

Correction de l'Examen Final – Semestre 2-

Questions de cours (8,5 pts)

1. Quelle est la différence entre application web et site web ? (1pt)

- La principale différence entre une application Web et un site Web est qu'un site Web est une collection de pages universellement accessibles et imbriquées qui contiennent un seul nom de domaine. Alors qu'une application web est un programme ou un logiciel qui est
- accessible à l'aide de n'importe quel navigateur web.

2. Quel est le principe de fonctionnement de l'architecture Client/Serveur. Donnez un exemple? (1pt)

Une architecture client-serveur représente l'environnement dans lequel des applications de machines clientes communiquent avec des applications de machines de type serveurs. L'exemple classique est le navigateur Web d'un client qui demande (on parle de "requête") le contenu d'une page Web à un serveur Web qui lui renvoie le résultat (on parle de "réponse"),

3. Pour coder en PHP, quels sont les 3 produits incontournables de XAMP à utiliser? (1pt)

Pour coder en PHP, on a besoin des 3 produits incontournables:

- Le serveur **Web Apache**
- Le moteur de scripts **PHP**
- Le Système de Gestion de base de données **MySQL**

Apache, MySQL, PHP : Ces trois noms désignent des logiciels essentiels, installés sur le site web (machine serveur), ayant chacun des rôles différents.

4. Quels sont les acronymes CMS ? Donnez 3 CMS les plus populaires du marché. (1pt)

CMS est l'acronyme de Content Management System, c'est-à-dire système de gestion de contenu. **3 CMS les plus populaires du marché :**

WordPress – Drupal - Joomla

5. Citer les étapes pour valider un code HTML on line. **(1pt)**

Validation sur le site du w3c :

Le site distant du w3c permet de vérifier la conformité HTML5 d'une page html :

- en saisissant son url si elle est disponible sur un site distant : <https://validator.w3.org/>
- en saisissant son emplacement si elle est disponible localement en tant que fichier : https://validator.w3.org/#validate_by_upload
- par copié / collé : https://validator.w3.org/#validate_by_input

Validation de code HTML par par copié / collé :

- ouvrir la page à valider avec un éditeur de texte (visualstudio code, medit, geany, ...)
- sélectionner l'ensemble du texte (Ctrl A)
- le copier (Ctrl C)
- le coller dans la fenêtre de la page sur le site w3c (Ctrl V): validator.w3.org
- et cliquer sur "check"

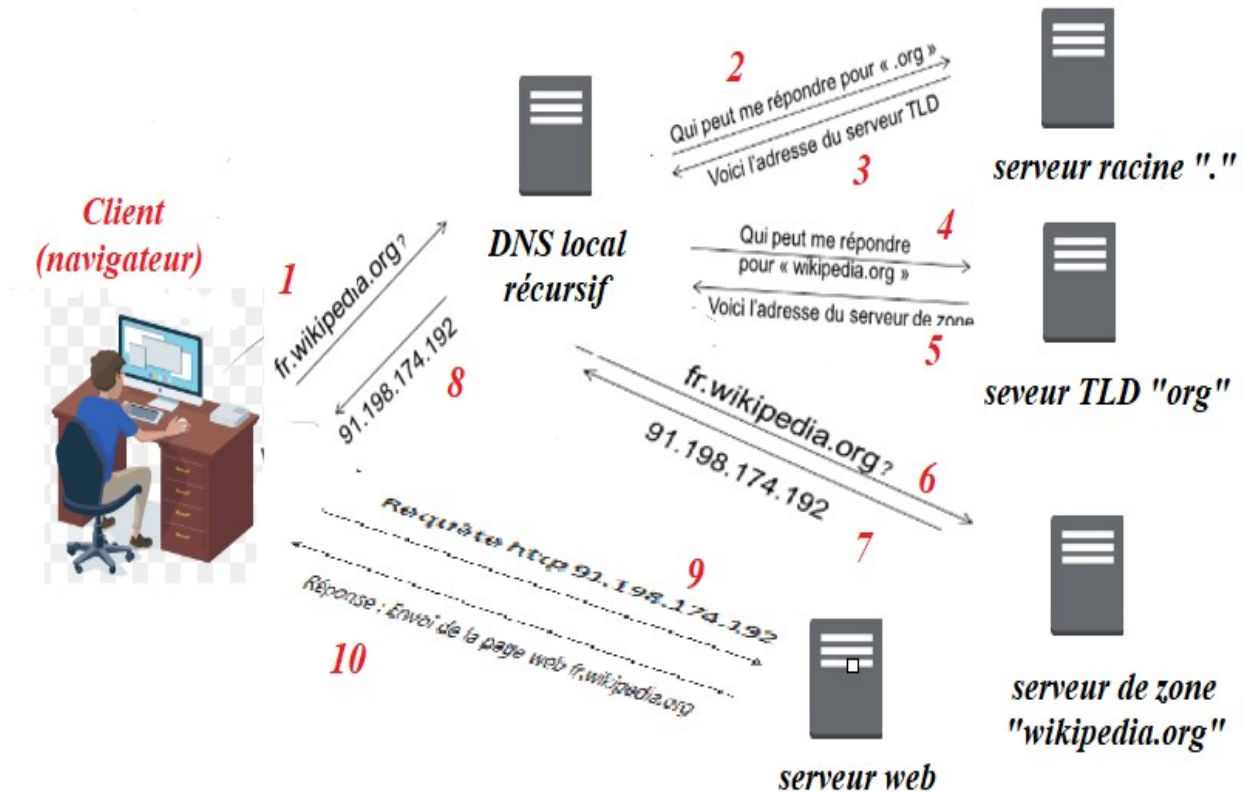
6. Citer 3 catégories de serveurs informatiques. **(1pt)**

Compte tenu de la multitude de services qu'il propose, le serveur informatique se décline en plusieurs catégories.

- Les serveurs web : HTTPS
- Les serveurs de fichiers : FTP
- Les serveurs de bases de données:
- Les serveurs de messagerie :
- Les serveurs DNS :
- Les serveurs d'impression :
- Les serveurs d'applicatifs métiers :
- Les serveurs PROXY :

➤ Le serveur virtuel

7. Remplir les vides en pointillé et donner un titre à la figure: (2.5 pts)



Titre : Architecture Client/Serveur avec Serveur de noms DNS -

Exercices

Exercice1 : HTML (2 pts)

- Écrivez un document HTML qui reproduit le formulaire suivant :

Changement de mot de passe

Entrer votre nom d'utilisateur :

Entrer votre ancien mot de passe :

Entrer votre nouveau mot de passe :

Confirmer votre nouveau mot de passe :

```

<!DOCTYPE html">
<html>
<head>
<title>Changer Password</title>
</head>
<body>
<h1>Changement de mot de passe </h1>
<form method="get" action="">
<p>
<label for="user">Entrer votre nom d'utilisateur :</label>
<input type="text" name="user" id="user" value="" />
</p>
<p>
<label for="oldPass">Entrer votre ancien mot de passe :</label>
<input type="password" name="oldPass" id="oldPass" value="" />
</p>
<p>
<label for="newPass">Entrer votre nouveau mot de passe :</label>
<input type="password" name="newPass" id="newPass" value="" />
</p>
<p>
<label for="ConfirmPass">Confirmer votre nouveau mot de passe :</label>
<input type="password" name="ConfirmPass" id="ConfirmPass" value="" />
</p>
<p>
<input type="submit" name="submit" value="Changer le mot de passe" />
<input type="reset" name="reset" value="Annuler" />
</p>
</form>
</body>
</html>

```

- Citer les deux attributs et les méthodes indispensables (les plus utilisés) pour construire un formulaire fonctionnel.

Pour créer un formulaire, vous aurez besoin de la balise HTML **<form>** **</form>**, elle permet d'indiquer le début et la fin du formulaire.

Voici les deux attributs indispensables pour construire un formulaire fonctionnel :

1. L'attribut **method** indique par quel moyen les données vont être envoyées.
2. Les méthodes les plus utilisées sont **get** et **post**.
3. L'attribut **action** indique l'adresse de la page ou du programme qui va traiter les informations.

Tout cela va nous permettre de créer la structure de base d'un formulaire en HTML :

Dans notre programme HTML cité ci-dessus je donne la valeur **get** à l'attribut **method** ; et je laisse un champ vide : " " pour **action**.

Pourquoi est-ce qu'on le laisse un champ vide dans l'attribut action ?

Si notre projet comportait également du code en **backend*, c'est là qu'on indiquerait une URL qui pourrait traiter notre formulaire. Mais ici, en laissant un champ vide, on indique à notre formulaire qu'on restera sur la même URL, ce qui nous permettra plus tard d'inspecter ce qu'on obtient, assez simplement.

***En règle générale, un site internet est composé de deux parties :

1. une partie **frontend**, c'est la partie visible de notre site internet, celle qui s'affiche dans votre navigateur grâce à HTML, CSS et JavaScript ;
2. une partie **backend**, qui se trouve hébergée et qui s'exécute sur un serveur web. Elle permet de faire des tâches plus complexes, comme traiter un formulaire, envoyer un mail, sauvegarder des informations dans une base de données, etc.

Exercice 2 : HTML, PHP (6 pts)

Soit la Base de données « Etudiant », composée de la table « Notes » avec les champs : Matricule, Nom , Prenom, MoySem1, MoySem2, MoyGen.

- Créer un formulaire de recherche permettant de retrouver tous les étudiants dont la moyenne est supérieure à une valeur saisie. On choisit quel semestre et on choisit la moyenne minimale à saisir. Pour cela :

1. Ecrire la page HTML « formulaire.html » contenant un formulaire de recherche avec deux champs :
 - Le premier sous forme d'une liste déroulante avec 3 options (Sem1, Sem2, Général).
 - Le deuxième est un champ où on saisit la moyenne minimale.
2. Écrire le script php « mysqlBdd.php » contenant une fonction connecterBase() permettant la connexion au serveur de la machine locale et le choix de la base de données.
3. Ecrire le script php « recherche.php » permettant de :
 - Se connecter à la base de données Etudiant.
 - Effectuer la requête sql demandée pour MoySem1 et moyenne minimale = 12 .

Le fichier formulaire.html

```
<html>
<head>
<title> Formulaire de Recherche </title>
</head>
<body>
<form action="recherche.php" method="get" name = "form1" onsubmit="return valider();">
  Marque : <select name="marque" />
    <option value="1">SEM1
    <option value="2">SEM2
    <option value="3">Général
  </select>
  <br>
  Modèle : <input type="text" name="moyenne" /><br>
    <input type="submit" value="Rechercher" />
</form>
</body>
</html>
```

Le fichier mysqlBdd.php

```
<?php
function ConnecterBase()
{
  $host = "localhost"; $username = "root"; $password = " "; $database = "Etudiant";
  $connect = mysqli_connect($host, $username, $password, $database)
  or die("Erreur " . mysqli_error($connect)) ;
  return $connect;
}
?>
```

Le fichier recherche.php

```
<?php
include("mysqlBdd.php");
$connexion = ConnecterBase();
$table = "Notes";
$semestre=$_GET['semestre'];
$moyenne=$_GET['moyenne'];
if(($semestre=="")&&($moyenne=="")){
  echo "Veuillez choisir un Semestre et/ou saisir la moyenne !" ;
}else{
  $select = "SELECT * FROM $table WHERE MoySem1> '$moyenne' " ;
  $result = mysqli_query($connexion, $select)
  or die("Erreur lors de la consultation de la BD." . mysqli_error($connexion));
}
?>
```

Exercice 3: HTML, PHP (1.5 pts)

Ecrire une classe PHP Calculator qui permet de calculer la somme, la soustraction, la multiplication, et la division de deux valeurs réelles x et y.

```
<?php
class Calculator {

    // déclaration d'une propriété

    public $x;
    public $y;

    // déclaration des méthodes

    public function Addition ($x,$y) {
        | echo $this->x+y;
    }
    public function Soustraction($x,$y) {
        | echo $this->x*y;
    }

    public function Multiplication($x,$y) {
        | echo $this->x*y;
    }

    public function Division($x,$y) {
        | if (y <> 0) { echo $this->x/y; }
    }
}

$cal=new Calculator;
$cal->Addition(1,2);
echo $cal;
?>
```

Exercice 4: HTML, PHP (2 pts)

Ecrire un script PHP qui calcule le factoriel d'un nombre entier n dans les 2 cas :

1. En utilisant une boucle.

2. En utilisant une fonction récursive.
3. Comment choisir la meilleure stratégie de programmation (Boucle ou Fonction Récursive).

Programme avec boucle for :

```
<?php
function facto($n)
{
    $fact=1;
    for ($i=1; $i<=$n; $i++){
        $fact*=$i;
    }

    return $fact;
}
// Affichage
// Exemple d'Appel de la fonction avec n=100
echo "factorielle =",facto(100);
?>
```

Programme avec fonction récursive :

```
<?php
function facto($n)
{
    if ($n==0) return 1;
    else {return $n*facto($n-1);}
}
// Affichage
// Exemple d'Appel de la fonction avec n=100
echo "factorielle =",facto(100);
?>
```

- Choisir la meilleure stratégie de programmation revient à calculer la complexité algorithmique de chaque programme et choisir celui qui a la plus petite complexité.

Nota : La syntaxe doit être correcte car elle est prise en considération dans la notation.

Bon courage.