

**Examen  
Semestre 3  
Deuxième session  
2020/2021**

**Licence 2 Informatique**

**EC : INFO0303**

**Intitulé : Technologies Web 2**

**Responsable : Cyril RABAT**

Durée de l'épreuve : 1h30

Nombre de page(s) : 7

**RAPPEL**

**Aucun document, outil de calcul ou de communication autorisé. Sauf indications contraires, vous ne devez pas spécifier de style dans vos scripts. Toute réponse doit être justifiée.**

---

**Sujet : M. Cyril Rabat (20 points)**

### Exercice 1 (QCM - 5 points)

**Vous ne devez pas répondre sur le sujet, mais sur votre copie.** Pour chaque question, indiquez le numéro de la question et la/les lettre(s) de la/des bonne(s) réponse(s).

1°) Dans une application Web écrite en PHP, quel protocole de communication est utilisé entre le client et le serveur Web ?

- a) UDP                              b) ♥ HTTP                              c) HTML                              d) PHP

2°) Quelle(s) affirmation(s) est/sont juste(s) concernant PHP ?

- a) PHP peut exécuter du SQL  
b) ♥ PHP peut générer du code *Javascript* à la volée  
c) ♥ PHP peut générer du code HTML à la volée  
d) PHP peut exécuter du *Javascript*

3°) Quel affichage produit le code suivant ?

```
<?php
$a = &$b;
$b++;
echo $a;
```

- a) Un message d'erreur      b) 0                              c) ♥ 1                              d) Rien n'est affiché

4°) Quel affichage produit le code suivant ?

```
<?php
$cyril = "Cyril";
$t = 'Bonjour $cyril';
echo $t;
```

- a) Un message d'erreur      b) ♥ Bonjour \$cyril              c) Bonjour Cyril              d) Salut, ça va ?

5°) Quelle(s) affirmation(s) est/sont juste(s) concernant l'héritage en PHP ?

- a) L'héritage d'une classe se fait à l'aide du mot clé **herits**  
b) L'héritage multiple est autorisé en PHP  
c) ♥ L'héritage d'une classe se fait à l'aide du mot clé **extends**  
d) L'héritage en PHP utilise les traits

6°) Quelle(s) affirmation(s) concernant les sessions est/sont juste(s) ?

- a) Lors du traitement d'une requête AJAX, il n'est pas possible de récupérer les données de la session  
b) Lors du traitement d'une requête AJAX, il est possible de récupérer les données de la session  
c) ♥ La session est supprimée lorsque l'onglet du navigateur est fermé  
d) Il est possible de manipuler directement les données de la session avec *Javascript*

7°) Quelle(s) affirmation(s) concernant les sessions est/sont juste(s) ?

- a) ♥ Les sessions permettent de stocker tous les types de données du côté du serveur  
b) Les sessions permettent de stocker uniquement des données de type primitif du côté du client  
c) Les sessions permettent de stocker uniquement des données de type primitif du côté du serveur  
d) Les sessions permettent de stocker tous les types de données du côté du client

8°) Que peut-on dire sur le routeur dans le modèle MVC ?

- a) ♥ Il décide vers quel contrôleur envoyer la requête en fonction des routes  
b) Il achemine les paquets IP vers les bons équipements  
c) Il charge les données et les envoie à la vue  
d) ♥ Il reçoit toutes les requêtes de l'application

9°) Quel affichage produit le code suivant ?

```
<?php
trait Titi {
    private function tata() { echo "Bonjour"; }
}
class Toto {
    use Titi;
}
$toto = new Toto();
print_r($toto->tata());
```

- a) Une erreur est affichée
- b) Rien n'est affiché
- c) ♥ Bonjour
- d) Au revoir Président

10°) Quel affichage produit le code suivant ?

```
<?php
namespace App\Classes;

class Toto {

}
namespace App\Main;

$toto = new App\Classes\Toto();
print_r($toto);
```

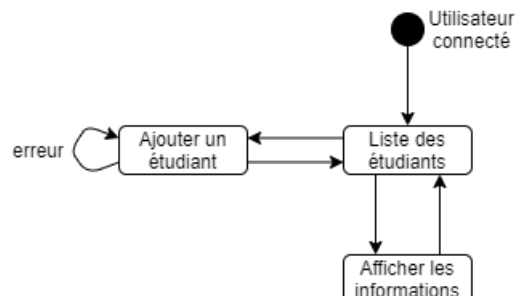
- a) Un message d'erreur
- b) ♥ App\Classes\Toto Object()
- c) Une autre réponse
- d) Object()

### Exercice 2 (Laravel - 4 points)

Nous souhaitons mettre en place une application permettant de gérer des étudiants, accessible uniquement à un utilisateur connecté. Il peut consulter la liste des étudiants, afficher les informations concernant l'un d'eux et d'ajouter un nouvel étudiant.

1°) Proposez un diagramme de navigation. (1 point)

**Solution :** *Voici une solution :*



2°) Expliquez tous les éléments que nous devons mettre en place dans *Laravel* pour implémenter cette application. (2 points)

**Solution :** *Éléments à mettre en place selon un diagramme de navigation (suivant la version) :*

- vues : le template, la vue liste, la vue information, la vue ajouter
- contrôleur : pour étudiant avec les fonctions pour lister, les informations, l'ajout + création BD

- *modèle* : le modèle étudiant (quelles données ?)
- *migration* : avec la création de la table étudiants
- *les routes (web.php) et le setup*
- *Optionnel* : seeder, connexion (UI)

3°) Expliquez ce que l'on doit mettre en place sous *Laravel* pour réaliser des requêtes *AJaX*. (1 point)

**Solution** : Il faut d'abord un contrôleur qui répondra aux différentes requêtes (et la route associée). Dans la vue, il faut un script Javascript qui envoie les données, parmi lesquelles il faut spécifier le token pour la validité des données.

### Exercice 3 (PHP - 6 points)

1°) Écrivez une classe PHP permettant de représenter un animal caractérisé par : un nom et une date de naissance. Vous préciserez les méthodes classiques : constructeur(s), accesseur(s), méthode(s) de conversion en chaîne de caractères. (2 points)

**Solution** : Voici un exemple :

```
class Animal {

    private string $nom;
    private string $dateNaissance;

    public function __construct(string $nom, string $dateNaissance) {
        $this->nom = $nom;
        $this->dateNaissance = $dateNaissance;
    }

    public function getNom() : string { return $this->nom; }
    public function getDateNaissance() : string { return $this->dateNaissance; }

    public function setNom(string $nom) : void { $this->nom = $nom; }
    public function setDateNaissance(string $dateNaissance) : void {
        $this->dateNaissance = $dateNaissance;
    }

    public function __toString() : string {
        return $this->nom . "_" . $this->dateNaissance . " ";
    }
}
```

2°) Écrivez un formulaire HTML permettant de saisir les informations d'un animal. (1 point)

**Solution** : Voici un exemple :

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <title>Exercice 1</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, _initial-scale=1">
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.3/
      css/bootstrap.min.css">
  </head>
  <body>
    <div class="container">
      <div class="card">
        <div class="card-header _bg-primary _text-white _text-center">Nouvel animal</div>
        <div class="card-body _bg-light">
```

```

<form action='valider.php' method='post'>
  <div class="form-group_row">
    <label for="inputNom" class="col-sm-2_col-form-label">Nom</label>
    <div class="col-sm-10">
      <input type="text" class="form-control" autofocus name="nom" id="
        inputNom" placeholder="Saisissez le nom"/>
    </div>
  </div>
  <div class="form-group_row">
    <label for="inputDateNaissance" class="col-sm-2_col-form-label">Date
      de naissance</label>
    <div class="col-sm-10">
      <input type="text" class="form-control" name="dateNaissance" id="
        inputDateNaissance" placeholder="Saisissez la date de
        naissance"/>
    </div>
  </div>
  <button class="btn btn-outline-primary" type="submit">Valider</button>
</div>
</div>
</body>
</html>

```

3°) Écrivez une méthode **fromForm** dans la classe PHP **Animal** qui permet de créer un objet animal à partir des informations saisies dans ce formulaire. Vous ajouterez les vérifications nécessaires. (2 points)

**Solution :** Voici un exemple :

```

static public function fromForm() : Animal {
    if(isset($_POST['nom']) && ($_POST['nom'] != "") &&
        isset($_POST['dateNaissance']) && ($_POST['dateNaissance'] != ""))
        return new Animal($_POST['nom'], $_POST['dateNaissance']);
    else
        return null;
}

```

4°) Nous souhaitons proposer à un utilisateur de modifier un animal via un formulaire. L'identifiant de l'animal est stocké en session. Expliquez quels éléments doivent être mis en place dans les différents scripts en donnant les inscriptions PHP spécifiques pour la session et expliquez l'intérêt d'une telle solution. (1 point)

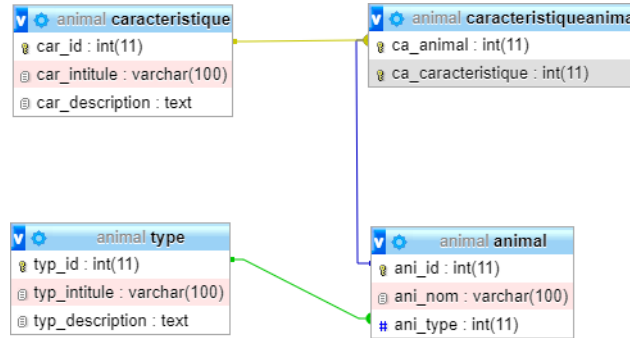
**Solution :** Il faut démarrer la session avec **session\_start()**. L'identifiant de l'animal à modifier est placé en session dans une variable. On peut également placer l'identifiant dans le formulaire dans un champ caché. Lorsque le formulaire est validé, on récupère les données envoyées depuis le formulaire (via la variable **\$\_POST**) et on peut les associer à l'animal dont l'identifiant a été sauvegardé. L'intérêt d'une telle solution est de vérifier que seul l'animal spécifié peut être modifié. Dans le cas contraire, il serait possible de modifier les données du formulaire pour spécifier l'identifiant d'un autre animal.

#### Exercice 4 (Base de données - 5 points)

Un animal est caractérisé par un nom, une date de naissance, un type (otarie, phoque, etc.) et une ou plusieurs caractéristiques physiques. Un type est caractérisé par un intitulé et une description longue. Une caractéristique physique est caractérisée par un intitulé et une description longue.

1°) Donnez le contenu des tables de la base de données, en précisant les types de données et les clés. (2 points)

**Solution :** Voici une solution :



2°) Écrivez le contenu de la classe CRUD PHP correspondant à un animal. (3 points)

**Solution :** Voici une solution :

```

class AnimalModel {

    /**
     * Retourne un animal à partir d'un tableau associatif.
     * @param array le tableau associatif
     * @return l'animal
     */
    public static function fromArray(array $array) : Animal {
        return new Animal($array['ani_id'], $array['ani_nom'], $array['ani_type'])
    }

    /**
     * Ajoute un animal dans la base.
     * @param animal l'animal à ajouter dans la base
     * @return 'true' en cas de réussite ou 'false' en cas d'erreur
     */
    public static function add(Animal $animal) : bool {
        $DB = MyPDO::getInstance();
        $SQL = "INSERT INTO `animal` (`ani_id`, `ani_nom`, `ani_type`) VALUES " .
            "(NULL, :nom, :type)";

        if (($requete = $DB->prepare($SQL)) &&
            $requete->execute(array(":nom" => $animal->getNom(),
                                   ":type" => $animal->getType()))) {
            $animal->setId($DB->lastInsertId());
            return true;
        }
        else
            return false;
    }

    /**
     * Récupère un animal depuis la base de données.
     * @param id l'identifiant de l'animal
     * @return l'utilisateur ou 'null' en cas d'erreur
     */
    public static function load(int $id) : Animal {
        $DB = MyPDO::getInstance();
        $SQL = "SELECT * FROM `animal` WHERE `ani_id`=:id";

        if (($requete = $DB->prepare($SQL)) &&
            $requete->execute(array(":id" => $id)) &&
            ($row = $requete->fetch()))
            return self::fromArray($row);
        else
    
```

```

        return null;
    }

/**
 * Sauve les modifications d'un animal dans la base.
 * @param animal l'animal
 * @return 'true' on success or 'false' on error
 */
public static function update(Animal $animal) : bool {
    $DB = MyPDO::getInstance();
    $SQL = "UPDATE `animal` SET `ani_nom`=:nom, `ani_type`=:type".
        "WHERE `ani_id`=:id";

    return (($requete = $DB->prepare($SQL)) &&
        $requete->execute(array(":id" => $animal->getId(),
                                ":nom" => $animal->getNom(),
                                ":type" => $animal->getType())));
}

/**
 * Supprime un animal de la base.
 * @param id l'identifiant de l'animal
 * @return 'true' on success or 'false' on error
 */
public static function delete(int $id) : bool {
    $DB = MyPDO::getInstance();
    $SQL = "DELETE FROM `animal` WHERE `ani_id`=:id LIMIT 1";

    if(($requete = $DB->prepare($SQL)) && $requete->execute(array(":id" => $id)))
        return true;
    else
        return false;
}
}

```