

## TP n°2 : PHP (2)

1. Déterminer le numéro de version de PHP, le nom du système d'exploitation du serveur ainsi que la langue du navigateur du poste client, en utilisant **les constantes prédéfinies** de PHP.
2. Ecrire un petit script qui teste si un nombre (déclaré directement dans le programme) est à la fois un multiple de 7 et de 3.
3. Ecrire un script qui indique **au bout de combien de coups** un nombre tiré aléatoirement est à la fois un multiple de 7 et de 3.
4. **Modularité.** Préparer trois fichiers : begin.php, end.php et calculette.php. begin.php contiendra l'entête (`<!DOCTYPE html ...> <html...><head>`, etc. jusqu'à `<body>`) ; end.php contiendra la fermeture de body et de html et calculette.php inclura (include ou require ?) les deux autres fichiers php et ne contiendra qu'un titre de niveau `<h1>` pour l'instant.
5. **Calculette.** Ecrire dans le fichier calculette.php un script permettant de faire les 5 opérations usuelles : +, -, \*, / et %. On utilisera un formulaire HTML avec deux `input` de type `text` pour les 2 nombres à manipuler, un `select / option` (`<select name="..."> <option value="...">...</option> ... </select>`) pour choisir l'opérateur et un bouton `submit` pour lancer le calcul. On enverra les données en méthode GET puis POST (récupérables dans la variable `$_GET` (tableau associatif) ou `$_POST`) et on observera la différence. NB : le traitement de l'opération à effectuer en PHP et le formulaire seront tous deux mis dans le même fichier calculette.php.
6. Reprendre l'exercice de la calculette et ajouter des **tests** pour *éviter les erreurs* (division par zéro, par exemple, valeurs non positionnées dans les variables du formulaire, etc.). On lira attentivement la documentation sur les Exceptions (<https://www.php.net/manual/en/language.exceptions.php> en prenant notamment l'exemple n°2 et [https://www.w3schools.com/php/php\\_exception.asp](https://www.w3schools.com/php/php_exception.asp)). Isoler le calcul proprement dit dans une **fonction**.
7. **Réflexion** : imaginons un jeu dans lequel on demande à l'utilisateur de deviner quelque chose (un nombre, un mot, etc. cf. le pendu, par exemple). Comment s'y prendrait-on ? Expliquer en français les étapes. Analyser ensuite les échanges client/serveur. Est-ce que le PHP serait le langage le plus adapté pour cela ? Pourquoi ? Quel autre langage pourrait être utilisé si PHP n'est pas le plus adapté ?
8. **Sessions.** Créer un fichier login.php dans lequel on créera 3 variables de session : `$_SESSION['nom']`, `$_SESSION['prenom']` et `$_SESSION['motPasse']` (valeurs par défaut). Ajouter un formulaire dans lequel on demande à l'internaute ces trois informations. Le bouton validation pointera vers une page `verif.php`. Vérifier, en les affichant, que les 3 variables de session sont bien visibles et connues du script `verif.php`.
9. **Jeu : nombre à deviner.** Ecrire un programme en PHP qui tire un nombre entier aléatoirement et qui demande à l'utilisateur de le deviner (via un formulaire). Si le nombre proposé par l'utilisateur est inférieur au nombre aléatoire, le programme répond « inférieur », s'il est supérieur, il répond « supérieur ». S'il est égal, il répond « gagné » et indique le nombre de coups qu'il a fallu jouer pour trouver le bon nombre.
10. **Cookies.** Reprendre le fichier login.php en le dupliquant. Y ajouter le traitement des données saisies dans le formulaire (nom, prenom, motPasse) par des cookies. Afficher ensuite le résultat des cookies dans une autre page PHP (nommée `cookies.php`) pour vérifier le bon fonctionnement. Enfin, modifier dans le fichier `cookies.php` la valeur du cookie `prenom` en mettant **Emilie** à la place.
11. **Fichiers.** Le but est d'écrire dans un fichier les noms et prénoms entrés dans le formulaire précédent.  
Reprendre le fichier `login.php` -- en faisant une copie -- et modifier le traitement associé au formulaire : il s'agit donc de sauvegarder chaque couple (nom, prenom), à raison d'un couple par ligne et en insérant une virgule entre le nom et le prenom, dans un fichier nommé `nomInternaute.txt` à enregistrer dans le même répertoire que celui dans lequel on travaille. Après le traitement, vérifier que le fichier se remplit correctement en créant une page nommée `nomInternaute.php` qui affiche la liste des internautes, ligne par ligne.