

Techniques et langages de l'Internet Avancés
1ère session 2017–2018
4ETI – CPE

ETI– DS TLI Avancés 2017–2018 1S

1ère partie (QCM) : 30 minutes, aucun document n'est autorisé.

2ème partie : 1h30, tout document autorisé

Durée : 2 heures

Le barème est sur 100 points.

1 point du barème vaut 0,2 points sur 20

1 point sur 20 correspond à 5 points du barème.

Il est obligatoire de remplir complètement une case de réponse (une simple croix ne suffit pas, il faut noircir la case complètement au stylo (pas au crayon à papier)).

1 Première partie : QCM, 30 points

Aucun document n'est autorisé.

ATTENTION :

- **Vous devez reporter vos réponses sur la feuille volante donnée en fin de sujet.**
- contrairement aux QCMs faits en TP, **une mauvaise réponse est pénalisée.**

Question 1 CGI a défini des variables pour le web dynamique dont la plupart sont encore utilisées aujourd'hui

vrai

faux

Question 2 ♣ PHP est un langage :

A compilé côté serveur

B compilé côté client

C interprété côté client

D interprété côté serveur

E Aucune de ces réponses n'est correcte.

Question 3 ♣ Quelle(s) technique(s) permet(tent) de réaliser un aiguillage avec un seul point d'entrée sur un site web écrit en PHP ?

A les techniques de routage

B l'utilisation de la variable GET pour spécifier le contexte

C l'utilisation de la variable POST pour spécifier le contexte

D Aucune de ces réponses n'est correcte.

CORRECTION

Question 4 Quel est l'objectif premier d'un moteur de template ?

- Faciliter la séparation du contenu et de la structure d'un document HTML
- Simplifier l'écriture et la réutilisation du code PHP

Question 5 PDO permet d'être *complètement* indépendant du SGBD utilisé : par exemple, n'importe quelle requête SQL pourra être exécutée sur n'importe quel SGBD.

- vrai
- faux

Question 6 La méthode query de PDO ne permet pas d'utiliser l'instruction UPDATE de SQL

- vrai
- faux

Question 7 Dans un code utilisant PDO, si une requête SQL n'est pas correcte, il est possible de poursuivre l'exécution du code.

- faux
- vrai

Question 8 En PHP, on accède aux variables de session à l'aide d'une variable super globale.

- faux
- vrai

Question 9 Une requête préparée en PDO peut tout aussi bien utiliser l'instruction SELECT ou l'instruction UPDATE de SQL

- vrai
- faux

Question 10 Il est possible de demander au protocole HTTP de maintenir une connexion entre un client et le serveur dans le cas des sessions.

- vrai
- faux

Question 11 La méthode fetch de PDO ne permet pas d'utiliser l'instruction SELECT de SQL

- faux
- vrai

CORRECTION

Question 12 ♣ En PHP, avec PDO, à quoi sert le paramètre `PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES XXX"` lors de la création de la connexion à la base de donnée ?

- à indiquer quelle table de codage de caractères (encoding) est utilisée dans la base de données
- à rien, c'est une option facultative historique qui n'a plus d'utilité aujourd'hui
- à rendre le code compatible entre windows, linux ou mac
- à indiquer le nom de la base de données (XXX)
- Aucune de ces réponses n'est correcte.

Question 13 CGI est un langage de programmation permettant d'exécuter PHP sur un serveur web

- vrai
- faux

Question 14 En PHP, quelle instruction de boucle est correcte :

- `foreach($t as $k:$v)`
- `foreach($t as $k=>$v)`

Question 15 Comment appelle-t-on les fonctions dont dispose tout objet en PHP ?

- Les fonctions globales
- Les fonctions magiques

Question 16 Avec smarty, peut-on assigner un tableau PHP à une variable de template ?

- non
- oui

Question 17 En PHP, à quoi sert l'instruction `die()` ?

- à arrêter l'interprétation du bloc courant, sortir du bloc et reprendre l'exécution.
- à arrêter complètement l'interprétation du script PHP.

Question 18 Est-il possible de programmer une boucle dans un template smarty?

- oui
- non

Question 19 Qu'est-ce qu'une assignation dans un moteur de template PHP ?

- un mécanisme permettant de choisir le fichier template à utiliser
- un mécanisme permettant d'affecter des valeurs à des variables du template

Question 20 ♣ Pourquoi smarty est-il considéré comme étant un VRAI moteur de template en PHP ?

- A Il permet de séparer la structure du contenu
- B Il est lui-même écrit en PHP
- C Il utilise un langage interprété qui n'est pas du PHP
- D Il interprète lui-même les fichiers templates
- E Aucune de ces réponses n'est correcte.

2 Deuxième partie : Exercices (40 points)

2.1 Compréhension technique (20 points)

Lisez l'article (figure 1) tiré du tutoriel *Évoluer vers une architecture MVC en PHP* de Baptiste Pesquet (<http://bpesquet.developpez.com/tutoriels/php/evoluer-architecture-mvc/>).

[...Qx...] signale une partie de l'article qui fera l'objet d'une question.

Répondez ensuite aux questions.

Question 21 4 points Q1: Expliquez en quoi consiste le mécanisme d'URL rewriting. Donnez deux exemples simples et rapide de mise en oeuvre.

Question 22 2 points

Q2: A quel fichier fait référence l'auteur ?

Question 23 4 points

Q3: Donnez la forme des URL d'entrée et la forme des URL ré-écrites.

Question 24 8 points

Q4: Que pouvez-vous dire sur la technique de templating utilisée par l'auteur? Qu'en pensez-vous?

Proposez une technique différente et indiquez les modifications à reporter dans le code (contentez-vous d'indiquer les lignes à modifier, et le contenu des modifications).

Question 25 2 points

Que pouvez-vous dire sur la méthode get utilisée ligne 9 du dernier code ?

2.2 Mise en oeuvre (20 points)

Soit le code suivant :

```

1 <?php
2 $stmt = $dbh->prepare("SELECT * FROM REGISTRY where name = ?");
3 if ($stmt->execute(array($_GET['name']))) {
4     while ($row = $stmt->fetch()) {
5         print_r($row);
6     }
7 }
8 ?>
```

Question 26 2 points

Donnez le code simple à exécuter avant la ligne 2.

Question 27 2 points

Le code donné protège-t-il des injections SQL ?

V-B-3. Mise en place d'URL génériques

L'automatisation du routage nous a conduit à définir pour notre site des URL de la forme `monsite.fr/index.php?controleur=xxx&action=yyy&id=zzz`. Afin de faciliter leur lisibilité et leur référencement, il serait souhaitable de les remplacer par des URL de la forme `monsite.fr/controleur/action/id`. Ce format d'URL est adopté par la plupart des plates-formes PHP modernes (frameworks, CMS, etc.). Par exemple, l'URL pour accéder à l'action `index` sur le billet 3 sera `monsite.fr/billet/index/3`.

Cette amélioration peut être réalisée grâce au mécanisme de réécriture d'URL (URL rewriting). [...Q1...]

Le serveur Web Apache dispose d'un puissant module de réécriture d'URL nommé `mod_rewrite`. Pour le mettre en œuvre, il faut ajouter à la racine de notre site un fichier de configuration nommé [...Q2...].

```

1 # exemple
2 RewriteEngine on
3 RewriteRule ^([a-zA-Z]*)/?(([a-zA-Z]*)?/?([a-zA-Z0-9]*)?/?$ index.php?
   controleur=$1&action=$2&id=$3 [NC,L]

```

[...Q3...]

Cependant, ce nouveau format d'URL provoque des erreurs avec les liens relatifs de nos vues (inclusion de feuilles de style, d'images, de fichiers JavaScript, etc.). En effet, le navigateur résout ces liens à partir du chemin relatif défini par la nouvelle URL et non plus depuis la racine du site. Pour résoudre ce problème, il faut ajouter dans toutes nos vues la balise HTML `base`. La solution naturelle est d'inclure cette balise dans notre gabarit commun à toutes les vues.[...Q4...]

```

1 <!-- template.php -->
2 <!doctype html>
3 <html lang="fr">
4 <head>
5 <meta charset="UTF-8" />
6 <base href="<?=$_$racineWeb_?>" >
7 ...

```

La valeur de l'élément `base` est donnée par la variable PHP `$racineWeb` qui doit être définie au moment de la génération d'une vue. Pour cela, on modifie la méthode associée dans la classe `Vue`.

```

1 //Vue.php
2 ...
3 public function generer($donnees) {
4     // Génération de la partie spécifique de la vue
5     $contenu = $this->genererFichier($this->fichier, $donnees);
6     // On définit une variable locale accessible par la vue pour la
       racine Web
7     // Il s'agit du chemin vers le site sur le serveur Web
8     // Nécessaire pour les URI de type controleur/action/id
9     $racineWeb = Configuration::get("racineWeb", "/"); [...Q5...]
10    // Génération du gabarit commun utilisant la partie spécifique
11    $vue = $this->genererFichier('Vue/gabarit.php',
12    array('titre' => $this->titre, 'contenu' => $contenu,
13    'racineWeb' => $racineWeb));
14    // Renvoi de la vue générée au navigateur
15    echo $vue;
16 }
17 ...

```

Figure 1: Évoluer vers une architecture MVC en PHP (extrait).

CORRECTION

Question 28 2 points

En quoi le code donné ne respecte pas les bonnes pratiques de codage en PHP ?

Question 29 14 points

Ré-écrivez ce code :

1. en corrigeant les problèmes relevés ;
2. en encapsulant le code dans une fonction *getInfo* prenant en paramètre un objet PDOConnexion et une chaîne de caractère name ;
3. cette fonction retournera un tableau contenant les résultat de la requête SQL. Le tableau sera vide si la requête n'a renvoyé aucun résultat ou a planté.
4. il n'y aura plus d'utilisation de print_r

Vous donnerez également un exemple complet d'utilisation de cette fonction (3 ou 4 lignes suffisent), comme si vous souhaitiez la tester.

3 Troisième partie : Synthèse (20 points)

Question 30 20 points

Donnez 4 (uniquement 4) exemples de mauvaises pratiques en web dynamique qui n'ont pas été mentionnées dans la partie exercice.

Choisissez des exemples hétérogènes, en essayant de couvrir au maximum le spectre des techniques et technologies vues pendant le module.

Vous expliquerez pour chaque exemple :

1. le ou les risques liés à cette mauvaise pratique ;
2. la ou les technologies impliquées ;
3. le ou les moyens de l'éviter.

Feuille de réponses :

0	0	0	0
1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4
5	5	5	5
6	6	6	6
7	7	7	7
8	8	8	8
9	9	9	9

Techniques et langages de l'Internet
2ème session 2017–2018
3ETI – CPE

← codez votre numéro d'identification ci-contre, et
inscrivez votre nom et prénom ci-dessous.

Nom et prénom :
.....

- Question 1 : A B
- Question 2 : A B C D E
- Question 3 : A B C D
- Question 4 : A B
- Question 5 : A B
- Question 6 : A B
- Question 7 : A B
- Question 8 : A B
- Question 9 : A B
- Question 10 : A B
- Question 11 : A B
- Question 12 : A B C D E
- Question 13 : A B
- Question 14 : A B
- Question 15 : A B
- Question 16 : A B
- Question 17 : A B

- Question 18 : A B
- Question 19 : A B
- Question 20 : A B C D E
- Question 21 : F P A B C
- Question 22 : F P C
- Question 23 : F P A B C
- Question 24 : F P A B C
- Question 25 : F P C
- Question 26 : F P C
- Question 27 : F P C
- Question 28 : F P C
- Question 29 : F P A B C
- Question 30 : . F I P A B C