

Les objectifs de cet exercice sont les suivants :

- Maîtriser les principes des algorithmes : variables, chaînes de caractères, conditions, boucles, tableaux, dates, programmation orientée objet.
- Manipuler les balises HTML de base : table, a, select, input, img, ...
- Exploiter des ressources en ligne de référence telles que : **PHP.net**, **Developpez.com** ou **W3schools.com**

I. Exercice 1

Créer une fonction personnalisée `convertirMajRouge()` permettant de transformer une chaîne de caractère passée en argument en majuscules et en rouge.

Vous devrez appeler la fonction comme suit : `convertirMajRouge($texte)` ;

Affichage (si `$texte = « Mon texte en paramètre »`)

MON TEXTE EN PARAMETRE

II. Exercice 2

Soit le tableau suivant :

```
$capitales =  
[France=>"Paris", "Allemagne"=>"Berlin", "USA"=>"Washington", "Italie"=>"Rome"];
```

Réaliser un algorithme permettant d'afficher le tableau HTML suivant (notez que le nom du pays s'affichera en majuscule et que le tableau est trié par ordre alphabétique **du nom de pays**) grâce à une fonction personnalisée.

Vous devrez appeler la fonction comme suit : `afficherTableHTML($capitales)` ;

Affichage

| Pays | Capitale |
|-----------|------------|
| ALLEMAGNE | Berlin |
| FRANCE | Paris |
| ITALIE | Rome |
| USA | Washington |

III. Exercice 3

Afficher un lien hypertexte permettant d'accéder au site d'Elan Formation. Le lien devra s'ouvrir dans un nouvel onglet (`_blank`).

IV. Exercice 4

A partir de l'exercice 2, ajouter une colonne supplémentaire dans le tableau HTML qui contiendra le lien hypertexte de la page Wikipédia de la capitale (le lien hypertexte devra s'ouvrir dans un nouvel onglet et le tableau sera trié par ordre alphabétique **de la capitale**).

On admet que l'URL de la page Wikipédia de la capitale adopte la forme :

<https://fr.wikipedia.org/wiki/>

Le tableau passé en argument sera le suivant :

```
$capitales = ["France"=>"Paris", "Allemagne"=>"Berlin",  
             "USA"=>"Washington", "Italie"=>"Rome", "Espagne"=>"Madrid"];
```

Affichage

| Pays | Capitale | Lien wiki |
|-----------|------------|----------------------|
| ALLEMAGNE | Berlin | Lien |
| ESPAGNE | Madrid | Lien |
| FRANCE | Paris | Lien |
| ITALIE | Rome | Lien |
| USA | Washington | Lien |

V. Exercice 5

Créer une fonction personnalisée permettant de créer un formulaire de champs de texte en précisant le nom des champs associés.

Exemple :

```
$nomsInput = ["Nom", "Prénom", "Ville"];  
afficherInput($nomsInput);
```

Affichage

Nom

Prénom

Ville

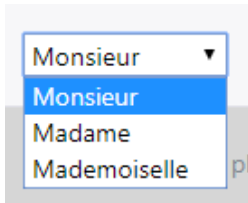
VI. Exercice 6

Créer une fonction personnalisée permettant de remplir une liste déroulante à partir d'un tableau de valeurs.

Exemple :

```
$elements = ["Monsieur", "Madame", "Mademoiselle"];  
alimenterListeDeroulante($elements);
```

Affichage



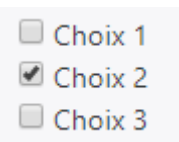
VII. Exercice 7

Créer une fonction personnalisée permettant de générer des cases à cocher. On pourra préciser dans le tableau associatif si la case est cochée ou non.

Exemple :

```
genererCheckbox($elements);  
//où $elements est un tableau associatif clé => valeur avec 3 entrées.
```

Affichage



VIII. Exercice 8

Soit l'URL suivante : <http://my.mobirise.com/data/userpic/764.jpg>

Créer une fonction personnalisée permettant d'afficher l'image N fois à l'écran.

Exemple :

```
repeterImage($url, 4);
```

Affichage



IX. Exercice 9

Créer une fonction personnalisée permettant d'afficher des boutons radio avec un tableau de valeurs en paramètre ("Monsieur", "Madame", "Mademoiselle").

Exemple :

```
afficherRadio($nomsRadio);
```

Affichage

- ☐ Masculin
- ☐ Féminin
- ☒ Autre

X. Exercice 10

En utilisant l'ensemble des fonctions personnalisées créées précédemment, créer un formulaire complet qui contient les informations suivantes : champs de texte avec nom, prénom, adresse e-mail, ville, sexe et une liste de choix parmi lesquels on pourra sélectionner un intitulé de formation : « Développeur Logiciel », « Designer web », « Intégrateur » ou « Chef de projet ».

Le formulaire devra également comporter un bouton permettant de le soumettre à un traitement de validation (submit).

XI. Exercice 11

Ecrire une fonction personnalisée qui affiche une date en français (en toutes lettres) à partir d'une chaîne de caractère représentant une date.

Exemple :

```
formaterDateFr("2018-02-23");
```

Affichage

vendredi 23 février 2018

XII. Exercice 12

La fonction `var_dump($variable)` permet d'afficher les informations d'une variable.

Soit le tableau suivant :

```
$tableauValeurs=[true,"texte",10,25.369,["valeur1","valeur2"]];
```

A l'aide d'une boucle, afficher les informations des variables contenues dans le tableau.

Affichage

```
bool(true)
string(5) "texte"
int(10)
float(25.369)
array(2) { [0]=> string(7) "valeur1" [1]=> string(7) "valeur2" }
```

XIII. Exercice 13

Comme nous l'avons vu à l'exercice précédent, les `var_dump()` vont nous servir à débbugger. Mais l'affichage peut-être un peu chaotique. Afin de l'améliorer nous allons utiliser un outil bien pratique : Xdebug

Pour l'installation de Xdebug :

- Laragon doit être démarré
- Créer un fichier `phpinfo.php` dans lequel on appelle la méthode `phpinfo()`;

```
phpinfo.php
1  <?php
2
3  phpinfo();
4
```

- exécuter le fichier `phpinfo.php` dans votre navigateur. La page vous affiche la configuration de votre version actuelle de PHP

| PHP Version 8.3.0 | |
|-----------------------------------|---|
| System | Windows NT DELLP0122-01 10.0 build 22621 (Windows 11) AMD64 |
| Build Date | Nov 21 2023 17:46:17 |
| Build System | Microsoft Windows Server 2019 Datacenter [10.0.17763] |
| Compiler | Visual C++ 2019 |
| Architecture | x64 |
| Configure Command | cscript /nologo /e:jscript configure.js "--enable-snapshot-build" "--enable-debug-pack" "--with-pdo-oci=.\\..\\..\\instantclient\\sdk,shared" "--with-oci8-19=.\\..\\..\\instantclient\\sdk,shared" "--enable-object-out-dir=.\\obj\\" "--enable-com-dotnet=shared" "--without-analyzer" "--with-pgo" |
| Server API | Apache 2.0 Handler |
| Virtual Directory Support | enabled |
| Configuration File (php.ini) Path | no value |
| Loaded Configuration File | C:\laragon\bin\php\php-8.3.0-Win32-vs16-x64\php.ini |

- CTRL + A / CTRL + C pour copier l'intégralité de la page de configuration
- Se rendre sur XDebug Wizard : <https://xdebug.org/wizard>
- Dans la zone de texte CTRL + V pour coller le texte précédemment copié

Installation Wizard

This page helps you finding which file to download, and how to configure PHP to get Xdebug running. Please paste the **full** output of `phpinfo()` (either a copy & paste of the HTML version, the HTML source or `php -i` output) and submit the form to receive tailored download and installation instructions.

```
PHP logo
PHP Version 8.3.0
System Windows NT DELLP0122-01 10.0 build 22621 (Windows 11) AMD64
Build Date Nov 21 2023 17:46:17
Build System Microsoft Windows Server 2019 Datacenter [10.0.17763]
Compiler Visual C++ 2019
Architecture x64
Configure Command cscript /nologo /e:jscript configure.js "--enable-
snapshot-build" "--enable-debug-pack" "--with-pdo-
oci=../../..\..\instantclient\sdk,shared" "--with-oci8-
19=../../..\..\instantclient\sdk,shared" "--enable-object-out-dir=../obj/" "--
enable-com-dotnet=shared" "--without-analyzer" "--with-pgo"
Server API Apache 2.0 Handler
Virtual Directory Support enabled
Configuration File (php.ini) Path no value
Loaded Configuration File C:\laragon\bin\php\php-8.3.0-Win32-vs16-
x64\php.ini
See this dir for additional ini files (open)
```

- Puis appuyer sur le bouton "Analyse my phpinfo output"
- On vous précisera que XDebug n'est pas installé

• Xdebug installed: no

- Plus bas, les instructions nécessaires à l'installation vous sont précisées : télécharger le fichier .dll (forcer le téléchargement si nécessaire) et renommer le en "php_xdebug.dll", copier le fichier pour le déplacer dans le dossier spécifié (dossier ext de votre dossier PHP dans Laragon), modifier ensuite le fichier php.ini (avec un éditeur de texte de votre choix) et copier / coller la ligne "zend_extension = xdebug" en fin de document par exemple.
- Redémarrer Laragon pour prendre en compte les modifications du fichier php.ini

Pour tester si Xdebug est bien installé, tester ceci :

```
$array = ["Texte 1", "Texte 2", "Texte 3", 1000, 100.5, true, ["Texte 4", "Texte 5"]];
var_dump($array);
```

Sans Xdebug :

```
array(7) { [0]=> string(7) "Texte 1" [1]=> string(7) "Texte 2" [2]=> string(7) "Texte 3" [3]=> int(1000) [4]=> float(100.5) [5]=> bool(true) [6]=> array(2) { [0]=> string(7) "Texte 4" [1]=> string(7) "Texte 5" } }
```

Avec Xdebug (mieux non ?) :

Vous remarquez que la mise en page est beaucoup plus agréable notamment via les couleurs associées à chaque type de données et l'indentation globale

```
C:\laragon\www\xdebug\phpinfo.php:6:
array (size=7)
  0 => string 'Texte 1' (length=7)
  1 => string 'Texte 2' (length=7)
  2 => string 'Texte 3' (length=7)
  3 => int 1000
  4 => float 100.5
  5 => boolean true
  6 =>
    array (size=2)
      0 => string 'Texte 4' (length=7)
      1 => string 'Texte 5' (length=7)
```

C'est également valable pour la mise en forme des erreurs

| (!) Warning: Undefined variable \$ararray in C:\laragon\www\xdebug\phpinfo.php on line 6 | | | | |
|--|--------|--------|----------|-------------------|
| Call Stack | | | | |
| # | Time | Memory | Function | Location |
| 1 | 0.0003 | 482688 | {main}() | ...\phpinfo.php:0 |

XIV. Exercice 14

En utilisant les ressources de la page <http://php.net/manual/fr/book.filter.php>, vérifier si une adresse e-mail a le bon format.

Affichage

L'adresse **elan@elan-formation.fr** est une adresse e-mail valide

L'adresse **contact@elan** est une adresse e-mail invalide