

# Structures de langage et fonctions prédéfinies en PHP

Questionnaire sur les Structures de langage, les fonctions prédéfinies (ou méthodes/procédures) vues jusqu'à présent en PHP

Points: 32/100

1. Cette fonction affiche de nombreuses informations sur PHP, concernant sa configuration courante : options de compilation, extensions, version, informations sur le serveur, environnement PHP, informations sur le système, chemins, valeurs générales et locales de configuration, en-têtes HTTP et la licence PHP. \*

(4 points)

- ☐ infopHP()
- ☐ var\_dump()
- ☒ phpinfo()

2. Pour créer des commentaires sur une ligne, vous pouvez utiliser \*

(4 points)

choix multiple

- ☐ <!-- -->
- ☒ //
- ☒ #
- ☐ ;

3. Lors de l'affichage d'une chaîne de caractères avec un echo ou un print, quelle est la différence entre du texte entre les guillemets doubles "" et du texte entre guillemets simples ' ' \*

(4 points)

Par exemple que va afficher : \$a = 33; \$b = "maison"; echo "ma \$b a été construite ".il y a \$a ans';

- ☐ ma \$b a été construite il y a \$a ans
- ☐ ma maison a été construite il y a 33 ans
- ☒ ma maison a été construite il y a \$a ans
- ☐ ma \$b a été construite il y a 33 ans

4. Quels sont les opérateurs de concaténation en PHP \*

(4 points)

Cochez aussi les opérateurs ne fonctionnant pas dans tous les cas

- ☒ .
- ☐ +
- ☐ -
- ☐ \$
- ☒ ,
- ☐ |

5. Utilisez l'écriture courte de PHP pour remplacer le code ci-dessous en gardant l'html \*

(5 points)

<p><?php echo "Hello World!" ; ?> </p>

<p><?="Hello World!";?> </p>

6. Quel caractère permet d'échapper les guillemets simples ou doubles ( ' ou " ) dans une variable de type string ? \*

(4 points)

- ☐ !
- ☐ :
- ☐ /
- ☒ \
- ☐ \*

7. Quels sont les types de données qui peuvent être contenus dans une variable ? \*

(10 points)

boolean, integer, float, string, array, NULL, object

8. Quel raccourci permet de transformer le contenu d'une variable (et donc son type) en chaîne de caractères? \*

(4 points)

\$mavar = ... 5.27;

- ☐ (char)
- ☐ (varchar)
- ☐ (str)
- ☒ (string)
- ☐ (str\_replace)
- ☐ (text)

9. `var_dump()` est utile pour \*  
(4 points)

- ☐ Afficher du texte
- ☐ pour déboguer une variable avec un affichage simple
- ☐ pour déboguer une variable avec un affichage complet
- ☒ pour déboguer une variable ou plusieurs variables avec un affichage complet
- ☐ pour déboguer une variable ou plusieurs variables avec un affichage simple

10. Créez une variable nommée `$myArray` qui contient un tableau indexé avec 4 valeurs sur 1 niveau \*  
(5 points)

```
$myArray = [14,18,52,46];
```

11. Créez une variable nommée `$myAssocArray` qui contient 3 clefs nommées 'un', 'deux', 'trois' \*  
(8 points)

La valeur de 'un' doit être un booléen, 'deux' un float, 'trois' un autre tableau indexé ayant comme valeurs 2 entrées : "coucou" et 25

```
$myAssocArray = ["un"=>true,  
                "deux"=>42.42,  
                "trois"=>["coucou",  
                          25  
                        ]  
                ];
```

12. En lien avec la question 11, qu'utiliseriez-vous pour afficher la variable `$myAssocArray` dans du html en mode débogage simplifié? \*  
(4 points)

```
<pre> ... </pre>
```

- ☐ echo
- ☐ var\_dump
- ☐ print\_f
- ☒ print\_r
- ☐ print

13. Ecrivez la suite du code pour afficher l'état du calcium à pression normale, en sachant que le calcium est solide en dessous de 842°: \*

(15 points)

```
$tempHasard = mt_rand(-100, 2500);  
$CaLiquide = 842;  
$CaGazeux = 1484;
```

/\* 3 possibilités: (avec la température prise au hasard affichée, donc interprétée)

- Le Calcium est solide à \$tempHasard °
- Le Calcium est liquide à \$tempHasard °
- Le Calcium est gazeux à \$tempHasard °

\*/

```
$tempHasard = mt_rand(-100, 2500);  
$CaLiquide = 842;  
$CaGazeux = 1484;  
  
echo $tempHasard."°<hr>";  
  
if($tempHasard < $CaLiquide){  
    echo "- Le Calcium est solide à ".$tempHasard . "°";  
}elseif($tempHasard < $CaGazeux){  
    echo "- Le Calcium est liquide à ".$tempHasard . "°";  
}else{  
    echo "- Le Calcium est gazeux à ".$tempHasard . "°";  
}
```

14. Faites un résumé des 4 types de boucles vues au cours \*

(25 points)

Il faudra leurs noms, mais également un exemple pratique pour chacune d'entre elles. Ne faites pas un simple copier/coller, montrez que vous comprenez le fonctionnement, et inutile de le prendre de quelqu'un d'autre, je le verrai! Bon boulot !

```
<h3>while()</h3>
<p>Boucle utilisée quand on ne connaît pas le nombre à l'avance de tours à effectuer</p>
<?php
$userInput=false;
$seconde=0;
while($userInput===false){
    echo "Vous devez entrer qqchose<br>";
    $seconde++;
    if($seconde >=5){
        $userInput = true;
    }
}

<h3>do while()</h3>
<p>Boucle utilisée quand on ne connaît pas le nombre à l'avance de tours à effectuer mais
qu'on veut exécuter le code de l'itération au moins une fois</p>
<?php
$nbpg = mt_rand(-5,10);
$pg = 1;

echo '$pg = '.$pg.' | $nbpg = '.$nbpg.'<br>';
do{
    if($pg===1) echo "page ";
    if($nbpg<1) $nbpg = 1;
    echo "$pg -";
    $pg++;
}while($pg<=$nbpg);

?>

<h3>for()</h3>
<p>Boucle quand on connaît le nombre d'itération ou on peut la calculer (mathématique)
</p>
<?php
$array = [1,2,3];
$nbArray = count($array);
for ($i=0;$i<$nbArray;$i++){
    echo "$i => $array[$i]<br>";
}
?>

<h3>foreach()</h3>
<p>Boucle qui permet de lister des tableaux ou des objets, il part du premier élément et
s'arrête au dernier élément</p>
<?php
$array = [1,2,3];
// juste avec la valeur
foreach($array as $valeur){
    echo '$valeur = '.$valeur."<br>";
}
```

```
// clef et valeur
foreach($array as $clef => $valeur){
    echo '$clef=>$valeur = '.$clef.">".$valeur."<br>";
}
```

Ce contenu est créé par le propriétaire du formulaire. Les données que vous soumettez sont envoyées au propriétaire du formulaire. Microsoft n'est pas responsable des pratiques de confidentialité ou de sécurité de ses clients, y compris celles de ce propriétaire de formulaire. Ne donnez jamais votre mot de passe.

Avec Microsoft Forms | [Confidentialité et cookies](#) | [Conditions d'utilisation](#) | [Accessibilité](#)