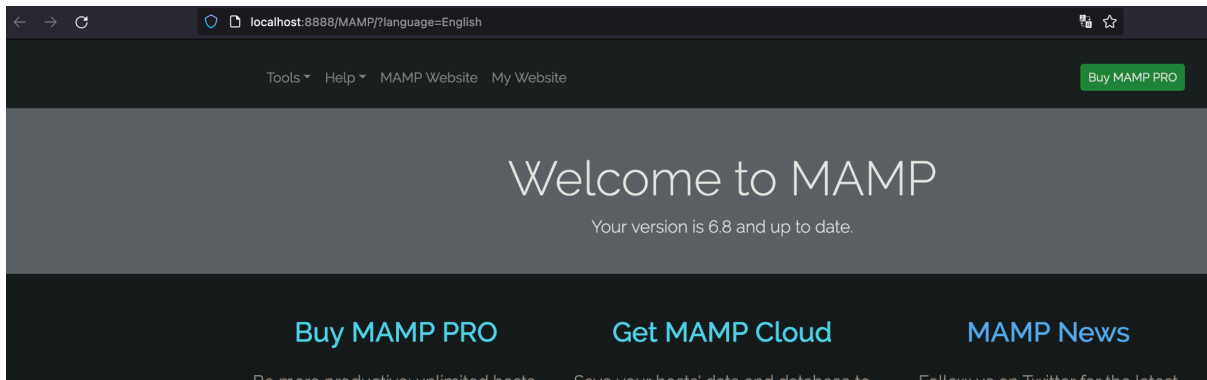
Super, vive linux ! Je vous invite à suivre toute la procédure pour installer LAMP. Cette adresse officielle la décrit pour les utilisateur de distributions fondées sur debian (comme Ubuntu) : <https://doc.ubuntu-fr.org/lamp>

N'oubliez pas de lancer **apt update** avant de commencer !

C'est bon ? Tout est installé ? Lancez le serveur avec le bouton "Start" et cliquez sur



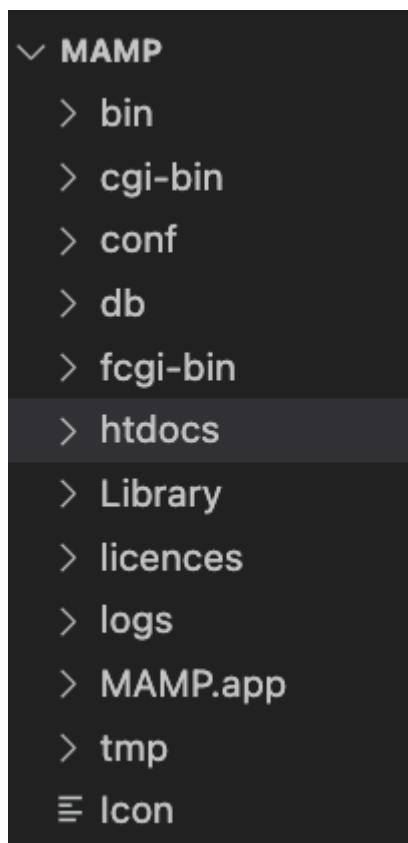
"WebStart". Vous verrez alors votre navigateur ouvrir une page avec la bonne URL qui indique le port (8888 par défaut) comme ceci :



Tout est bon pour commencer l'aventure à dos d'éléphant ! 

Votre premier script PHP

- Ouvrez le dossier MAMP dans Visual Studio Code. Vous le trouverez dans Applications sous mac, dans C:\MAMP sous Windows, /opt/lamp ou /var/www/html sous Linux.
- Vous verrez alors tout le dossier dans VSCode



- C'est dans le dossier **htdocs** que vous allez mettre vos fichiers .php
- Créez-y un fichier nommé **premierpas.php**

Maintenant vous pouvez enfin coder en PHP. Et pour cela... commençons par du HTML 😄



- Dans votre fichier mettez le code HTML suivant

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Premier pas</title>
</head>
<body>
  <h1>Bonjour HTML</h1>
</body>
</html>
```

Tout simple n'est-ce pas ?

- Maintenant rendez-vous sur votre navigateur à l'adresse <http://localhost:8888/premierpas.php> . Alors ? Pas mal non ? On voit du HTML, c'est supeeeer.
 - Attention, si vous êtes sous Windows, l'adresse de base sera alors <http://localhost/premierspas.php> car le port 80 est utilisé par défaut (c'est le port de référence de HTTP d'où le fait qu'il n'y ait pas besoin de l'indiquer)
- Dans votre fichier **premierpas.php**, dans le body ajoutez une demi balise ouvrante php et insérez y du texte. Comme ceci :

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Premier pas</title>
</head>
<body>
  <h1>Bonjour HTML</h1>
  <?php
    echo "mon premier pas 🥳";
  ?>
</body>
</html>
```

- Sauvegardez et rechargez la page dans votre navigateur.

Et voilà ! Le premier pas effectué avec succès. 🥳

Gardez à l'esprit qu'il faut toujours utiliser la demi balise `<?php` pour pouvoir mettre du code PHP qui sera interprété. Contrairement à certaines pratiques, je vous conseille également de toujours mettre la demi balise fermante `?>`. Cela vous permettra de bien intégrer différents morceaux de PHP dans le fichier HTML.

Aussi, vous pouvez voir que nous avons utilisé le mot clé **echo** pour afficher une valeur. La sortie standard de PHP étant le fichier HTML, la valeur s'affiche dans votre navigateur.

Enfin, vous voyez bien que les instructions se terminent pas un point virgule. C'est obligatoire, sinon le code ne fonctionnera pas.

Maintenant, nous allons nous amuser un peu avec PHP pour prendre en main ce langage. N'ayez crainte, ce langage est considéré plus accessible que Javascript 😊

Les variables

En PHP les variables se déclarent avec le préfixe `$`. Cela indique toujours que nous voulons accéder à une variable, aussi bien lors de la déclaration que lors de l'utilisation de ces variables.

Une variable a toujours un type. Les principaux types sont :

```
// String (chaîne de caractères)
$mavariabLe = "Yo ! Ça roule ?";
$mavariabLe = 'Yo ! Ça roule ?';

// Integer (entier)
$mavariabLe = 9999;

// Float (décimales)
$mavariabLe = 99.99;

// Boolean (booléens)
$mavariabLe = true;
$mavariabLe = false;

// Array (tableau)
$mavariabLe = array("élément1😂", "élément2");

// NULL (une variable nulle)
$mavariabLe = null;
```

Le type d'une variable peut être récupéré sous forme de String à l'aide de [gettype\(\)](#).

À partir de maintenant, **tout ce TD est à faire dans ce fichier premierpas.php !**

(N'oubliez pas d'utiliser la balise HTML `</br>` pour ajouter une ligne si besoin.)

Ok ? C'est parti ! 😊

Manipuler des chaînes de caractères (String)

Les Strings peuvent être déclarées par quatre manières en PHP, mais seules deux manières nous intéressent ici : soit à l'aide de double guillemets `""` soit à l'aide de simples guillemets `'`. Voyons les différences en pratique. Aidez-vous de la [documentation](#) si besoin.



- Avec l'utilisation de doubles guillemets :
 - Créez une variable **prenom** contenant votre doux prénom
 - Créez une variable **salutation** contenant "Salut " accompagné de l'appel de la variable précédente comme ceci `"Salut $prenom";`
 - Affichez la
- Faites la même chose mais avec des simples guillemets `'` et une variable **salutation2**

Pas mal n'est-ce pas ? Les guillemets simples permettent d'afficher littéralement le contenu en tant que texte, tandis que les guillemets doubles vont l'analyser.

PHP possède des fonctions dédiées à la manipulation de Strings. Voyons les incontournables en pratique :



- Affichez la longueur de la variable **prenom**
 - Aidez-vous de la fonction [strlen\(\)](#)
- Affichez le nombre de mots de **salutation**
 - Aidez-vous de la fonction [str_word_count\(\)](#)
- Remplacez "Salut" par "Yo" dans **salutation**
 - Aidez-vous de la fonction [str_replace\(\)](#)

Il serait pas mal d'ajouter du contenu à la variable **salutation** non ? Par exemple, un point d'exclamation ou un emoji ? Pour cela il faut concaténer des Strings. En PHP cela se fait avec l'opérateur `.` ou, pour ajouter à la fin d'un contenu existant, avec `.=` (plus d'infos dans la [documentation](#))



- Créez une variable contenant uniquement un point d'exclamation
- Concaténez **salutation** avec cette variable et mettez le résultat dans une nouvelle variable, puis affichez cette dernière variable.
 - Vous avez besoin de l'opérateur `.` (le point)

- À l'aide de l'opérateur `.` = modifiez directement la dernière variable créée en y ajoutant un emoji de votre choix.

Cela devrait suffire pour une introduction aux Strings. Passons désormais aux numériques



Manipuler des numériques (Integer, Float)

En PHP les numériques suivent la répartition de types habituelle entre entiers (Integers), flottants (Floats) mais également d'autres types comme Infinity (infini), et NaN (not a number, indique que ce n'est pas un numérique). Dans ce TD nous allons nous concentrer sur les deux premiers, car ce sont les deux principaux types et les plus importants.

Commençons par les entiers ([integer](#)). L'objectif des entiers est de prévoir moins d'espace en mémoire vive (RAM) que les nombres flottants ([floats](#)). Comme le typage de PHP est libre, si la taille est trop importante un integer deviendra un float.

Voyons cela en pratique :



- Créez une variable **age** contenant votre âge. Affichez la.
- Affichez le type de la variable **age**.
- Créez une variable contenant les cinq premiers chiffres du nombre Pi puis affichez-la.
- Affichez le type de cette dernière variable.

Comme en Javascript, les numériques peuvent être utilisés à l'aide de plusieurs opérateurs. Vous pouvez les trouver dans la [doc officielle de PHP](#). Voici la table qui nous intéresse :

Opérations élémentaires		
Exemple	Nom	Résultat
<code>+\$a</code>	Identité	Conversion de <code>\$a</code> vers int ou float, selon le plus approprié.
<code>-\$a</code>	Négation	Opposé de <code>\$a</code> .
<code>\$a + \$b</code>	Addition	Somme de <code>\$a</code> et <code>\$b</code> .
<code>\$a - \$b</code>	Soustraction	Différence de <code>\$a</code> et <code>\$b</code> .
<code>\$a * \$b</code>	Multiplication	Produit de <code>\$a</code> et <code>\$b</code> .
<code>\$a / \$b</code>	Division	Quotient de <code>\$a</code> et <code>\$b</code> .
<code>\$a % \$b</code>	Modulus	Reste de <code>\$a</code> divisé par <code>\$b</code> .
<code>\$a ** \$b</code>	Exponentiation	Résultat de l'élévation de <code>\$a</code> à la puissance <code>\$b</code> .

La division entière se faisant à l'aide de la fonction [intval\(\)](#).

À l'aide de ces opérateurs :



- Mettez l'heure actuelle dans une variable, avec les secondes en tant que nombres flottants. Affichez la.
- Divisez cette variable par deux puis affichez-la.
 - N'oubliez pas de réassigner sa nouvelle valeur directement avec `=`

Cas d'usage 1 : Note du bac



Un de vos amis a obtenu ses notes de bac mais n'arrive pas à connaître sa note finale. Il a obtenu 10 en anglais, 5 en philosophie et 19 en mathématiques. Sachant que les coefficients de ces matières sont respectivement : 2, 6 et 3. Combien a-t-il obtenu ? Servez vous des opérateurs numériques pour afficher la réponse.

Manipuler des booléens, des comparaisons et des conditions

Les booléens en PHP sont... des booléens 🙌 Il n'y a rien de spécial, vous pouvez appliquer toutes vos connaissances en logique pour manipuler des booléens.

Les booléens sont donc soit vrais (true) soit faux (false), sachant que chaque condition renvoie un booléen. Gardez cependant à l'esprit que vous ne pouvez pas afficher un booléen directement dans le HTML. Je vous conseille d'utiliser la fonction [var_dump\(\)](#) pour le rendre intelligible.

L'usage des booléens est intéressant quand il est combiné aux [opérateurs de comparaison](#) que voici :

Opérateurs de comparaison		
Exemple	Nom	Résultat
<code>\$a == \$b</code>	Égal	true si <code>\$a</code> est égal à <code>\$b</code> après le transtypage.
<code>\$a === \$b</code>	Identique	true si <code>\$a</code> est égal à <code>\$b</code> et qu'ils sont de même type.
<code>\$a != \$b</code>	Différent	true si <code>\$a</code> est différent de <code>\$b</code> après le transtypage.
<code>\$a <> \$b</code>	Différent	true si <code>\$a</code> est différent de <code>\$b</code> après le transtypage.
<code>\$a !== \$b</code>	Différent	true si <code>\$a</code> est différent de <code>\$b</code> ou bien s'ils ne sont pas du même type.
<code>\$a < \$b</code>	Plus petit que	true si <code>\$a</code> est strictement plus petit que <code>\$b</code> .
<code>\$a > \$b</code>	Plus grand	true si <code>\$a</code> est strictement plus grand que <code>\$b</code> .
<code>\$a <= \$b</code>	Inférieur ou égal	true si <code>\$a</code> est plus petit ou égal à <code>\$b</code> .
<code>\$a >= \$b</code>	Supérieur ou égal	true si <code>\$a</code> est plus grand ou égal à <code>\$b</code> .
<code>\$a <=> \$b</code>	Combiné	Un entier inférieur, égal ou supérieur à zéro lorsque <code>\$a</code> est inférieur, égal, ou supérieur à <code>\$b</code> respectivement.

Les conditions booléennes peuvent être associées avec des [opérateurs logiques](#) que voici :

Les opérateurs logiques		
Exemple	Nom	Résultat
<code>\$a and \$b</code>	And (Et)	<code>true</code> si <code>\$a</code> ET <code>\$b</code> valent <code>true</code> .
<code>\$a or \$b</code>	Or (Ou)	<code>true</code> si <code>\$a</code> OU <code>\$b</code> valent <code>true</code> .
<code>\$a xor \$b</code>	XOR	<code>true</code> si <code>\$a</code> OU <code>\$b</code> est <code>true</code> , mais pas les deux en même temps.
<code>! \$a</code>	Not (Non)	<code>true</code> si <code>\$a</code> n'est pas <code>true</code> .
<code>\$a && \$b</code>	And (Et)	<code>true</code> si <code>\$a</code> ET <code>\$b</code> sont <code>true</code> .
<code>\$a \$b</code>	Or (Ou)	<code>true</code> si <code>\$a</code> OU <code>\$b</code> est <code>true</code> .



- À l'aide des opérateurs, vérifiez vous-même si le OU est exclusif ou inclusif en PHP.
- Affichez le résultat entre deux variables true associées par un OU exclusif.

Pour compléter la logique du premier ordre (sans quantificateurs) il nous faut des conditions. Les conditions [SI ALORS SINON](#) suivent la syntaxe suivante en PHP :

```

if ($condition1){
    // SI condition1 est vraie ALORS
}else{
    // SINON ...
}

if ($condition1){
    // SI condition1 est vraie ALORS
}elseif ($condition2){
    // SINON SI condition 2 est vraie ALORS
}else{
    // SINON si aucune n'est vraie ALORS
}
  
```

Cas d'usage 2 : Marie Antoinette





Marie Antoinette est célèbre pour avoir répondu au peuple : “Ils n’ont pas de pain ? Qu’ils mangent de la brioche !”. À l’aide de variables booléens, représentez l’état de départ (le peuple a faim et n’a pas de pain ni de brioche). Puis mettez en place toute la structure logique de cette phrase. La représentation finale de l’action historique qui en découle (exécution de Marie Antoinette) est à votre discrétion (emojis, String, etc.) 🤖

Manipuler des tableaux

Les tableaux en PHP sont de plusieurs types : tableaux à une dimension, à plusieurs dimensions (même si techniquement il s’agit de tableaux de tableaux), tableaux associatifs et tableaux de tri. Nous nous intéressons ici seulement aux [tableaux à une dimension](#) ainsi qu’aux [fonctions qui permettent de les manipuler](#).

Les tableaux servent à représenter des séquences de données, chaque élément étant indexé à une position. Un array se déclare comme ceci :

```
// // Array (tableau)
$mavariabale = array("élément1🤖", "élément2");
```



- Créez un tableau **artistes** contenant 5 de vos artistes préférés (tout média confondu). Affichez ce tableau.
 - Utilisez **var_dump()** pour bien l’afficher
 - Vous pouvez voir qu’il s’agit d’une association indice⇒valeur
- Ajoutez au tableau **artistes** un autre artiste que vous appréciez. Puis affichez **artistes**.
 - Pour ajouter un élément vous devez utiliser la fonction [array_push\(\)](#)
- Accédez au dernier artiste du tableau et affichez-le.
- Supprimez le dernier élément du tableau
 - Utilisez la fonction [array_pop\(\)](#).
- Triez artistes par ordre alphabétique et affichez-le.
 - Utilisez [asort\(\)](#)
- Ajoutez un nouvel artiste en début du tableau **artistes** puis affichez-le.
 - Utilisez la fonction [array_unshift\(\)](#)
- Triez par ordre décroissant le tableau puis affichez-le.
 - Utilisez [arsort\(\)](#)
- Enlevez ce même premier élément du tableau **artistes** puis affichez-le.
 - Utilisez [array_shift\(\)](#)
- Ajoutez l’artiste “Minnie Riperton” en troisième position du tableau **artistes**, puis affichez-le.
 - Utilisez [array_splice\(\)](#). Elle s’utilise de la même manière qu’en Javascript.

Pas mal, vous avez vu les principales fonctions pour manipuler des tableaux. Pour obtenir des infos sur ces tableaux, je vous invite par exemple à regarder les fonctions [count\(\)](#) et [array_count_values\(\)](#) qui sont très utiles pour connaître le nombre d’éléments dans un tableau ou le nombre d’occurrence par élément différent.

Parcourir un tableau

Voyons désormais comment parcourir un tableau avec une boucle POUR ([for](#)). Bien entendu la boucle TANTQUE ([while](#)) est également disponible.

La boucle for en PHP fonctionne comme en Javascript, Java, etc. En voici un exemple pour compter de zéro à 9.

```
for ($indiceDeDepart = 0 ; $conditionDarret ; $pasDeParcours) {
}
// par exemple
for ($i = 0; $i < 10; $i++) {
    echo $i; // compte de 0 à 9
}
```

À l'aide de cette syntaxe :



- Affichez chaque artiste de la variable **artistes** un à un, un par ligne.
- Affichez chaque artiste dans le sens inverse.
- Affichez un artiste tous les trois artistes.

Cas d'usage 3 : pas de petite économie !



Monsieur Picsou a besoin de savoir quel est le plus petit prix parmi une liste de prix désordonnée. Aidez monsieur Picsou à ordonner cette liste et affichez le prix le plus bas.

Voici la liste désordonnée :

```
$prix = array(234, 5235, 235, 112, 98.5, 154);
```

Manipuler des tableaux associatifs

Comme vous l'avez vu, les tableaux sont des associations indice⇒valeur. Donc nous pouvons déjà quelque peu les considérer comme des tableaux... associatifs 😊

Les tableaux associatifs sont donc des associations entre une clé et une valeur. En PHP, vous pouvez les associer avec le caractère que j'utilise depuis le début et que vous avez vu à chaque fois que vous affichez un tableau : la flèche ⇒ .

Voici un tableau associatif de base :

```
// déclaration d'un tableau associatif
$tableauAssociatif = array("clé" => "valeur");
// accès au meilleur fruit
$tableauAssociatif["clé"];
```

Sachant cela :



- Créez un tableau associatif **moimoi** contenant votre nom, votre prénom et votre âge.
- Accédez à votre prénom à partir de **moimoi** et affichez-le.

Cas d'usage 4 : Mon personnage de MMO (jeu en ligne)



Vous avez besoin de représenter les données de votre personnage de MMO. Vous l'aviez déjà en Javascript dans le TD précédent. Convertissez donc ce tableau associatif de Javascript en tableau associatif de PHP puis affichez-le.

```
var personnage = {
  "tête": "Casque des abysses",
  "torse": "Armure fébrile",
  "jambes": "Sandales",
  "sac": ["potion", "ordinateur"],
  "sorts": {
    "feu": {
      "description": "Une flamme incandescente vient purifier tous vos adversaires",
      "visuel": "🔥",
      "puissance": 76
    }
  }
};
```

Parcourir un tableau associatif

Voyons désormais comment parcourir les éléments d'un tableau associatif. Pour ce faire, nous allons voir une autre approche de la boucle for (POUR) qui est le "pour chaque" ([foreach](#)). Cette approche est également disponible pour itérer sur de simples tableaux mais ne sert alors que de raccourci syntaxique. En revanche, pour un tableau associatif, l'ordre ne compte pas, donc ce sont les clés qui nous intéressent. C'est pourquoi le foreach devient LA manière de les parcourir.

Voici la syntaxe de base :

```
foreach ($tableauAssociatif as $clef => $valeur) {  
    echo "$clef => $valeur";  
}
```

Maintenant :



- Affichez les valeurs une à une, une par ligne à partir de **moimoi**
- Affichez ensemble les clés et les valeurs, une par une, sur chaque ligne à partir de **moimoi**



Nouveau

Big Star
Baskets basses - pink
45,99 €



Keddo
Sandales compensées - black
44,99 €



Hot Drop Exclusivité
New Balance
WS327 JEWELRY - Baskets basses - white
109,95 €



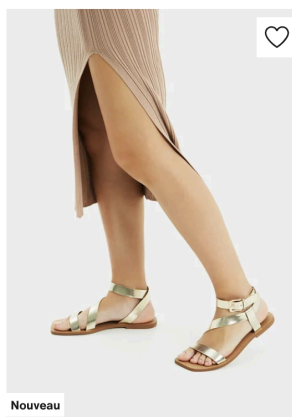
Nouveau

Jordan
AIR JORDAN 1 - Baskets basses - sunset ...
119,95 €



Nouveau

Bershka
CAGE 11810160 - Sandales à plateforme - ...
35,99 €
🚚 Livraison longue distance



Nouveau

Bershka
METALLIC STRAP - Sandales - gold colou...
19,99 €
🚚 Livraison longue distance

Cas d'usage 5 : Placard à chaussures plein



Vous avez perdu la représentation des données de votre boutique en ligne. Mince alors !!! Vos supérieurs hiérarchiques seront furieux ! Heureusement, vous avez une image de ces données. Il ne vous reste donc plus qu'à les représenter sous forme de tableau associatif puis à parcourir chaque entrée pour les afficher une par ligne. Les données qui vous intéressent sont le prix de la chaussure, son nom, sa couleur et son type (chaussure basse, etc.). Il vous faudra aussi indiquer si la livraison longue distance est disponible ou non.

C'est bon pour ce TD ! Youhou ! Vous en êtes venu à bout !



Liens utiles

- <https://www.php.net/manual/fr/>
- <https://www.w3schools.com/php/default.asp>
- <https://emojipedia.org/>