# Exercices de boucles, arrays, fonctions

1. Créez une boucle **for** qui affiche 10 fois "Bonjour, je suis dans une boucle !"
2. Créez une boucle **for** qui affiche les valeurs entre 1 et 100 (incluses)
3. Créez une boucle **for** qui affiche les valeurs entre 10 et 50 (incluses)
4. Créez un array **$vals** contenant les valeurs 20,50,7,54
5. Rajoutez les valeurs 150 et 300 à l'array précédant (ailleurs de la déclaration de l'array)
6. Créez une boucle **foreach** pour afficher l'array $vals. Créez une boucle **for** pour afficher l'array **$vals**
7. Créez une fonction qui reçoit un nom et affiche "Bonjour (nom)!". Testez-la avec plusieurs noms
8. Créez une fonction **afficheTableau** qui reçoit un array et affiche son contenu. Utilisez foreach pour vous faciliter la tâche. Testez-la avec l'array **$vals**
9. Créez une fonction qui reçoit un array et renvoie un array inversé
10. Créez une fonction qui reçoit un array et le renverse (l’array original **change**, on a besoin d’utiliser une référence)
11. Créez un array associatif $film1 contenant le nom, le genre et le réalisateur d'un film. Créez $film2 et $film3 contenant les données d'autres films
12. Affichez le contenu d'un des films qu'on vient de créer en utilisant une boucle foreach
13. Créez une fonction **afficheAssoc** qui reçoit un array associatif et affiche son contenu en utilisant une boucle foreach. Testez la avec l'array que vous venez de créer
14. Rajoutez une "dateSortie" à chacun de films crées en 11)
15. Appelez la fonction **afficheAssoc**  que vous avez créé pour voir le contenu de chaque film. On devrait voir la dateSortie en plus du nom, genre et réalisateur…
16. Effacez le réalisateur du $film1. Affichez le contenu du film1 pour vous assurer que le réalisateur n'est plus là
17. Créez une fonction **soustraction** qui renvoie la soustraction de deux valeurs reçues en paramètre. Ex: soustraction (25,15) va renvoyer la valeur 10
18. Modifiez la fonction précédente pour que, avant de renvoyer le résultat de la soustraction, elle affiche "Le résultat est négatif!" si ce résultat est négatif
19. Créez une fonction **afficheValeurs** qui affiche tous les entiers qui se trouvent entre deux valeurs reçues comme paramètre. Par exemple:

afficheValeurs (40,100) affichera toutes les valeurs entre 40 et 100

afficheValeurs (20,80) affichera toutes les valeurs entre 20 et 80 etc…

1. Le code suivant nous posera de problèmes. Pourquoi? Qu'est-ce qu'on peut faire pour l'arranger?

$i=0;

while ($i<15){

echo "Salut à tous! ".$i;

}

1. Le code suivant nous posera de problèmes. Pourquoi? Arrangez-le

$tab= array(4,5,1,7);

for ($i=0;$i<10;$i++){

echo $tab[i];

}

1. Créez un array contenant 5 noms d'acteurs
2. Créez une fonction **afficheTableau** qui affiche le contenu d'un tableau indexé (pas associatif). Appelez cette fonction pour afficher le contenu du tableau d'acteurs
3. Créez un array associatif contenant de couples personne-âge
4. Affichez le contenu avec une boucle foreach
5. Créez un array contenant le nom de 4 langues
6. Changez le contenu de la position 3 par "Danois"
7. Créez une boucle (un for, un foreach et un while) pour afficher le contenu de l'array
8. Amélioriez les boucles for et while précédents en utilisant la fonction **count**
9. Effacez l'element de la position 4 dans l'array en utilisant la fonction **unset**
10. Utilisez la fonction **array\_search** avec deux paramètres pour afficher la position d'un certain élément dans l'array et "Element pas trouvé" dans le contraire
11. Créez un array vide $languesCopie. Créez une boucle qui rajoute chaque element de l'array $langues dans l'array $languesCopie
12. Créez les fonctions suivantes
    1. **afficher** qui reçoit un array indexé (pas associatif) et affiche ses valeurs
    2. **afficherHTML** qui reçoit un array indexé (pas associatif) et construit un petit tableau HTML contenant les valeurs de l'array
    3. **afficherCase** qui reçoit deux parametres:
13. un array indexé (pas associatif) et affiche ses valeurs
14. un string **$case** contenant le mot "majuscules" ou "minuscules"

La fonction afficherCase doit afficher les valeurs de l'array en majuscules ou minuscules selon la valeur du paramètre $case. Pour ce faire, regardez la documentation des fonctions **strtoupper** et **strtolower**

# Exercices de création de fonctions (+arrays)

1. Développez une fonction qui crée un array de 25 entiers aléatoires (utilisez mt\_rand) et l'affiche
2. Modifiez la fonction précédente pour qu'elle puisse recevoir le nombre de valeurs qui contient l'array
3. Développez une fonction capable de créer un array de 25 entiers aléatoires (utilisez rand) et le renvoyer. Testez-la.
4. Développez une fonction capable de créer un array d'un nombre aléatoire de valeurs (entre 20 et 40) et le renvoie. Affichez son contenu dans le script principal, ainsi que le nombre d'éléments de l'array
5. Créez une fonction qui reçoit un array associatif et affiche son contenu dans un tableau HTML
6. Créez une fonction qui reçoit un array associatif et renvoie un nouvel array contenant uniquement ses clés
7. Modifiez l'exercice précédent pour que la fonction renvoie un array contenant les clés tries par ordre alphabétique
8. Créez une fonction qui reçoit un array associatif et renvoie un nouvel array contenant uniquement ses valeurs

# Exercices de fonctions d'Arrays natives

1. Affichez les chiffres pairs entre 0 et 100. Affichez les chiffres impairs entre 0 et 100
2. Affichez un tableau HTML contenant trois colonnes : la première contient de chiffres pairs, la deuxième de chiffres impairs et la troisième de chiffres aléatoires entre 0 et 100 (fonction **rand**, voir la documentation PHP sur le net). Créez les en-têtes pour les colonnes
3. Créez un array contenant les chiffres pairs entre 0 et 100 en utilisant
   * Une boucle for
   * Une boucle while
4. Créez un array d'un nombre aléatoire d'entiers (min. 10 valeurs et max. 30 valeurs).

Affichez l'array et le nombre d'éléments de l'array.

1. Utilisez **array\_slice** pour créer un nouveau tableau contenant les 10 premières valeurs de l'array précédant (array\_slice n'efface pas des valeurs de l'array original)
2. Triez les éléments de l'array en utilisant la fonction **sort**. Comment est-ce qu'on pourrait trier les valeurs en ordre descendant ? Cherchez la fonction dans la documentation de PHP. Quel type d'array vous avez utilisé, indexé ou associatif ?
3. Créez un array associatif contenant des pays-capitales. Affichez son contenu en utilisant foreach
4. Rajoutez deux nouveau pays à votre array et imprimez son contenu en utilisant **print\_r**
5. Faites la même chose en utilisant array\_push()
6. Triez l'array en utilisant **asort**. Pourquoi ce n'est pas une bonne idée d'utiliser **sort** dans ce cas ?
7. Trouvez la fonction pour trier l'array par pays au lieu de par villes
8. Créez une boucle pour afficher un tableau HTML contenant les pairs ville-pays. Les villes s'afficheront en gras
9. Utilisez la fonction **array\_slice** pour extraire les trois premiers pays de votre array et les stocker dans un nouvel array
10. Créez un array pour stocker les informations d'un mp3 (nom, auteur, style, durée).

Créez plusieurs "morceaux" de cette manière

1. Créez une boucle capable d'afficher les infos d'un morceau dans un tableau HTML
2. Créez un array bidimensionnel $disque contenant les morceaux que vous venez de créer
3. Rajoutez un nouveau morceau au disque
4. Utilisez **array\_slice** pour extraire les deux premiers morceaux
5. A partir de l'array de l'exercice 4, utilisez **array\_slice** pour
   * Eliminer les valeurs à partir de la vingtième position de l'array
   * Insérer les valeurs 99 et 100 juste avant la 5ème position de l'array
   * Remplacer les valeurs dans les positions 2 et 3 par 2000 et 3000

Utilisez print\_r pour vous assurer que les résultats sont corrects

1. Utilisez la fonction **explode** pour transformer un string contenant de valeurs séparées par virgules en un array
2. Utilisez la fonction **implode** pour créer un string à partir des valeurs d'un array
3. Utilisez la fonction **array\_merge** pour fusionner deux arrays
4. Créez un array vide $hobbies

* Rajoutez des hobbies au tableau
* Affichez le tableau
  + 1. Avec print\_r
    2. Avec var\_dump
    3. Avec une boucle foreach
    4. Avec une boucle for
    5. Avec une boucle while
    6. Avec une boucle for en utilisant la fonction **count**
    7. Avec une boucle while en utilisant la fonction **count**
    8. En commençant par le dernier élément avec un while (utilisez aussi **count**)

1. Créez un array associatif $acteur1 contenant les couples

Nom – Van Damme

Prenom – Jean Claude

* + Rajoutez le couple clé-valeur "Nationalité – Belgique" au tableau $acteur1
  + Affichez le tableau $acteur1 en utilisant un boucle foreach qui
    1. affiche les clés (Nom, ….) et les valeurs de l'array (Van Damme…)
    2. affiche uniquement les clés de l'array
    3. affiche uniquement les valeurs de l'array
  + Rajoutez quelques acteurs qui vous aimez bien au tableau
  + Affichez le tableau
    1. Avec print\_r
    2. Avec une boucle foreach en affichant uniquement les valeurs (Van Damme, Jean Claude, Belgique) de l'array
    3. Avec une boucle for
    4. Avec une boucle while
    5. Avec une boucle for en utilisant la fonction **count**
    6. Avec une boucle while en utilisant la fonction **count**
    7. En commençant par le dernier élément avec un while (utilisez aussi **count**)

1. La fonction **array\_map** nous permet d'appliquer une fonction ("callback") de notre choix à chaque élément de l'array (cherchez vous-mêmes la documentation de cette fonction).

La fonction **reçoit** deux paramètres :

* + Le nom de la fonction qu'on va appliquer sur chaque élément (qui doit être créé préalablement)
  + L'array sur lequel on veut appliquer la fonction

La fonction **renvoie** :

* + L'array modifié par la fonction