

QCM PHP et PHP POO

1- Expliquez la différence entre les sites Web statiques et dynamiques ?

Réponse : Dans les sites Web statiques, le contenu ne peut pas être modifié après l'exécution du script. Vous ne pouvez rien changer sur le site. C'est prédéfini.

Dans les sites Web dynamiques, le contenu du script peut être modifié au moment de l'exécution. Son contenu est régénéré à chaque visite ou rechargement d'un utilisateur.

2- Quel est le nom du moteur de script qui interprète le PHP ?

Réponse : Le moteur de script qui interprète le PHP, s'appelle Zend Engine 2.

3- Quelle est la différence entre GET et POST ?

Réponse :

- GET affiche les données soumises dans l'URL, alors dans la méthode POST, les informations ne sont pas affichées car elles sont encodées dans la requête.
- GET peut gérer un maximum de 2048 caractères, POST n'a pas de telles restrictions.
- GET n'autorise que les données ASCII, POST n'a aucune restriction, les données binaires sont également autorisées.
- Normalement, GET est utilisé pour récupérer des données tandis que POST pour insérer et mettre à jour des données.

4- Expliquez la différence entre PHP4 et PHP5 ?

Réponse :

- PHP4 ne prend pas en charge le concept [POO \(Programmation Orientée Objet\)](#) et utilise Zend Engine 1.
- PHP5 prend en charge le concept POO et utilise Zend Engine 2.

5- À quel langage de programmation PHP ressemble-t-il ?

Réponse : La syntaxe PHP ressemble à Perl et C

6- Quelle est la version PHP actuellement utilisée ?

Réponse : La version 7.1 ou 7.2 est la version recommandée de PHP.

7- Comment afficher la sortie directement dans le navigateur ?

Réponse :

```
<?= $a= 12; ?>
```

```
<?php echo $a=12 ; ?>
```

8- L'héritage multiple est-il pris en charge en PHP ?

Réponse : PHP ne prend en charge que [l'héritage simple](#); cela signifie qu'une classe peut être étendue à partir d'une seule classe en utilisant le mot-clé 'extends'.

9- Quelle est la différence entre \$message et \$\$message ?

Réponse : \$message est une variable simple alors que \$\$message est une variable de variable.

Exemple :

```
<?php
```

```
$Welcome = "Hello World!";
```

```
$message = "Welcome";
```

```
echo $message;
```

```
echo $$message;
```

```
?>
```

Sortie:

Welcome

WayToLearnX

10- Qu'est-ce qu'un cookie persistant ?

Réponse : Un cookie persistant est un cookie qui est stocké dans un fichier en permanence sur votre ordinateur. Par défaut, les cookies sont créés en tant que cookies temporaires stockés uniquement dans la mémoire du navigateur. Lorsque le navigateur est fermé, les cookies temporaires seront effacés. Vous devez décider quand utiliser les cookies temporaires et quand utiliser les cookies persistants en fonction de leurs différences :

- Les cookies temporaires ne peuvent pas être utilisés pour le suivi des informations à long terme.
- Les cookies persistants peuvent être utilisés pour le suivi des informations à long terme.
- Les cookies temporaires sont plus sûrs car aucun programme autre que le navigateur ne peut y accéder.
- Les cookies persistants sont moins sécurisés car les utilisateurs peuvent ouvrir les fichiers cookie en consultant les valeurs des cookies.

11- Comment définissez-vous une constante dans PHP ?

Réponse : Avec la directive define(), exemple:

```
define("MYCONSTANT", 100);
```

12- Quelles sont les différences entre Require, Include et Include_once?

Réponse : « require_once() » et « include_once() » permettent d'inclure et d'évaluer un fichier qu'une seule fois. Tandis que « require() » et « include() » le feront autant de fois qu'on leur demande de le faire.

13- Quelle est la différence entre « mysql_fetch_object » et « mysql_fetch_array »?

Réponse : [mysql_fetch_object](#) est similaire à [mysql_fetch_array](#), avec une seule différence est qu'il retourne un objet au lieu d'un tableau.

14- Comment effectuer le Passage par référence en PHP ?

Réponse : Les références sont un moyen d'avoir plusieurs variables référençant le même contenu d'une variable avec des noms différents comme un alias. Ainsi, quel que soit le nom que vous utilisez, une opération sur cette variable aura toujours un effet sur les autres.

Exemple :

```
<?php
$x = 10;
$y = &$x;
$y = 15;
var_dump($x);
var_dump($y);
?>
```

Maintenant, dans la ligne 4, \$x et \$y auront la même valeur 15.

15. Quels sont les systèmes de gestion de contenu (CMS) populaires en PHP ?

Réponse :

- WordPress
- Joomla
- Magento
- Drupal

16. Quels sont les frameworks populaires en PHP ?

Réponse :

Symfony, Laravel, CakePHP, CodeIgniter, Yii 2, Zend Framework.

17. Qu'est-ce que « echo » en PHP ?

Réponse : echo génère une ou plusieurs chaînes. C'est une construction de langage et non une fonction. L'utilisation de parenthèses n'est donc pas obligatoire. Mais si vous souhaitez transmettre plus d'un paramètre à echo, l'utilisation de parenthèses est requise.

18. Qu'est-ce que « print » en PHP ?

Réponse : **print** affiche une chaîne de caractères. C'est une construction de langage et non une fonction. Ainsi, l'utilisation de parenthèses n'est pas requise en cas d'une liste d'arguments. Contrairement à echo, il renvoie toujours 1.

19- Quelle est la différence entre « echo » et « print » en PHP ?

Réponse :

echo peut afficher une ou plusieurs chaînes mais print ne peut afficher qu'une seule chaîne et renvoie toujours 1.

echo est plus rapide que print car il ne renvoie aucune valeur.

Exemple:

```
<?php
echo "Welcom to PHP World";

$n = print "Welcom to PHP World";

?>
```

20- Comment une variable est déclarée en PHP ?

Réponse : Une variable est un nom d'un emplacement mémoire qui contient des données. C'est un stockage temporaire.

```
$myVar = value;
```

21- Comment on peut définir une constante en PHP ?

Réponse : Une constante 'PHP' est une variable qui ne peut pas être modifiée pendant l'exécution du script. Une constante 'PHP' est définie de deux manières :

Utilisation de la fonction define() ; Utilisation de la fonction « const() »

22- Combien de types de données existe-t-il en PHP ?

Réponse : Les types de données PHP sont utilisés pour contenir différents types de données ou de valeurs. Il existe 7 types de données primitifs qui sont classés en 3 types :

Types scalaires : (boolean, integer, float, string, etc.)

Types composés : (array et object)

Types spéciaux : (NULL)

23- A quoi sert la fonction isset() ?

Réponse : La fonction isset() vérifie si la variable est définie et non null.

24- Comment savoir si un nombre est pair ou impair sans utiliser ni des conditions ni des boucles ?

Réponse :

```
<?php
$tab = array("0"=>"Pair", "1"=>"Impair");
$n = 4;
echo "Le nombre est: ".$tab[$n%2];
```

25- Combien de types de tableaux existe-t-il en PHP ?

Réponse : Il existe trois types de tableau en PHP :

Tableau indexé : un tableau avec une clé numérique.

Tableau associatif : un tableau où chaque clé a sa valeur spécifique.

Tableau multidimensionnel : un tableau contenant un ou plusieurs tableaux.

26- Citez quelques fonctions du tableau(array) en PHP ?

Réponse : Il existe de nombreuses fonctions de tableau en PHP :

```
array(); in_array(); next(); count(); sort(); array_reverse(); array_search(); sizeof();
shuffle(); range(); arraydiff(); array_values(); array_key_exists(); key(); array_keys();
list(); extract(); compact(); array_merge(); array_fill(); array_sum(); array_replace();
array_replace_recursive(); array_slice(); array_rand(); array_pop(); array_push();
array_shift(); array_chunk(); array_intersect();
```

27 - Quelle est la différence entre un tableau indexé et un tableau associatif ?

Réponse :

Un tableau indexé contient des éléments sous une forme indexée qui est représentée par un nombre qui commence à partir de 0 et incrémenté de 1. Par exemple : \$couleur = array("bleu","rouge","vert","orange");

Un tableau associatif contient des éléments associés à un nom. Par exemple :

```
$personne = array("alex" => "25", "thomas" => "30", "emily" => "18");
```

28- Quelle est la différence entre la boucle for et foreach en PHP ?

Réponse : La boucle « for » est principalement utilisée pour itérer un nombre prédéfini de fois et la boucle « foreach » est utilisée pour lire les éléments du tableau ou l'ensemble de résultats MySQL où le nombre d'itérations est inconnu.

29 -Quelle est la différence entre unset() et unlink()

Réponse : unset() détruit une variable, tandis que la fonction unlink() supprime un fichier que nous lui transmettons du système de fichiers.

Exemple : unset() `<?php $var = 'WayToLearnX'; unset($var); ?>`

Exemple : unlink() `<?php $file = "myfile.txt"; unlink($file); ?>`

30- Quelle est la sortie du code suivant :

`$a = '1'; $b = &$a; $b = "2$b"; echo $a.", ".$b;` Réponse : 21, 21

31- Citez quelques fonctions sur les chaînes de caractères en PHP ?

Réponse : Il existe de nombreuses fonctions sur les chaînes de caractères en PHP:

strlen()

strtolower()

strtoupper()

lcfirst()

ucfirst()

ucwords()

strrev()

32- Quelles sont les méthodes pour soumettre un formulaire en PHP ?

Réponse : Il existe deux méthodes GET et POST.

33- Qu'est-ce qu'une classe finale et méthode finale en PHP ?

Réponse : Le mot-clé 'final' est introduit en PHP5. Une classe finale signifie que cette classe ne peut pas être héritée et qu'une méthode finale ne peut pas être surchargée.

34- Comment vous pouvez inclure un fichier en PHP ?

Réponse : PHP vous permet d'inclure un fichier afin que le contenu d'une page puisse être réutilisé à nouveau. Il existe deux façons d'ajouter un fichier en PHP.

Include et require

33- Comment les échecs d'exécution sont-ils traités avec les fonctions include() et require() ?

Réponse : Si la fonction require() ne peut pas accéder au fichier, elle se termine par une erreur fatale. Pourtant, la fonction include() donne un warning et le script PHP continue de s'exécuter.

34- Quand utiliser explode() et implode() en PHP?

Réponse : La fonction explode() est utilisée pour diviser une chaîne en un tableau et la fonction implode() est utilisée pour créer une chaîne en combinant les éléments du tableau.

Exemple :

```
<?php
$str = "Welcome to WayToLearnX";
print_r (explode(" ", $str));

$lang = array('PHP', 'Java', 'Python');
echo implode(" ", $lang);

?>
```

Sortie:

```
Array (
    [0] => Welcome
    [1] => to
    [2] => WayToLearnX
)
PHP Java Python
```

35- A quoi sert la fonction PHP explode()?

Réponse : La fonction PHP explode() découpe une chaîne en un tableau suivant un délimiteur.

Exemple:

```
<?php
$result = explode(" ", "Welcome to WayToLearnX");

// Afficher le résultat
print_r($result);

?>
```

Sortie:

```
Array
```

```
(  
    [0] => Welcome  
    [1] => to  
    [2] => WayToLearnX  
)
```

37- Supposant on a la chaîne suivante \$str = "1,2,3,4,5,6,7,8,9";. Comment obtiendriez-vous la somme des entiers contenus dans cette chaîne?

Réponse : `<?php echo array_sum(explode(',',$str));`

38- A quoi sert la fonction PHP preg_split()?

Réponse : La fonction PHP preg_split() divise une chaîne en un tableau selon une expression régulière.

Exemple:

```
<?php  
  
$result = preg_split("/[\\s,]+/", "Welcome to WayToLearnX");  
  
// Afficher le résultat  
  
print_r($result);  
  
?>
```

Sortie:

Array

```
(  
    [0] => Welcome  
    [1] => to  
    [2] => WayToLearnX  
)
```

39- Quels sont les 3 niveaux de visibilité disponibles en PHP et comment les définiriez-vous?

Réponse :

1. Private – Visible uniquement dans sa propre classe
2. Public – Visible partout.
3. Protected – Visible uniquement par les classes parent(s) et les classes qui héritent la classe actuelle

40- Comment renvoyer une valeur à partir d'une fonction ?

Réponse : Une fonction renvoie une valeur en utilisant l'instruction `return $val;`

41- Quelle est la différence entre une classe abstraite et une interface ?

Réponse : Dans les classes abstraites, il n'est pas nécessaire que chaque méthode soit abstraite. Mais dans l'interface, chaque méthode est abstraite.

Une méthode d'interface ne doit être que public. La méthode dans une classe abstraite en PHP peut être private, public ou protected.

Une interface peut hériter d'autres interfaces. Mais une classe abstraite ne peut hériter qu'une classe abstraite.

La classe fille de la classe mère abstraite n'a pas besoin d'implémenter toutes les méthodes dans la classe mère. Mais avec l'interface, la classe fille doit implémenter toutes les méthodes dans l'interface.

Dans une classe abstraite, vous pouvez définir ainsi que déclarer des méthodes. Mais dans l'interface, vous ne pouvez pas définir vos méthodes.

42- Qu'est-ce que « PDO » ?

Réponse : La forme complète de **PDO = PHP Data Objects**. Il s'agit d'une extension PHP légère qui utilise une interface pour accéder à la base de données. Grâce à PDO, un développeur peut facilement passer d'un serveur de base de données à l'autre. Mais il ne prend pas en charge toutes les fonctionnalités avancées.

43- Le constructeur de la classe mère est-il appelé implicitement dans un constructeur d'une classe fille?

Réponse : Non, le constructeur de la classe mère doit être appelé explicitement comme suit : **`parent::__construct($value)`**

Exemple:

```
<?php
class A{
    public function __construct(){
        echo "Je suis la classe mère.<br>";
    }
}
class B extends A{
    public function __construct(){
```

```
parent::__construct();  
echo "Je suis la classe fille."  
}  
}
```

```
$b = new B();  
?>
```

Sortie :

Je suis la classe mère.

Je suis la classe fille.

Session & cookies :

44- Quelle est la méthode pour enregistrer une variable dans une session ?

Réponse : <?php session_register(\$session_var);

45- Comment récupérer une valeur de cookie ?

Réponse : Vous pouvez lire une valeur de cookie en utilisant : \$_COOKIE['NOM_DE_COOKIE'];

Exemple : echo \$_COOKIE ["user"];

46- Dernière Question : A quoi sert la fonction setcookie() en PHP ?

Réponse : La fonction PHP setcookie() est utilisée pour définir un cookie avec une réponse HTTP. Une fois le cookie défini, vous pouvez y accéder par la variable superglobale \$_COOKIE.

setcookie(name, value, expire, path, domain, secure, httponly);