

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 = array  
("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 = array ("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 :

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

Affichage

Nom

Prénom

Ville

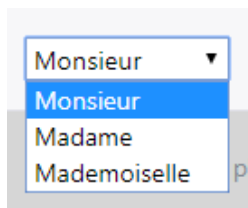
VI. Exercice 6

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

Exemple :

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

Affichage



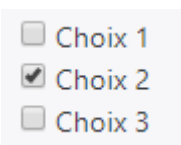
VII. Exercice 7

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=array(true,"texte",10,25.369,array("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

Créer une classe Voiture possédant les propriétés suivantes : `marque`, `modèle`, `nbPortes` et `vitesseActuelle` ainsi que les méthodes `demarrer()`, `accelerer()` et `stopper()` en plus des accesseurs (`get`) et mutateurs (`set`) traditionnels. La vitesse initiale de chaque véhicule instancié est de 0. Une méthode personnalisée pourra afficher toutes les informations d'un véhicule.

`v1` → "Peugeot", "408", 5

`v2` → "Citroën", "C4", 3

Coder l'ensemble des méthodes, accesseurs et mutateurs de la classe tout en réalisant des jeux de tests pour vérifier la cohérence de la classe Voiture. Vous devez afficher les tests et les éléments suivants :

```
Le véhicule Peugeot 408 démarre
Le véhicule Peugeot 408 accélère de 50 km / h
Le véhicule Citroën C4 démarre
Le véhicule Citroën C4 est stoppé
Le véhicule Citroën C4 veut accélérer de 20
Pour accélérer, il faut démarrer le véhicule Citroën C4 !
La vitesse du véhicule Peugeot 408 est de : 50 km / h
La vitesse du véhicule Citroën C4 est de : 0 km / h
```

Infos véhicule 1

Nom et modèle du véhicule : Peugeot 408

Nombre de portes : 5

Le véhicule Peugeot est démarré

Sa vitesse actuelle est de : 50 km / h

Infos véhicule 2

Nom et modèle du véhicule : Citroën C4

Nombre de portes : 3

Le véhicule Citroën est à l'arrêt

Sa vitesse actuelle est de : 0 km / h

Bonus : ajouter une méthode `ralentir(vitesse)` dans la classe Voiture.

XIV. Exercice 14

Créer une classe Voiture possédant 2 propriétés (`marque` et `modèle`) ainsi qu'une classe VoitureElec qui hérite (*extends*) de la classe Voiture et qui a une propriété supplémentaire (`autonomie`).

Instancier une voiture « classique » et une voiture « électrique » ayant les caractéristiques suivantes :

```
+ Peugeot 408 : $v1 = new Voiture("Peugeot", "408");
+ BMW i3 150 : $ve1 = new VoitureElec("BMW", "I3", 100);
```

Votre programme de test devra afficher les informations des 2 voitures de la façon suivante :

```
echo $v1->getInfos()."<br/>";
echo $ve1->getInfos()."<br/>";
```

XV. Exercice 15

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