**PHP**

**1.** Laquelle des versions PHP suivantes peut exécuter ce script?

*// script.php*

$data = [1, 2, 3];

**echo** $data[0];

1. 5.3+

2. 5.4+

3. 5.5+

4. 5.6+

Réponse : 2,3,4

**2.** Dans la liste suivante, il y a deux fonctionnalités qui n'ont jamais été disponibles en PHP (jusqu'à 5.6).

1. Variadic functions

2. Generators

3. Named parameters

4. Generics

Réponse : 3,4

**3.** En PHP, quelle est la valeur de MyClass :: class\_name?  
1. Depuis PHP 5.5, :: class\_name contient le nom complet de la classe

2. Depuis PHP 5.5, :: class\_name contient le nom de la classe (sans l'espace de noms)

3. Depuis PHP 5.5, :: class\_name contient une référence à la classe elle-même

4. À moins qu'il ne soit défini, :: class\_name n'existera pas

Réponse : 4

**4.** Les interfaces peuvent-elles définir des constantes en PHP?

1. Yes

2. No

Réponse : 1

**5.** En PHP, les classes / interfaces s'étendant à partir d'autres classes / interfaces peuvent-elles remplacer les constantes? du parent?  
1. Oui

2. Seules les classes peuvent remplacer les constantes

3. Seules les interfaces peuvent remplacer les constantes

4. Non, les constantes ne peuvent jamais être annulées

Réponse : 2

**6.** Quels sont les effets de déclarer une classe comme finale en PHP?  
1. Les classes finales ne peuvent pas être étendues

2. Les classes finales peuvent être étendues mais les classes enfants ne pourront pas remplacer les méthodes

3. Les classes finales ne peuvent pas être instanciées

4. Le mot-clé final n'existe pas

Réponse : 1

**7.** L'héritage multiple est-il supporté en PHP?

1. Oui, il est pris en charge dans les classes et les interfaces

2. Il est uniquement pris en charge dans les classes

3. Il est uniquement pris en charge dans les interfaces

4. Non

Réponse : 3

**8.** Quel est le problème avec le code PHP suivant?

*// example.php*

**use** Symfony\Component\HttpFoundation\Request;

**final class MyRequest extends** Request

{

**protected function** test() {}

}

$request = **new** MyRequest();

1. Les classes finales ne peuvent pas être instanciées

2. Les classes finales ne peuvent pas définir de méthodes protégées

3. Les classes finales ne peuvent pas s'étendre d'une autre classe

4. Rien, le code est correct

Réponse : 4

**9.** Comment pouvez-vous vérifier la syntaxe d'un script PHP sans l'exécuter?

1. En utilisant la fonction php\_check\_syntax ()

2. En utilisant la fonction eval ()

3. Utiliser le binaire PHP avec l'option -l

4. Utilisation du serveur web intégré en PHP 5.4+

Réponse : 3

**10.** Y at-il un problème avec le code PHP suivant?

*// script.php*

**class Book** {

**var** $title = 'Title';

}

$book = **new** Book();

**echo** $book->title;

1. Oui, les propriétés doivent être déclarées publiques, protégées ou privées, et non var

2. Oui, les propriétés déclarées avec var sont comme private, donc $ book-> title renvoie une erreur fatale

3. Oui, le titre de propriété n'est pas initialisé

4. Non

Réponse : 4

**11.** Lesquelles des phrases suivantes sont vraies à propos des classes abstraites en PHP?

1. Ils ne peuvent pas être instanciés

2. Ils prennent en charge plusieurs interfaces similaires à l'héritage

3. Ils peuvent être déclarés définitifs

4. Ils peuvent inclure des méthodes abstraites

Réponse : 1,4

**12.** La définition de classe suivante renvoie-t-elle une erreur? Si oui, quelle erreur ?

*// abstract\_method.php*

**class Book** {

**abstract function** getTitle();

}

1. Oui

2. Non

Réponse : 1 ( une erreur fatale )

**13.** Y a-t-il une erreur dans le code suivant?

*// script.php*

**namespace** Entity {

**class Book** {}

}

**namespace** Service {

**class Book** {}

}

**namespace** {

$book1 = **new** Entity\Book();

$book2 = **new** Service\Book();

}

1. Oui, la déclaration de l'espace de nom doit être la toute première déclaration du script

2. Oui, un seul espace de noms peut être déclaré dans un fichier

3. Oui, la classe Service \ Book est introuvable

4. Non

Réponse : 4

**14.** Quelle est la sortie du code suivant?

*// script.php*

**namespace** Bundle\Entity;

**echo** \_\_NAMESPACE\_\_;

1. Entity

2. Bundle

3. Bundle\Entity

4. Fatal error, \_\_NAMESPACE\_\_ is not defined

Réponse : 3

**15.** Quelle est la sortie de : **echo print("hello");?**

1. hello

2. hello0

3. hello1

4. hello5

Réponse : 3

**16.** Quelle est la valeur de \_\_NAMESPACE\_\_ lorsqu'il est utilisé dans un code global?

1. /

2. \

3. Chaîne vide

4. Il n'est pas défini et ne peut pas être utilisé

Réponse : 3

**Architecture Symfony**

**1.** Lesquelles des classes suivantes n'appartiennent pas au composant HttpFoundation?

1. Request

2. Response

3. Firewall

4. Session

Réponse : 3

**2.** Lesquels des fichiers suivants sont des contrôleurs frontaux?

1. app/console

2. web/app.php

3. web/app\_dev.php

4. Symfony\Bundle\FrameworkBundle\Controller

Réponse : 1.2.3

**3.** Lesquels des répertoires suivants ne se trouvent pas dans le répertoire de l'application?

1. cache

2. web

3. config

4. vendor

Réponse : 2.4

**4.** Symfony 3 utilisera une structure de répertoire légèrement différente. Lequel des fichiers suivants ou les répertoires existeront?

1. /bin/console

2. /var/cache

3. /var/logs

4. /app/config

Réponse : 1.2.3.4

**5.** Quel est le modèle de conception qui implémente le composant EventDispatcher?

1. Strategy

2. Factory

3. Adapter

4. Mediator

Réponse :4

**Standardisation Symfony**

**1.** Quel fichier doit contenir des options de configuration liées à l'infrastructure, telles que la base de données

paramètres de connexion?

1. app/config/parameters.yml

2. app/config/config.yml

3. app/config/config\_dev.yml

4. app/config/routing.yml

Réponse : 1

**2.** Que se passe-t-il si vous ajoutez un nouveau paramètre au fichier app / config / parameters.yml.dist et puis exécutez la mise à jour avec Composer ?

1. Il est copié dans app / config / parameters.yml

2. Composer demande la valeur du nouveau paramètre, puis l'ajoute à l'application / config / parameters. fichier yml.

3. Composer refuse de mettre à jour les dépendances jusqu'à app / config / parameters.yml.dist et app / config / parameters.yml ont les mêmes options (peu importe si les valeurs sont différentes).

4. Rien.

Réponse :2

**3.** Qu'est-ce que PSR-2?

1. Une interface commune d'enregistreur

2. Un moyen standard de convertir les noms complets en chemins de fichiers

3. Un utilitaire pour convertir les classes PHP non-namespaced en classes

4. Un guide de style de codage

Réponse :4

**4.** Quel est le problème avec le contrôleur Symfony suivant?

*// src/BookBundle/Controller/BookController.php*

**use** Symfony\Component\HttpFoundation\Response;

**use** Symfony\Bundle\FrameworkBundle\Controller\Controller;

**class BookController** {

**function** ImageAction() {

$return\_response = **new** Response('hello');

**return** $return\_response;

}

}

1. Cela ne fonctionnera pas en raison de problèmes de syntaxe

2. Cela ne fonctionnera pas en raison de problèmes de chargement automatique

3. Cela fonctionnera, mais a besoin de quelques corrections de style de codage

4. Il n'y a rien de mal

Réponse : 3