

## L2, développement *Web*

(tous documents autorisés)

### 1. Un peu de PHP conceptuel

Produire juste le fragment suivant qui est valide pour la grammaire *XHTML 1.0 Strict* à l'aide des fonctions de `std.php` :

```
<table border="1" cellpadding="10" class='collapse'>

  <tr>
    <th align='left'> Saison </th>
    <th align='left'> Episode </th>
    <th align='left'> Titre </th>
    <th align='left'> Langue </th>
  </tr>

  <tr>
    <td align='right'> 1 </td>
    <td align='right'> 1 </td>
    <td align='left'> L'hiver vient </td>
    <td align='left'> FR </td>
  </tr>

</table>
```

Il n'y a pas besoin de préciser `d'include()` ni `debutPage()` ou `finPage()`, ni de gérer l'indentation du code XHTML produit par PHP. Il est conseillé d'utiliser `entetesTableau()`.

## 2. MYSQL seul puis couplé à PHP

On veut gérer l'affichage des épisodes de séries télévisées en MYSQL. Pour cela on suppose qu'on dispose de quatre tables de données dans la base STADATA nommées `series`, `episodes`, `artistes` et `personnages`.

- la table `series` contient, entre autres, les champs : `idSerie`, clé identifiante de la série ; `nomSerie`, son nom pour le grand public ; `acroSerie`, son acronyme (par exemple `GoT` pour *Game of Thrones*).
- la table `episodes` contient, entre autres, les champs : `idEpisode`, clé identifiante de l'épisode ; `numSerie`, valeur censée être reliée à l'identifiant d'une série ; `numSaison`, numéro de la saison de l'épisode ; `numEpisode`, numéro de l'épisode dans la saison ; `nomEpisode`, nom de l'épisode.
- la table `artistes` contient, entre autres, les champs : `idArtiste`, clé identifiante de l'artiste ; `prenomArtiste`, prénom de l'artiste ; `nomArtiste`, nom de l'artiste.
- la table `personnages` contient, entre autres, les champs : `idPersonnage`, clé identifiante du personnage ; `artiste`, valeur censée être reliée à l'identifiant d'un artiste ; `episode`, valeur censée être reliée à l'identifiant d'un épisode ; `nomPersonnage`, `prenomPersonnage`, nom et prénom du personnage.

### *Question 2.1*

Donner une expression MySQL qui permet de compter le nombre de séries en tout.

### *Question 2.2*

Donner une expression MySQL qui permet de compter le nombre d'épisodes (dans la base de données) de la série dont le nom grand public est "*Game of Thrones*", toutes saisons confondues.

### *Question 2.3*

Donner une expression MySQL qui affiche tous les noms et prénoms des personnages avec aussi leur nom et leur prénom d'artiste.

On affichera les informations par nom de personnage croissant, puis par ordre de prénom de personnage croissant en cas d'égalité de nom de personnage.

Pour les plus fort(e)s d'entre vous, on pourra compléter la requête pour n'afficher que les personnages de la série dont le nom grand public est "*Plus belle la vie*".

#### Question 2.4

Soit `$qry` la chaîne de caractères PHP définie par :

```
$qry = "" ;
$qry .= " SELECT      numSaison, numEpisode, nomSerie    " ;
$qry .= " FROM        series, episodes                  " ;
$qry .= " WHERE       series.idSerie=episodes.numSerie  " ;
$qry .= " AND         nomSerie='Game of Thrones'       " ;
$qry .= " ORDER BY   numSaison ASC, numEpisode ASC ;    " ;
```

Donner le code PHP des instructions qui :

- exécutent la requête MySQL associée à `$qry` ; si la requête échoue, le code doit afficher "Erreur REQUETE" et quitter le script PHP ; si la requête réussit, la ressource obtenue sera mise dans la variable `$res`.
- convertissent la ressource `$res` en un tableau PHP nommé `$tabRes` autant de fois que nécessaire et qui en affichent le contenu dans une ligne d'un tableau XHTML avec trois éléments `<td>`.

On fournira un code complet qui produit, dans les cas où la requête réussit, un tableau XHTML complet et valide. Là encore, pas besoin du début de la page ni de la fin de la page. Juste un fragment XHTML complet et valide.

Il est très fortement recommandé d'utiliser du PHP conceptuel à l'aide des fonctions de `std.php` de façon à écrire un minimum de code PHP. On cadrera bien sûr à droite les nombres et à gauche les chaînes de caractères.

### 3. Une discussion argumentée

Répondez à la question suivante en dix lignes au minimum, sans limite de maximum. Votre réponse devra mettre en évidence votre culture, votre recul et votre esprit de synthèse quant à la programmation Web et à ses langages.

Il est conseillé d'utiliser au moins trois mots de trois syllabes ou plus pour « transmettre un contenu rédactionnel fort ».

*Est-il facile d'interfacer PHP et Javascript ? Et surtout, pourquoi ?*