

# Internet et Outils (IO2)

## Examen de session 1 (durée 2h)

2 mai 2018

*Documents autorisés : trois feuilles A4 recto-verso manuscrites ou imprimées.  
Les ordinateurs, les téléphones et les tablettes sont interdits.  
Le barème est donné à titre indicatif.*

Lisez bien tout le sujet avant de vous mettre au travail. Certaines questions dépendent d'autres, mais vous pouvez les aborder même sans avoir terminé les questions précédentes.

### A. HTML et CSS

**Question 1 — 4 points** Écrire une page `Q1.html` comportant un bloc principal qui contient 12 blocs, chacun contenant un article avec un titre et un peu de texte. (Les 12 blocs seront tous identiques, il suffit d'en écrire un seul.) Y rattacher une feuille de style `Q1.css`. Écrire `Q1.css` pour obtenir les effets décrits dans les questions ci-dessous :

- Le bloc principal est centré, sur fond bleu clair et occupe 80 % de la largeur de la page.
- Les blocs contenant les articles sont organisés en rangées : un même nombre d'articles sur chaque rangée ; ce nombre ne change pas quand la fenêtre change de taille. Faire en sorte qu'en changeant uniquement la classe du bloc principal, on puisse obtenir des rangées de deux, ou de trois ou de quatre.
- Les articles ont fond blanc, bordure grise et ne sont pas collés l'un à l'autre, ni aux bords du bloc principal (et ainsi laissent ressortir sa couleur de fond). De plus il sont également espacés entre eux, et le texte des articles n'est pas collé au bord gris.
- Quand la souris survole un article, il prend couleur de fond jaune et son titre devient souligné.

### B. Développement serveur et bases de données

Dans les réponses aux questions suivantes, veillez à utiliser les fonctions PHP pour factoriser votre code, pouvoir le réutiliser et le rendre lisible. Le but est de réaliser un site (très simplifié) pour un service de réservation de taxi d'une ville. Il est demandé de réaliser uniquement un système de réservation qui interagit avec les utilisateurs ; nous ne nous soucierons pas de l'interaction avec les chauffeurs.

**Attention** : Toutes les pages du site doivent être attachées à la même feuille de style `monStyle.css` (qu'il n'est pas nécessaire d'écrire), et doivent comporter, en pied de page, l'adresse email du Webmaster.

**Question 2 — 4 points** Définir un schéma de base de données relationnelle qui permet de représenter plusieurs informations sur les taxis et les courses.

Un taxi a une plaque d'immatriculation et une catégorie de véhicule, ainsi que une adresse représentant la dernière position dans laquelle le taxi était en arrêt. Cela veut dire qu'un taxi qui est libre est dans cette position, alors qu'un taxi qui est en train d'effectuer une course est parti de cette position avant de se rendre à l'adresse de départ de la course.

Chaque adresse a un identifiant entier, un numéro de rue et un nom de rue.

Une catégorie de véhicule a un identifiant, un nom et une image associée (représenter dans la base de données uniquement le nom du fichier image).

Une course a une adresse de départ, une adresse d'arrivée, une heure de départ et une heure d'arrivée. Cette dernière est NULL si la course est en cours. L'heure de départ est également NULL si la course est réservée mais le taxi n'est pas encore arrivé à l'adresse de départ.

Pour plus d'efficacité la base de données stocke également l'ensemble des taxis libres (i.e. les taxis qui n'apparaissent dans aucune course en cours) bien que cette information soit redondante.

Définir les tables nécessaires en faisant attention à ne pas introduire d'autres redondances dans le schéma.

Prévoir également une table qui décrit, pour chaque couple d'adresses, leur distance ainsi qu'une estimation du temps nécessaire à la parcourir.

Produire les commandes SQL de CREATE TABLE correspondantes. Veiller à bien introduire des contraintes de clef primaire, et de clef étrangère, et toute autre contrainte nécessaire pour représenter la spécification ci-dessus.

**Question 3 — 4 points** Écrire un script nommé `home.php` présentant la liste de tous les taxis disponibles (i.e. libres), récupérés de la base de données. Pour chaque taxi la page montre la photo de la catégorie du véhicule et sa position (adresse). Chaque description de taxi est accompagnée d'un radio button qui permet de le sélectionner. Le formulaire qui contient ces radio buttons est géré par `home.php` lui même, de la façon demandée dans le prochain exercice. Ce formulaire contient également en début de page :

- le bouton de soumission nommé *Soumettre*,
- deux champs de texte pour l'adresse de départ et d'arrivée.

Veillez à insérer suffisamment d'informations dans les radio buttons pour que leur soumission permette au serveur d'identifier le taxi choisi. Il est fortement conseillé d'utiliser des fonctions pour écrire les différentes parties de la page, car il sera demandé dans les points suivants de réécrire `home.php` pour qu'il n'affiche pas toujours les mêmes éléments du formulaire.

**Question 4 — 6 points** modifier `home.php` (qui sera maintenant nommée `home1.php`) pour gérer la soumission du formulaire comme suit.

Si l'utilisateur a tout renseigné (taxi, adresse de départ et d'arrivée), et les adresses sont valides (i.e. présents dans la base de données), les modifications nécessaires sont faites dans les bases de données et une page de confirmation est affichée. Cette page affiche un message de réservation effectuée, ainsi qu'une estimation du temps d'attente (pour le déplacement du taxi de son point d'arrêt à l'adresse de départ).

Les modifications à faire dans la base de données sont les suivantes :

- le taxi est retiré de la table listant les taxis disponibles ;
- une nouvelle course est ajoutée avec les adresses de départ et d'arrivée (mais sans heure de départ, ni d'arrivée).

Si l'utilisateur n'a pas tout renseigné (parmi : adresse de départ, adresse d'arrivée et taxi), ou si les adresses sont introuvables, la page est reaffichée comme à l'exercice 3, exception faites des points suivants :

- les champs de texte qui ont été renseignés seront affichés pre-rempli avec les valeurs précédemment soumises ;
- une erreur d'adresse introuvable est affichée à côté de chaque champs d'adresse dont la saisie était erronée, le cas échéant ;
- si le taxi a été renseigné, uniquement ce taxi sera listé au lieu de tous les taxis disponibles ;
- si une adresse de départ valide a été renseignée, mais pas le taxi, uniquement les taxis disponibles à moins de 2 km de l'adresse de départ sont listés. De plus pour chaque taxi, une estimation du temps d'attente est affichée ;
- si ni le taxi ni une adresse de départ valide n'ont été renseignés, alors tous les taxis disponibles sont listés.

**Question 5 — 2 points** Modifier le schéma de la base de données pour représenter également les utilisateurs du site (avec nom, prénom, adresse par défaut, login et mdp). Chaque course devra maintenant représenter également l'utilisateur qui l'a réservée. Veiller à bien concevoir le schéma pour éviter les redondances. Produire les commandes SQL de CREATE TABLE correspondantes. Veiller à bien définir toutes les contraintes de clef étrangère.

**Question 6 — bonus** 1. Introduire une page de login qui vérifie les informations de login dans la table des utilisateurs, dans la base de données. La page s'occupe de démarrer une session PHP en cas de succès, et redirige ensuite vers `home.php`. Veiller à stocker les bonnes informations dans les variables de session. Sans les réécrire entièrement, indiquer quelles modifications sont nécessaires dans `home1.php` et dans la base de données pour que :

- l'accès à la home soit protégé par login ;
- quand ni le taxi ni l'adresse de départ ne sont renseignées, la page affiche les taxis à moins de deux km de l'adresse par défaut de l'utilisateur, ainsi qu'une estimation du temps d'attente ;
- chaque course dans la base de données indique également l'utilisateur qui l'a réservée ;
- un lien *Déconnexion* soit affiché sur la page.

2. Écrire le script `deconnexion.php` pointé par ce lien, qui s'occupe de détruire la session courante, et rediriger vers la page principale.