

## Travaux dirigés n° 1

### Premiers pas en PHP

#### Exercice 1 (Premiers scripts)

1. Écrivez un script PHP qui affiche les 10 tables de multiplication sous la forme de tableaux distincts. La première colonne contient l'opération détaillée et la seconde le résultat.
2. Cette fois-ci, nous souhaitons utiliser 10 boutons pour les 10 chiffres. Lorsque l'utilisateur clique sur un bouton, seule la table correspondante est affichée. Nous n'utiliserons pas de formulaire.
3. Pour passer une valeur à un script PHP, nous pouvons ajouter à l'URL "?exemple=toto". La valeur "toto" est récupérée via une case du tableau associatif `$_GET : $_GET['exemple']`. Modifiez le script précédent en utilisant ce principe.
4. Reprenez la question 2 en utilisant cette fois-ci *Javascript* (uniquement pour l'affichage et le masquage des tables).
5. Pour chacune des questions précédentes, indiquez à l'aide d'un schéma où se trouvent les différents scripts (avant et après l'exécution de la requête) et où se déroule l'exécution de chaque élément.
6. Pour les solutions proposées dans les questions 2 à 4, donnez les avantages et inconvénients en termes de données échangées, requêtes effectuées, scripts à écrire, etc.

#### Exercice 2 (Nombre premier)

Nous souhaitons écrire un script PHP qui permet de calculer si un nombre saisi dans un formulaire est premier ou non.

1. Donnez un algorithme permettant de vérifier si un entier est premier. Transposez cet algorithme en PHP.
2. Nous souhaitons écrire un script qui permette à un utilisateur de saisir un entier. Le script affiche si le nombre passé est premier ou non.

#### Exercice 3 (Questions de date)

Pour gérer les dates, PHP propose une API complète comportant un grand nombre de fonctions. Nous utiliserons **uniquement** les fonctions décrites en annexes.

1. Écrivez un script qui affiche un message de bienvenue avec la date du jour en Français, jour et mois en toutes lettres.
2. Cette fois-ci, affichez "Demain nous serons le" suivi du jour du lendemain.
3. Maintenant : "Nous sommes le ?ième jour de l'année" (en remplaçant ? par le numéro du jour dans l'année).

#### Exercice 4 (Mini-forum)

Nous souhaitons écrire un script permettant à un utilisateur de laisser des messages sur une page. Il doit saisir son login et son message dans un formulaire (utilisez les balises `input` et `textarea`). Les informations saisies sont affichées à l'écran dans une balise `<div>`, le login mis en évidence en gras, par exemple.

1. Quels problèmes cela pose-t-il si l'on souhaite conserver les retours à la ligne dans le message ?
2. En utilisant la fonction `str_replace`, proposez une solution.

3. Que se passe-t-il à votre avis, si l'utilisateur tape du code HTML dans la zone de texte ?
4. Nous souhaitons en plus que le pseudo et le message soient remplis par défaut lorsque le formulaire est validé. Quels cas doit-on gérer ?
5. Donnez le script complet.
6. Nous souhaitons maintenant ajouter des statistiques sur le message : nombre de caractères, nombre de mots. Proposez une solution.
7. En tapant le message suivant "C'est un élève très doué !", la taille retournée est de 30. Comment expliquer ce résultat ?

### Exercice 5 (Tri)

Nous souhaitons développer notre propre algorithme de tri d'un tableau d'entiers (sans utiliser les méthodes de tri proposées par l'API PHP). Pour cela, nous souhaitons utiliser l'algorithme de tri par sélection/échange. Pour rappel, il faut rechercher la case du tableau contenant la valeur minimum, puis on échange la valeur de la case trouvée avec celle de la première case. On procède ensuite avec la deuxième case en recherchant le minimum entre la deuxième case et la dernière. Et ainsi de suite.

1. Écrivez la fonction `indiceMin` qui retourne l'indice de la case contenant la valeur minimum d'un tableau à partir d'un indice donné.
2. Écrivez la fonction `trier` qui trie un tableau passé en paramètre.
3. Complétez en écrivant une fonction `afficher` qui affiche un tableau PHP passé en paramètre dans un tableau HTML.
4. Donnez le reste du code de votre script qui utilise les fonctions précédentes.

### Exercice 6 (Fonctions)

Nous considérons un tableau de données dont chaque élément est un tableau associatif. Chaque élément correspond à une ligne, les indices étant les noms des colonnes. À noter que les éléments de chaque ligne peuvent être dans un ordre quelconque.

1. Donnez une fonction permettant de créer un tableau HTML à partir d'un tel tableau. Vous n'oublierez pas la ligne de titre.
2. Nous souhaitons laisser la possibilité à un utilisateur de trier les données du tableau en fonction de la colonne cliquée. En cliquant une deuxième fois, on réalise le tri inverse. Proposez une solution, sans donner le code.
3. Écrivez la fonction permettant de trier le tableau.

## Annexes - Manuel PHP

**date** : formate une date/heure locale

```
string date ( string $format [, int $timestamp = time() ] )
```

Retourne une date sous forme d'une chaîne, au format donné par le paramètre `format`, fournie par le paramètre `timestamp` ou la date et l'heure courantes si aucun *timestamp* n'est fourni. En d'autres termes, le paramètre `timestamp` est optionnel et vaut par défaut la valeur de la fonction `time()`.

Liste de paramètres :

- `format` : le format de la date désirée. Voir les quelques options de formatage ci-dessous.
- `timestamp` : le paramètre optionnel `timestamp` est un *timestamp* Unix de type entier qui vaut par défaut l'heure courante locale.

Car.	Description	Exemples
d	Jour du mois sur deux chiffres	01 à 31
j	Idem sans le 0	1 à 31
D	Jour de la semaine sur 3 lettres	Mon à Sun
l	Jour de la semaine version longue	Sunday à Saturday
N	Représentation numérique du jour de la semaine	1 (Lundi) à 7 (Dimanche)
W	Numéro de semaine	36 pour la 36ème
F	Mois version longue	January à December
n	Numéro du mois	1 à 12
t	Nombre de jours dans le mois	28 à 31
L	Année bissextile ou non	1 si oui 0 sinon
Y	Année sur 4 chiffres	2018
y	Année sur 2 chiffres	18
G	Heure	0 à 23
H	Idem avec 0 non significatif	00 à 23
i	Minutes avec les zéros initiaux	00 à 59
s	Secondes avec les zéros initiaux	00 à 59

**time** : retourne le *timestamp* UNIX actuel.

```
int time ( void )
```

**mktime** : retourne le *timestamp* UNIX d'une date.

```
int mktime ( [ int $hour = date("H") [, int $minute = date("i")
[, int $second = date("s") [, int $month = date("n")
[, int $day = date("j") [, int $year = date("Y")]]]] ] )
```

Liste de paramètres : `hour`, l'heure; `minute`, les minutes; `second`, les secondes; `month`, le mois; `day`, le jour; `year`, l'année

**setlocale** : modifie les informations de localisation

```
setlocale ( int $category , array $locale ) : string
```

Pour ce qui nous concerne : `setlocale (LC_TIME, 'fr_FR.utf8','fra');`

**explode** (version simplifiée) :

```
array explode ( string $delimiter , string $string [, int $limit = PHP_INT_MAX ] )
```

Retourne un tableau de chaînes, chacune d'elle étant une sous-chaîne du paramètre `string` extraite en utilisant le séparateur `delimiter`. Le paramètre `limit` est le nombre maximum d'éléments.

---

**htmlentities** (version simplifiée) :

```
string htmlentities ( string $string )
```

Convertit tous les caractères éligibles en entités HTML.

---

**str\_replace** :

```
mixed str_replace(mixed $search ,mixed $replace ,mixed $subject [, int &$amp;count ] )
```

`str_replace()` retourne une chaîne ou un tableau, dont toutes les occurrences de `search` dans `subject` ont été remplacées par `replace`.

*Liste de paramètres :*

- `search` : la valeur à chercher. Un tableau peut être utilisé pour désigner plusieurs masques.
  - `replace` : la valeur de remplacement à substituer aux valeurs trouvées. Un tableau peut être utilisé pour désigner plusieurs valeurs de remplacement.
  - `subject` : la chaîne de caractères ou le tableau sur lequel on va effectuer la recherche et le remplacement. Si `subject` est un tableau, alors le remplacement se fera sur chaque élément de celui-ci, et la valeur retournée sera aussi un tableau.
  - `count` : si fournie, cette variable contiendra le nombre de remplacements effectués.
- 

**strlen** :

```
int strlen ( string $string )
```

Calcule la taille d'une chaîne de caractères.

---

**utf8\_decode** :

```
string utf8_decode ( string $data )
```

Décode la chaîne `data`, en supposant qu'elle est au format UTF-8, et la convertit au format ISO-8859-1. Les octets dans la chaîne qui ne sont pas valide en UTF-8 et les caractères UTF-8 qui n'existent pas en ISO-8859-1 (qui sont, les caractères au-dessus de U+00FF) sont remplacé par ?.