# Exercices formulaires (I)

1. Créez un site contenant un formulaire où l'utilisateur saisira son prénom, nom et âge. Le site doit afficher "Bienvenu XXXX XXXX, vous avez XXXX ans!"
2. Créez une page contenant un formulaire où on saisit un nom et un prénom. Un autre script traitera les données envoyées et affichera un message de salutation. Si les valeurs rentrées correspondent à votre nom et prénom, le script affichera le message "Que la force soi avec toi." et un lien vers le site officiel de Star Wars
3. Créez un site où l'utilisateur doit deviner un chiffre au hasard entre 0 e 10 (ce chiffre est créé par l'ordinateur en utilisant la fonction mt\_rand()). Si la personne gagne, affichez une photo comme prix. Conseil : Apprenez à utiliser la fonction avant d'essayer de créer le site
4. Créez un script contenant un formulaire où on saisit deux valeurs entières. Un autre script traitera les données envoyées et affichera toutes les valeurs entières qui existent entre les deux dans un tableau HTML
5. Créez un jeu de lotto en PHP. Stockez les valeurs gagnantes dans un array.
6. Créez un formulaire qui permette de saisir un prix et le rajoute une TVA du 12%. L’utilisateur cliquera sur un bouton et le résultat du calcul sera affiché dans une autre page.
7. Créez un formulaire qui nous permette de choisir le nom d'un animal d'une liste déroulante. Quand on envoie les données, une image de l'animal choisi sera affichée. Les images se trouvent dans le serveur
8. Créez un formulaire qui nous permette d'introduire un prix et de calculer le prix TVAC. La TVA est choisie par l'utilisateur parmi les valeurs d'une liste déroulante. Forcez (HTML) à l'utilisateur à saisir une valeur numérique.
9. Pour une application web de réservations de chambres dans un hôtel nous avons besoin d'un formulaire permettant de choisir le nombre de chambres de chaque type (simples, doubles, suites) souhaitées. Une fois les données sont envoyées, on calcule le prix total de la réservation.
10. Nous avons une liste de produits dans un array. Créer un code PHP capable d'afficher une liste déroulante contenant tous ces éléments.
11. Remplacez la liste déroulante pour des images de chaque produit avec de radio buttons
12. Créez un formulaire où l’utilisateur peut saisir un prix. Le formulaire a aussi une liste déroulante contenant de possibilités de paiement et un bouton « Payer ». Quand l’utilisateur clique sur le bouton payer on affiche le prix final : pour Mastercard on rajoute 5 euros au prix et pour VISA 3 euros).
13. Créez un formulaire qui permette d'introduire un prix et le rajouter la TVA. Cette fois ci l'utilisateur choisit la valeur de TVA parmi plusieurs valeurs en utilisant des radios-buttons
14. Créez un formulaire où on saisit un prix. Le formulaire doit avoir aussi une caisse à cocher "Réduction" qui permet d'appliquer une réduction d'un 10% au prix
15. Créez un formulaire contenant un prix. La page doit contenir deux radio buttons "Envoyer à domicile" ou "Retirer sur place". Rajoutez un bouton "Envoyer" qui nous permettra d'envoyer les données du formulaire et afficher le prix. Si le client demande l'envoi à domicile le prix sera augmenté de 5 euros
16. Créez un générateur de langue de bois (petit et simple) similaire à <http://www.encyclopedie-incomplete.com/?Le-Pipotron-Generateur-Automatique>

Si vous voulez un défi plus grand, essayez d'implémenter l'Enrhubatron!

1. Créez un formulaire qui permette à un utilisateur d'introduire son login et son password.
   * Aucun de deux ne peut être vide ;)
   * Les deux doivent contenir uniquement des lettres et des numéros et avoir entre 8 et 30 caractères

Si les valeurs sont correctes, le script les affiche en majuscule. Utilisez les fonctions **ctype, strlen** et **strtoupper** pour vous faciliter la tâche

# Exercices formulaires (II)

## Exercice 1

(Génération des éléments d'un formulaire HTML en PHP)

Nous voulons faire un site qui affiche de belles photos de chats. Nous avons une petite bibliothèque de 10 images pour le moment, mais on compte rajouter de nouvelles.

Créez un formulaire où on peut juste saisir le nombre de photos qu'on veut afficher. Quand on clique un bouton, le site nous affichera le nombre de photos qu'on a choisi.

Astuce: Stockez les photos dans un même dossier (/img/chats) sous le nom img1.jpg, img2.jpg… etc.

Important: Gênerez le HTML qui affiche les images en utilisant du PHP, car ça rendra très facile de changer le nombre d'images sans modifier une seule ligne de code.

## Exercice 2

(Génération des éléments d'un formulaire HTML en PHP)

Pour une application d'achat de billets d'avions on a besoin d'un formulaire où on indique le nombre de passagers qui prendront le vol. Une fois qu'on a choisi le nombre de passagers, notre site affichera un nouveau formulaire où on peut introduire le nom, prénom et date de naissance de chaque passager.

Dans ce nouveau formulaire il y aura un bouton "Envoyer" qui renvoie les données vers une autre page php. Dans cette dernière page, affichez le contenu reçu du formulaire ($\_POST) pour vous assurer que les données ont été bien envoyées.

## Exercice 3

Créez un formulaire qui demande l'âge à l'utilisateur. Si l'utilisateur a moins de 18 ans vous le redirigerez vers :

Si l'utilisateur a plus de 18 ans redirigez-le vers :

<https://www.youtube.com/watch?v=5Cb3ik6zP2I>

Utilisez la fonction :

header ("location: <http://www.lesite.com>");

## Exercice 4

Un utilisateur saisit le nom d'un animal et on affiche plusieurs images de cet animal qui se trouvent dans le serveur

Astuce: les images se trouvent dans le dossier **img** du serveur et portent le nom de l’animal plus un index. Ex: chat1.jpg, chat2.jpg, chat3.jpg, chien1.jpg, chien2.jpg, chien3.jpg

## Exercice 5

Modifiez l'exercice précédent pour que les noms de fichier contenant les images soient stockés dans un array en php.

L'utilisateur saisi le nom d'un animal et on doit chercher toutes les images qui contiennent ce nom (ex: chatmarrant.jpg, tristechat.jpg) et les afficher

## Exercice 7: cinéma

Un cinéma a établi plusieurs tarifs pour ses clients en fonction de l'âge et de son statut (étudiant ou pas étudiant). Ils offrent un site web qui permet aux clients potentiels de calculer le prix de ses entrées. Si le client a moins de 12 ans, le prix sera de 4 euros. Le prix normal est de 8 euros sauf pour les étudiants, qui ont une réduction du 20%.

## Exercice 8: frais d'essence

Créez un formulaire qui permette à l'utilisateur de calculer le prix d'un trajet en voiture d'un certain nombre de km. L'utilisateur peux choisir entre une voiture diesel où essence. Considérez que le prix/km moyen d'une voiture diesel est de 0.25euros/km et que le prix d'essence est de 0.30 euros/km.

## Exercice 9: listes déroulantes

1. Créez un formulaire qui nous permette de choisir un pays d'une liste déroulante. Quand on clique sur un bouton il nous affichera un drapeau du pays ainsi que le nom de sa capitale.

Étant donné qu'on ne dispose pas de bases de données, utilisez des array associatifs dans le script qui reçoit les données

1. Créez un formulaire qui nous permette de choisir un pays d'une liste déroulante.

Quand on clique sur un bouton il nous redirigera vers le site Wikipédia contenant l'information de ce pays (utilisez la fonction "header")

1. Modifiez le formulaire précédant pour pouvoir choisir la langue en plus du pays.

## Exercice 9

Nous avons classifié le temps de différents pays dans les catégories Chaud, Froid, Gelé.

Chaud : Maroc, Mexique, Espagne

Froid : Belgique, Angleterre, Holland

Gelé : Norvège, Islande, Finlande

L’utilisateur va choisir un pays d’une liste dans un formulaire et on va lui indiquer s’il fait chaud, froid ou gelé. Vous pouvez inclure une image illustrant le temps du pays.

## Exercice 10

Créez un formulaire de login pour votre site. L’utilisateur doit saisir un login et un mot de pass.

* Aucun de champs ne peut pas être vide
* Le login doit avoir 10 caractères au maximum

Réalisez la vérification en PHP. Vous pouvez rajouter un control dans le propre HTML en utilisant l’attribut « pattern » des inputs.

## Exercice 11

Créez le jeu pierre-papier-ciseaux en PHP. L’utilisateur joue contre l’ordinateur.

## Exercice 12

Nous allons créer un site capable de "deviner" nos gouts de TV. L'utilisateur du site doit saisir un mot clé sur une thématique qui lui intéresse et il sera redirigé automatiquement vers le site d'une série qui pourra lui plaire (utilisez "header")

Game of Thrones

Los Soprano

House

Breaking Bad

The wire

True Detective

Dexter

Fringe

# Exercices: Fonctions de base personnalisées

1. Créez une fonction qui nous dise bonjour
2. Créez une fonction qui calcule et affiche la somme de deux valeurs
3. Créez une fonction qui affiche la table de multiplication d'une valeur envoyée en paramètre
4. Créez une fonction qui reçoit une chaine de caractères et l'affiche en bleu dans un titre principal (H1)
5. Créez une fonction qui reçoit un texte et l'affiche dans un titre principal dans une couleur choisie
6. Créez une fonction qui reçoit un array et affiche ses éléments

(extra : affichez les éléments dans un tableau HTML)

1. Créez une fonction qui reçoit un prix en paramètre et affiche le prix TVAc
2. Créez une nouvelle version de la fonction précédente pour qu'elle renvoie la valeur TVAc calculée (cette fonction n’affiche rien).
3. Nous voulons créer un jeu de lotto. Développez une fonction valeursLotto() qui affiche 6 valeurs entières aléatoires (utilisez la fonction déjà existante mt\_rand)
4. On veut utiliser cette fonction dans plusieurs jeux différents. Créez une nouvelle version de la fonction précédente capable de recevoir le nombre de valeurs à générer
5. Nous avons un array contenant de prix. Utilisez la fonction qui calcule la TVA pour afficher tous ces prix avec la TVA
6. Nous avons un panier dans un tableau sous la forme **produits=>prix**:

Calculer le total du panier avec une fonction.

1. Créez un array de prix de vols. Créez une fonction qui reçoit un array et crée un tableau HTML contenant tous les prix. Les prix qui dépassent 300 euros seront affichés en rouge
2. Créez une fonction qui génère un array contenant un nombre aléatoire de prix (max 20) dont la valeur est aussi aléatoire (entre 100 et 800 euros)
3. Créer une nouvelle version de la fonction de 12) qui, au lieu d'afficher les valeurs dans la fonction même, stocke les valeurs dans un array qui est renvoyé (return).
4. Créez une fonction qui reçoit un array associatif et affiche son contenu dans un tableau HTML. Le texte sera affiché dans la couleur qu'on envoie en paramètre à la fonction
5. Créez une fonction qui reçoit un array de noms et renvoie un nouvel array contenant tous les noms en majuscule (paramètre par référence). Affichez ce nouvel array dans le script principal
6. Créez une fonction qui reçoit un array de noms et modifie son contenu en mettant tous les noms en majuscule (paramètre par référence). Affichez l'array original modifié dans le script principal
7. Créez une fonction qui reçoit un array associatif contenant de couples "VilleDepart-VilleDestination" et renvoie un nouvel array trié par ses indexes (les villes de départ)
8. Créez une fonction qui reçoit un array associatif et renvoie un nouvel array trié par ses valeurs (les villes de destination)
9. Nous voulons générer de noms complets de personnages pour un jeu de vidéo. Créez une fonction capable de générer un array de noms complets à partir d'un array de noms et un autre de prénoms.

Note: Pour éviter de générer à chaque fois les mêmes couples, utilisez la fonction shuffle, capable de mélanger au hasard les éléments d'un array

1. Nous voulons programmer un jeu de cartes en PHP et on doit stocker les noms des cartes dans un array. Trouvez la manière la plus efficace de le faire.
2. Utilisez la fonction array\_shift pour prendre une carte au hasard
3. Créez une fonction qui reçoit un array associatif et renvoie un nouvel array contenant uniquement ses valeurs
4. Créez une fonction qui reçoit un array associatif et renvoie un nouvel array contenant uniquement ses clés (indexes)
5. Vous avez reçu un array contenant de noms de villes en minuscules. Créez une fonction qui vous affiche les éléments de l'array en majuscules dans une liste HTML
6. Répétez l'exercice précédent, mais cette fois la première lettre de chaque ville sera en majuscule (chercher des infos sur la fonction uc\_first)

# Exercices Strings + fonctions de Strings + Formulaires

1. Créez une fonction qui reçoit un array de strings (ex: noms de personnes) et affiche chaque string ainsi que son nombre de caractères (utilisez **strlen**)
2. Vous êtes en train de faire un moteur de recherche de vols. L’utilisateur saisie uniquement une ville de départ dans un formulaire. Quand il clique, l’application affiche le nom de toutes les villes auxquelles on peut voyager à partir de la ville choisie. Pour simuler la BD, créez un array contenant quelques villes de départ et leurs destinations :

$vols = **array** ("Bruxelles"=>**array** ("Paris", "Londres", "Bucarest"),

"Rome"=>**array** ("Paris","Varsovie","Milan"),

"Pekin"=>**array** ("New York","Istambul","Seoul"));

1. Modifiez l’exercice précédent pour que l’utilisateur puisse saisir des espaces avant ou après la ville sans affecter le résultat
2. Modifiez l’exercice précédent pour que l’utilisateur puisse saisir la ville en majuscules et minuscules sans affecter le résultat
3. Créez une fonction qui reçoit un array contenant de noms d'utilisateurs et qui affiche l'information dans un tableau HTML. Si un nom a plus de 10 caractères il sera affiché en rouge
4. Nous avons déjà une liste de noms (strings) dans un array. Nous voulons créer un nouvel array contenant uniquement les trois premières lettres de chaque nom. Utilisez la fonction **substr**.
5. Nous voulons créer des adresses mail pour les utilisateurs de notre site. Nous avons un tableau $noms et un tableau $prenoms. Créez une fonction qui renvoie un tableau contenant des adresses de la forme *prenom.nom@monsite.be*

Ex: herbie.hancock@monsite.be

Affichez le tableau contenant les adresses dans le code du script principal

1. Modifiez l'exercice précédent pour que la fonction affiche son résultat toujours en majuscules. Cherchez sur le net comment mettre en majuscules un string en php.
2. Créez une fonction qui reçoit un texte et cherche le mot "bombe" à l'intérieur du texte reçu. Si le mot s'y trouve, un message de "Vous êtes un terroriste! La NSA vous observe!" sera affiché. Utilisez la fonction **strstr**, regardez la documentation pour comprendre son fonctionnement
3. Nous sommes en train de développer une application pour smartphone et on veut créer une fonction qui reçoit array contenant de noms de personnes, affiche tous les noms qui commencent par une lettre reçue en paramètre (utilisez **substr**)
4. Créez une fonction qui reçoit un array de strings (ex: noms de pays) et transforme toutes ses valeurs en majuscules ou minuscules selon la valeur d'un deuxième paramètre (true ou false)
5. Créez une fonction qui reçoit un array de strings et une lettre. La fonction renvoie un array contenant tous les valeurs de l'array qui commencent par cette lettre (utilisez **strpos**)
6. Créez une fonction qui reçoit un texte et un mot. La fonction affiche le nombre d'occurrences du mot dans le texte (**str\_count**)
7. Créez une fonction qui reçoit un fragment de texte et deux mots en paramètre. La fonction remplace le premier mot par le deuxième dans le texte et le renvoie (str\_replace)
8. Créez une fonction qui reçoit un array de noms de produits et transforme la première lettre de chaque nom en majuscule (**ucfirst**)
9. Utilisez **printf** pour imprimer la valeur d'un string et d'un entier à votre choix
10. Créez une fonction qui reçoit un array de noms strings et renvoie un array contenant la taille de chaque string de l'array original
11. Créez une fonction qui reçoit un array de strings et renvoie un array associatif contenant les strings de l'array et la taille de chaque string

# Exercices: Arrays + fonctions Array

1. Afficher un tableau HTML contenant trois lignes de 10 valeurs: la première contient de chiffres pairs (commencez par 0), la deuxième de chiffres impairs (commencez par 1) et la troisième de chiffres aléatoires entre 0 et 100 (fonction **ht\_rand**, voir la documentation PHP sur le net).
2. Vous avez cet array :

$disque1= array(

‘titre’=> ‘Hard Groove’,

‘groupe’=>’Roy Hargrove’,

‘style’=> ‘Jazz-Funk’)

Utilisez la fonction **implode** pour créer une requête SQL a la forme :

INSERT INTO disques (titre, groupe, style) VALUES (‘Hard Groove’, ‘Roy Hargrove’, ‘Jazz-Funk’)

1. 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 villes-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 3, utilisez array\_splice pour
   1. Eliminer les valeurs à partir de la vingtième position de l'array
   2. Insérer les valeurs 99 et 100 juste avant la 5ème position de l'array
   3. 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

# Exercices sur les sessions

1. Programmez le jeu pierre, papier et ciseaux. Utilisez la session pour stocker les ponctuations des joueurs (utilisateur vs. Machine)

Modifiez le jeu précédant : si un utilisateur gagne plus de 5 fois, on le redirige vers un script qui affiche l'écran de la victoire (fonction header)

1. Créez une page contenant une barre de navigation avec deux liens : login et contenu.

L’utilisateur doit être connecté pour visualiser le « contenu » du site (ex : une page contenant des affiches de films)

Pour se connecter, l’utilisateur clique sur login et il sera redirigé vers une page pour saisir son login et son password.

Si le login est « WEB » et le password est « MASTER » il sera redirigé vers la page du contenu. Sinon, il sera redirigé vers la page de login.

Rajoutez maintenant un lien « logout » à la barre de navigation. Ce lien est uniquement visible si l’utilisateur a fait login correctement.

1. Créez un formulaire simple contenant une liste déroulante remplie de noms de produits. Chaque fois que l'utilisateur choisit un produit et clique sur un bouton "Accepter" le produit choisit est stocké dans le champ de session "nomProd" comme un string. La valeur sera écrasée chaque fois qu'on clique sur "Accepter".
2. L'exemple précèdent nous permet de stocker uniquement un produit, ce qui n'est pas du tout pratique. Au lieu de stocker la valeur d'un seul produit en forme de string on va stocker un array dans la session contenant tous les produits.

Le contenu de cet array est mis à jour à chaque fois qu'on choisit un produit dans le formulaire.

1. Créez un site qui affiche une liste de produits et leur prix. Un bouton à coté de chaque produit permet à l'utilisateur de le rajouter au panier. Chaque fois qu'on rajoute un produit on peut voir le contenu actuel du panier.
2. Dans l'exercice précèdent, rajoutez les contrôles nécessaires pour permettre à l'utilisateur d'introduire le nombre d'unités de chaque produit.
3. Rajoutez un bouton dans le script principal qui vous permette d'afficher une liste de produits choisis ainsi que le prix total de la commande

# Exercice sur la master page

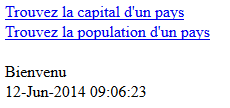
**Objectif:** Créer une **master page** **index.php** contenant le squelette d’une page (en-têtes, barre de navigation et pieds). On pourra charger le contenu d’autres pages dans la section centrale de la master page.

### Navigation

1. Le contenu de la section centrale Créez un script **nav.php** dans le dossier **include** qui contient deux liens **html**. Les deux liens pointent ver la master page (**index.php)** qu'on n'a pas encore crée. Chaque lien pointe vers index.php en rajoutant un paramètre **page** à l'URL, ce qui permet à la page index.php de charger un script ou l'autre.
2. Dans notre cas, on va avoir deux scripts possibles:

**index.php?page=Capitales**

**index.php?page=Populations**



1. La page index.php chargera un script paysCapitales.php quand le parametre page est "Capitales"ou l'autre selon la valeur de ce paramètre "page" reçu dans l'URL

### Contenu du site

Dans la section centrale de la master page on pourra charger la page paysCapitale.php ou paysPopulation.php.

1. Créez un script **paysCapitale.php** contenant un formulaire où on peut introduire le nom d'un pays. Ce formulaire envoie le nom à un autre script **viewCapitale.php** qui affiche la capitale de ce pays

(utilisez un array associatif contenant de couples pays=>capital dans le deuxième script)

1. Créez un script **paysPopulation.php** contenant un formulaire où on peut introduire aussi le nom d'un pays. Ce formulaire envoie le nom saisi à un autre script **viewPopulation.php** qui affiche la **population** de ce pays

(utilisez un array associatif contenant de couples pays=>population dans le deuxième script)

### Master Page

**Vous allez créer un master page index.php. Cette page charge ('include') un script ou l'autre selon la valeur d'un paramètre "page" qu'elle reçoit dans la URL.** Dans notre cas, ces deux scripts sont paysCapitales.php et paysPopulations.php.

Pour créer la masterpage **index.php**:

1. Créez les étiquettes de base <html>, <body> et <head>
2. Faites un **include** de paysPopulation.php ou paysCapitale.php **selon la valeur du paramètre "page" reçu dans l'URL ($\_GET['page'])**.
   * Si **page** est vide, affichez juste un message "Bienvenu".
   * Si **page** vaut "Capitales", faites un **include** de la page **paysCapitales.php**
   * Si **page** vaut "Populations", faites un **include** de la page **paysCapitales.php**

Note: Vous pouvez effacer maintenant les étiquettes <html>, <body> etc… du reste de scripts car elles se trouvent déjà dans le fichier index.php.

### Header et Footer

Vous pouvez créer des scripts contenant les en-têtes et les pieds communs à plusieurs pages du site pour ne pas devoir répéter le code à chaque fois. Ex:

1. Créez le script footer.php qui affiche la date et l'heure actuelle. Ce script sera inclus dans index.php

# Exercice sur les BDs

1. Créez un site contenant une page d’inscription et une autre de login. L’utilisateur remplira un formulaire contenant son nom, login et password. Il pourra aussi uploader une photo de profil. Toutes les données concernant l’utilisateur seront encodées dans un tableau de la BD.
   1. Rajoutez la vérification du mot de pass dans l’inscription (l’utilisateur doit re-taper son mot de pass pour être inscrit.
   2. Sécurisez les mots de passe du site en utilisant les fonctions **password\_hash** et **password\_verify**
   3. Filtrer les données du $\_POST en utilisant **filter\_input**
   4. Utilisez de variables de session pour que l’utilisateur reste connecté sur le site. Vérifiez qu’elles fonctionnent correctement
   5. Créez une page de logout
   6. Créez une page qui puisse permettre à l’utilisateur de changer sa photo de profil
   7. Créez une page pour supprimer le compte de l’utilisateur qui est connecté