

# Correction Algo PHP - Partie 1

---

## Exo 1

Soit la phrase « Notre formation DL commence aujourd'hui ». Ecrire un algorithme permettant de compter le nombre de caractères contenus dans cette phrase (espaces inclus).

```
<?php

$phrase = "Notre formation DL commence aujourd'hui";
// strlen() compte le nombre de caractères
$longueur = strlen($phrase);
echo "La phrase « $phrase » contient ".strlen($phrase)." caractères<br/>";
echo "La phrase « $phrase » contient $longueur caractères";

?>
```

## Exo 2

A partir de la phrase de l'exercice 1, écrire l'instruction permettant de compter le nombre de mots contenus dans celle-ci.

```
<?php

$phrase = "Notre formation DL commence aujourd'hui";
// str_word_count() compte le nombre de mots dans une chaîne de caractères
echo "La phrase « $phrase » contient ".str_word_count($phrase)." mots";

?>
```

## Exo 3

A partir de la phrase de l'exercice 1, écrire l'instruction permettant de remplacer le mot « aujourd'hui » par le mot « demain ». Afficher l'ancienne et la nouvelle phrase.

```
<?php

$phrase = "Notre formation DL commence aujourd'hui";
echo $phrase."<br>";
echo str_replace("aujourd'hui", "demain", $phrase);

?>
```

## Exo 4

Ecrire un algorithme permettant de savoir si une phrase est palindrome.

```
<?php

$phrase = "Engage le jeu que je le gagne";
$phraseLowRev = strtolower(strrev(str_replace(" ", "", $phrase)));

/*
FONCTIONS UTILISEES :
- strtolower : convertir une chaîne de caractères en minuscules
- str_replace : remplacer un mot par un autre dans une chaîne de caractères
(ici pour supprimer l'espace)
- strrev : inverse le sens d'une chaîne de caractères
*/

// if classique
if(strtolower(str_replace(" ", "", $phrase)) == $phraseLowRev){
    echo "La phrase « $phrase » est palindrome<br/>";
}else{
    echo "La phrase « $phrase » est non palindrome<br/>";
}

// ternaire
$reponse = (strtolower(str_replace(" ", "", $phrase)) == $phraseLowRev) ?
"palindrome" : "non palindrome";
echo "La phrase « $phrase » est $reponse";
?>
```

## Exo 5

Ecrire un algorithme qui déclare une valeur en francs et qui la convertit en euros. Attention, la valeur générée devra être arrondie à 2 décimales.

```
<?php

$valeurFrancs = 1000000;
// $valeurEuros = round($valeurFrancs / 6.55957, 2);
$valeurEuros = number_format($valeurFrancs / 6.55957, 2, ",", " ");
// number_format permet de paramétrer plus de choses à l'affichage :
séparateur de milliers ainsi que le symbole de la décimale (round() reste correct
cependant)

echo "<h2>Avec number_format()</h2>";
echo "Montant en francs : ". number_format($valeurFrancs, 2, ",", " ") ."
francs<br> " . number_format($valeurFrancs, 2, ",", " ") ." francs = $valeurEuros
€<br/>";

//ou
echo "<h2>Avec round()</h2>";
echo "Montant en francs : $valeurFrancs francs<br>$valeurFrancs francs = ".
round($valeurFrancs / 6.55957, 2). " €";
?>
```

## Exo 6

Ecrire un algorithme permettant de calculer un montant de facture à régler à partir de la quantité d'articles, son prix hors taxe et un taux de TVA (exprimé en décimal. Ex : 20 % -> 0.2)

```
<?php

$prixUnitaire = 9.99;
$qte = 5;
$tauxTVA = 0.2;

// Factorisation du calcul du prix ttc, d'autres méthodes étaient acceptées
$prixTTC = $prixUnitaire * $qte * (1 + $tauxTVA);

echo "PU : $prixUnitaire €<br>
Qté : $qte articles<br>
Taux TVA : " . $tauxTVA * 100 . " %<br>
Le montant de la facture à régler est de $prixTTC €";

?>
```

## Exo 7

Ecrire un algorithme permettant de renvoyer la catégorie d'un enfant en fonction de son âge :

- Poussin : entre 6 et 7 ans
- Pupille : entre 8 et 9 ans<
- Minime : entre 10 et 11 ans
- Cadet : à partir de 12 ans
- Si la catégorie n'est pas gérée, merci de le préciser.

```
<?php

$age = 14;
$cat = "";

switch(true){
    case $age >= 12: $cat = "Cadet"; break;
    case $age >= 10: $cat = "Minime"; break;
    case $age >= 8 : $cat = "Pupille"; break;
    case $age >= 6 : $cat = "Poussin"; break;
    default : $cat = "Non gérée";
}

echo "Age de l'enfant : $age ans<br/>";
echo "L'enfant qui a $age ans appartient à la catégorie : $cat";

?>
```

## Exo 8

Ecrire un algorithme qui renvoie la table de multiplication d'un nombre passé en paramètre

```
<?php

$table = 8;

// Méthode FOR
echo "<h2>Avec FOR</h2>";
for ($i=1; $i <= 10; $i++) {
    echo "$i x $table = ". $i * $table . "<br>";
}

// METHODE WHILE
echo "<h2>Avec WHILE</h2>";
$i = 1;
while ($i <= 10){
    echo "$i x $table = ". $i * $table . "<br>";
    $i++;
}

?>
```

## Exo 9

Nous souhaitons savoir si une personne est imposable en fonction de son âge et de son sexe. Si la personne est une femme dont l'âge est compris entre 18 et 35 ans ou si c'est un homme de plus de 20 ans, alors celle-ci est imposable (sinon ce n'est pas le cas, « non imposable »).

```
<?php

$age = 32;
$sexe = "F";

// on considère ici les conditions comme étant des booléens
$C1 = $age > 20 && $sexe == "M";
$C2 = $sexe == "F" && ($age >= 18 && $age <= 35);

// on teste si au moins l'une des 2 conditions est remplie (simplification du
if de ce fait)
if($C1 || $C2){
    echo "imposable<br/>";
}else{
    echo "non imposable<br/>";
}

// autre méthode avec une écriture ternaire (remplace le if traditionnel et
permet de gérer le tout en 6 lignes !)
$imp = ($C1 || $C2) ? "imposable" : "non imposable";
```

```
echo "La personne est : $imp";  
?>
```

## Exo 10

A partir d'un montant à payer et d'une somme versée pour régler un achat, écrire l'algorithme qui affiche à un utilisateur un rendu de monnaie en nombre de billets de 10 € et 5 €, pièces de 2 € et 1 €.

```
<?php  
  
$aPayer = 152;  
$verse = 200;  
$reste = $verse - $aPayer;  
  
echo "A payer : $aPayer €<br/>";  
echo "Somme versée : $verse €<br/>";  
echo "Reste : $reste €<br/>";  
  
// ***** Solution avec floor *****  
echo "<h2>Solution floor</h2>";  
$nb10 = floor($reste / 10);  
$reste = $reste - 10 * $nb10;  
$nb5 = floor($reste / 5);  
$reste = $reste - 5 * $nb5;  
$nb2 = floor($reste / 2);  
$reste = $reste - 2 * $nb2;  
  
echo "Billets de 10 : $nb10<br/>";  
echo "Billets de 5 : $nb5<br/>";  
echo "Pièces de 2 : $nb2<br/>";  
echo "Pièces de 1 : $reste<br/>";  
  
// ***** Solution avec intdiv *****  
echo "<h2>Solution intdiv</h2>";  
$resteBis = $verse - $aPayer;  
  
$nb10 = intdiv($resteBis, 10);  
$resteBis = $resteBis - 10 * $nb10;  
$nb5 = intdiv($resteBis, 5);  
$resteBis = $resteBis - 5 * $nb5;  
$nb2 = intdiv($resteBis, 2);  
$resteBis = $resteBis - 2 * $nb2;  
  
echo "Billets de 10 : $nb10<br/>";  
echo "Billets de 5 : $nb5<br/>";  
echo "Pièces de 2 : $nb2<br/>";  
echo "Pièces de 1 : $resteBis<br/>";  
?>
```

## Exo 11

Initialiser un tableau de x marques de voitures. Ecrire un algorithme permettant de parcourir ce tableau et d'en afficher le contenu (une marque de voiture par ligne). Il est également demandé d'afficher le nombre de marques de voitures présentes dans le tableau.

```
<?php

$voitures = ["Peugeot","Mercedes","Renault","BMW"];
$nbVoitures = count($voitures);

var_dump($voitures); // pensez à faire des var_dump de temps en temps pour
tester vos variables et leur contenu !

echo "Il y a $nbVoitures voitures dans le tableau<br>";

// Méthode FOR
echo "<ul>";
for($i = 0; $i < $nbVoitures; $i++) {
    echo "<li>".$voitures[$i]."</li>";
}
echo "</ul>";

echo "Il y a ".count($voitures)." voitures dans le tableau<br>";
// Méthode foreach (à privilégier)
echo "<ul>";
foreach ($voitures as $marque){ //$marque = 1 élément du tableau
    echo "<li>$marque</li>";
}
echo "</ul>";
?>
```

## Exo 12

A partir d'une fonction personnalisée et à partir d'un tableau de prénoms et de langue associée (tableau contenant clé et valeur), dire bonjour aux différentes personnes dans leur langue respective (français => « Salut », anglais => « Hello », espagnol => « ... »)

```
<?php

$personnes = ["Micka"=> "FRA", "Virgile"=>"ESP", "MC"=>"ENG"];
$salutations = ["FRA" => "Salut", "ESP" => "Hola", "ENG" => "Hello"];

var_dump($personnes);
var_dump($salutations);

ksort($personnes); // tri de A à Z sur la clé (key, ici le prénom)
// krsort($personnes); // tri de Z à A sur la clé (key, ici le prénom)
// asort($personnes); // tri de A à Z sur la valeur (ici la langue)
// arsort($personnes); // tri de Z à A sur la valeur (ici la langue)
```

```

echo "<strong>Méthode switch..case</strong><br>";
foreach($personnes as $prenom => $langue){
    //echo "$prenom parle $langue<br>";
    switch($langue){
        case "FRA": echo "Salut $prenom<br>"; break;
        case "ENG": echo "Hello $prenom<br>"; break;
        case "ESP": echo "Hola $prenom<br>"; break;
        default: echo "Langue non gérée pour $prenom<br>";
    }
}

echo "<strong>Méthode 2 tableaux associatifs</strong><br>";
foreach ($personnes as $prenom => $langue) {
    if(in_array($langue, array_keys($salutations))){
        $bonjour = $salutations[$langue];
        echo "$bonjour $prenom<br>";
    }else{
        echo "Langue non gérée pour $prenom<br>";
    }
}

?>

```

## Exo 13

Calculer la moyenne générale d'un élève dont les notes sont renseignées dans un tableau (pas de coefficient). Elle devra être affichée avec 2 décimales.

```

<?php

$notes = [10, 12, 8, 19, 3, 16, 11, 13, 9];
$nbNotes = count($notes);

var_dump($notes);

// arrondir à 2 décimales la somme des notes divisée par le nombre de notes
$moyenne = round(array_sum($notes) / $nbNotes, 2);
// ou number_format

// implode affiche le contenu du tableau avec le séparateur spécifié en 1er
argument snas devoir nécessairement passer par une boucle
echo "Les notes obtenues sont : " . implode(", ", $notes);
echo "<br>La moyenne est de : $moyenne";

// var_dump(array_values($notes));

?>

```

## Exo 14

Calculer l'âge exact d'une personne à partir de sa date de naissance (en années, mois, jours).

```
<?php

// $date1 et $date2 sont des instantiations de la classe DateTime (objets)
$date1 = new DateTime('1985-01-17');
$date2 = new DateTime(); //date du jour
$interval = $date1->diff($date2);
echo "La personne née le ".$date1->format('d-m-Y'). " a ".$interval-
>format('%y ans %m mois et %d jours');
?>
```

## Exo 15

Créer une classe **Personne** (nom, prenom, date de naissance), instancier 2 personnes et renvoyer leur infos :  
nom + prénom + âge

```
<?php

$p1 = new Personne("DUPONT", "Michel", "1980-02-19");
$p2 = new Personne("VALET", "Marion", "1985-01-17");

var_dump($p1);
var_dump($p2);

//Afficher les infos des personnes instanciées
echo $p1."<br>";
echo $p2."<br>";

$p1->setPrenom("Michèle");
echo $p1."<br>";

/**
 * Class Personne
 */
class Personne {
    //Attributs (ou variables d'instance), par convention en "private",
    "protected" éventuellement si on fait appel à de l'héritage de classe
    private $nom;
    private $prenom;
    private $dateNaissance;

    //Constructeur
    public function __construct($nom,$prenom,$dateNaissance){
        $this->nom = $nom;
        $this->prenom = $prenom;
        $this->dateNaissance = new DateTime($dateNaissance);
    }

    //Getters et setters (accesseurs et mutateurs)
    public function getNom(){
        return $this->nom;
    }
}
```



```
}

public function setNom($nom){
    echo $this->nom." ".$this->prenom." a changé de nom !<br/>";
    $this->nom=$nom;
}

public function getPrenom(){
    return $this->prenom;
}

public function setPrenom($prenom){
    echo $this->nom." ".$this->prenom." a changé de prénom !<br/>";
    $this->prenom=$prenom;
}

public function getDateNaissance(){
    return $this->dateNaissance;
}

public function setDateNaissance($dateNaissance){
    echo $this->nom." ".$this->prenom." a changé de date de naissance
(???) !<br/>";
    $this->dateNaissance=$dateNaissance;
}

//Méthode perso pour le calcul de l'âge
public function getAge(){
    $now = new DateTime(); // date du jour
    $interval = $this->dateNaissance->diff($now);
    $age = $interval->format('%y'); // récupérer le nombre d'années
    return $age;
}

public function __toString(){
    return $this->prenom." ".$this->nom." a ".$this->getAge()." ans";
}
}

?>
```