

Exercices PHP – Bases de la programmation

1. Variables et affichage

Exercice 1.1 – Affichage d'un prénom

Objectif : Déclarer une variable et afficher un message.

- Créez une variable appelée `$prenom` contenant votre prénom entre guillemets (chaîne de caractères).
- Utilisez `echo` pour afficher le message suivant :

Bonjour, [prénom] !

Ce qu'il faut utiliser : une variable, une chaîne de caractères, `echo`.

Exercice 1.2 – Addition de deux nombres

Objectif : Manipuler des variables numériques et effectuer une addition.

- Créez deux variables :
 - `$a` contenant un nombre entier (ex. : 4)
 - `$b` contenant un autre nombre entier (ex. : 7)
- Calculez leur somme et stockez-la dans une variable `$somme`.
- Affichez :

La somme de 4 et 7 est : 11

Ce qu'il faut utiliser : variables numériques, opérateur `+`, `echo`.

Exercice 1.3 – Présentation personnelle

Objectif : Concaténer des chaînes de caractères avec des variables.

- Créez les variables suivantes :
 - `$prenom` : chaîne de caractères
 - `$nom` : chaîne de caractères

- `$age` : nombre entier
- Affichez une phrase complète du type :

`Je m'appelle Marie Dupont et j'ai 28 ans.`

Ce qu'il faut utiliser : variables, concaténation (`.`), `echo` .

2. Conditions

Exercice 2.1 – Majeur ou mineur

Objectif : Utiliser une condition `if` .

- Créez une variable `$age` contenant un nombre.
- Si l'âge est supérieur ou égal à 18, affichez :

`Vous êtes majeur.`

- Sinon, affichez :

`Vous êtes mineur.`

Ce qu'il faut utiliser : `if` , opérateurs de comparaison (`>=`), `echo` .

Exercice 2.2 – Pair ou impair

Objectif : Tester un nombre avec l'opérateur modulo.

- Créez une variable `$nombre` contenant un nombre entier.
- Si ce nombre est divisible par 2, affichez :

`Le nombre est pair.`

- Sinon, affichez :

`Le nombre est impair.`

Ce qu'il faut utiliser : opérateur modulo `%` , `if` , `echo` .

Exercice 2.3 – Appréciation d'une note

Objectif : Utiliser plusieurs conditions `if / elseif / else` .

- Créez une variable `$note` contenant un nombre entre 0 et 20.
- Affichez une appréciation selon la grille suivante :
 - Moins de 10 : `Insuffisant`
 - De 10 à 13 inclus : `Passable`

- De 14 à 16 inclus : `Bien`
- 17 ou plus : `Très bien`

Ce qu'il faut utiliser : `if / elseif / else` , opérateurs logiques.

3. Boucles

Exercice 3.1 – Compter de 1 à 10

Objectif : Utiliser une boucle `for` .

- À l'aide d'une boucle `for` , affichez les nombres de 1 à 10, un par ligne.

Ce qu'il faut utiliser : `for` , `echo` .

Exercice 3.2 – Nombres pairs jusqu'à 20

Objectif : Utiliser une boucle `while` .

- Créez une variable `$i` initialisée à 0.
- Tant que `$i` est inférieur ou égal à 20 :
 - Si `$i` est pair, affichez-le.
 - Incrémentez `$i` de 1 à chaque tour.

Ce qu'il faut utiliser : `while` , `if` , `%` , `echo` .

Exercice 3.3 – Table de multiplication d'un nombre

Objectif : Utiliser une boucle pour générer une table.

- Créez une variable `$n` contenant un nombre entier (ex. : 5).
- Utilisez une boucle `for` pour afficher les 10 premiers résultats de la table de `$n` :
 - Exemple : `5 × 1 = 5` , `5 × 2 = 10` , ..., `5 × 10 = 50`

Ce qu'il faut utiliser : `for` , `*` , `echo` .

4. Débogage avec `var_dump`

Exercice 4.1 – Corriger les erreurs de syntaxe

Objectif : Identifier et corriger les erreurs avec `var_dump` .

Voici un code avec plusieurs erreurs.

Instructions :

- Copiez ce code dans un fichier PHP.
- Utilisez `var_dump()` pour afficher le contenu des variables et identifier les erreurs.
- Corrigez le script.

```
<?php
$nombre1 = 5
$nombre2 = "3";

$somme = $nombre1 + $nombre;

echo "La somme est : $somme
?>
```

Ce qu'il faut utiliser : `var_dump($variable);`, vérifier les noms, les points-virgules, les guillemets fermants.

Exercice 4.2 – Problème de condition

Voici un autre script erroné.

Instructions :

- Comprenez le comportement inattendu.
- Utilisez `var_dump()` pour observer la valeur réelle comparée.
- Corrigez l'erreur de comparaison.

```
<?php
$age = 18;

if($age = 21) {
    echo "Vous avez 21 ans.";
} else {
    echo "Vous n'avez pas 21 ans.";
```

```
}  
?>
```

Ce qu'il faut utiliser : `var_dump($age);`, correction de l'opérateur (`=` vs `==`), test de condition.

5. Exercices d'application (plus complets)

Exercice 5.1 – Générateur de mot de passe simplifié

Objectif : Générer une chaîne aléatoire de 8 caractères alphanumériques.

- Créez une variable `$motDePasse` vide.
- Utilisez une boucle `for` pour y ajouter 8 caractères aléatoires.
- Pour chaque caractère, tirez un nombre entre 0 et 25 (lettres) ou entre 0 et 9 (chiffres), puis convertissez-le en caractère à l'aide de `chr()`.

Ce qu'il faut utiliser : `for`, `rand()`, `chr()`, concaténation (`.=`).

Note : Ne pas utiliser de tableau de caractères. Utilisez directement des conversions numériques vers lettres.

Exercice 5.2 – Calculatrice simple

Objectif : Appliquer une opération arithmétique choisie.

- Créez trois variables :
 - `$a` : un nombre
 - `$b` : un autre nombre
 - `$operation` : un caractère représentant l'opération (`+`, `-`, `*`, `/`)
- Utilisez une structure `if / elseif` pour effectuer l'opération correspondant au symbole.
- Affichez le résultat.

Ce qu'il faut utiliser : `if`, opérateurs arithmétiques, `echo`.
