

## TP n° 1 : Révisions

La documentation complète des éléments `html` se trouve ici :

<https://html.spec.whatwg.org/multipage/#toc-semantic>

En préliminaire à ce TD, commencez par créer un fichier `html` nommée `html0.html` composée d'une section `head` et d'une section `body`. Spécifiez la valeur de l'attribut `lang` de la page comme `"fr"`. Dans la section `head` :

- ajoutez un `méta`-élément spécifiant par son attribut `charset` l'encodage de votre fichier (UTF-8 sous `emacs`).
- ajoutez un `title`-élément spécifiant le titre de la page.

Ce fichier servira de canevas à chacun des fichiers `html` à écrire dans les exercices ci-dessous.

## I) HTML

1. **Objets de base** Créer un fichier `html` nommée `html1.html`. Dans sa section `body` :

- afficher « Bonjour monde! »,
- ajouter un `a`-élément spécifiant un lien via son attribut `href`, et décrit par un texte,
- ajouter un `a`-élément spécifiant un lien décrit par une image (locale) via un `img`-élément (vous trouverez une image `bresil_couleur.jpg` dans l'archive `tp1.zip`),
- afficher deux `ul`-éléments imbriqués, ayant les mêmes puces via leur attribut `style ("circle")`.

2. **Formulaire** Dans un fichier `formulaire.html`, créer un formulaire à l'aide d'un `form`-élément, contenant les éléments ci-dessous. Chaque élément interactif de ce formulaire (sauf son bouton) devra être muni, outre son nom, d'un identifiant unique (`id`) associé à un `label` via son attribut `for`. Les éléments du formulaire seront disposés dans un `table`-élément sur deux colonnes : les labels à gauche, les éléments interactifs à droite. Choisissez des noms et des textes parlants.

- un champ texte permettant d'entrer un nom (un `input`-élément),
- un champ permettant d'entrer un mot de passe (en masquant ses caractères),
- un groupe vertical de cases cochables permettant de sélectionner quatre aliments au plus parmi : des frites, des pâtes, des épinards, des huîtres,
- un `select`-élément formant un menu déroulant permettant de sélectionner un unique aliment parmi : une bavette, un steak, du poulet, du canard,
- un `button`-élément permettant d'envoyer les données du formulaire à la même page (`post`).

## II) CSS

3. **Mise en page et mise en forme du texte.** Créer un fichier `html` nommé `html2.html` contenant éléments suivants :

- un titre et un sous-titre, dans des `h1` et `h2`-éléments ;
- un menu de navigation (un `nav`-élément) contenant quelques liens.
- un `article` comprenant :
  - un titre,
  - un paragraphe vide (un `p`-élément) vide, mais nommé (`name`),
  - un `script` insérant dans la paragraphe précédent la date d'aujourd'hui,
  - deux paragraphes de texte (vous trouverez des textes dans l'archive `tp1.zip`),
  - une image entre les deux paragraphes de texte (de même, `galere.jpg`, mais libre à vous de prendre les images ou les textes que vous voulez) ;

Associez ce fichier `html` à une feuille de style `html2.css` de manière à obtenir les styles suivants.

- d. le texte sera en *sans-serif*, gris foncé, *c.f.* :  
<https://drafts.csswg.org/css-fonts-4>  
<https://drafts.csswg.org/css-color-4>
  - e. les titre et sous-titre seront centrés, encadrés en bleu foncé par un cadre occupant toute la largeur de la page et de fond bleu pâle, *c.f.* :  
<https://drafts.csswg.org/css-text-3/#justification>  
<https://drafts.csswg.org/css-box-3>  
<https://drafts.csswg.org/css-backgrounds-3>
  - f. dans le menu de navigation, les liens doivent apparaitre en rouge s'ils n'ont pas été cliqués, en noir sinon et en italique quand on les survole à la souris, *c.f.* :  
<https://html.spec.whatwg.org/multipage/semantics-other.html#pseudo-classes>
  - g. le titre de l'article en gras, juste un peu plus grand que le reste du texte ;
  - h. le nom de l'auteur aligné à droite en gras.
4. **Articles en colonnes.** Créer un fichier `html3.html` associée à feuille de style `css`. Ce fichier devra afficher :
- a. un **header** avec un titre, un sous-titre et un menu de navigation contenant des liens vers les pages des exercices précédents,
  - b. six articles affichés sur deux lignes et trois colonnes, contenant chacun un **header** avec un titre, une image, le texte commençant à la même hauteur que le bord supérieur de l'image en la contournant par la droite,
  - c. un **footer** contenant une liste de liens vers vos sites préférés.
- Faites en sorte que les bords des articles ne se touchent pas (`padding`).
5. **Mise en page.** Créer un fichier nommé `html4.html`, toujours associé 'a une feuille de style.
- a. La page contiendra un unique bloc principal (**main**) centré horizontalement, et occupant horizontalement 60% de la largeur de la page.
  - b. Toute la zone couverte par le bloc sera affichée avec fond gris clair. La zone non couverte par le bloc sera de fond bleu clair.
  - c. Le bloc contiendra quatre articles, tous de la même hauteur, disposés sur deux colonne. Ces articles seront sur fond blanc, bordé par du noir, avec un titre et un texte suffisamment long pour avoir une barre de scroll.
  - d. Faire en sorte que le texte ne touche pas les bords des articles, que les bords des articles ne se touchent pas entre eux, et ne touchent pas le cadre du bloc principal, en laissant ressortir son fond gris.

Les liens suivants vous seront utiles :

<https://drafts.csswg.org/css-sizing-3>

<https://drafts.csswg.org/css-position>

### III) Révisions de PHP

Dans les exercices qui suivent, les fichiers devront être placés dans le répertoire de votre serveur privé, `/var/www/private` sous Ubuntu. L'ouverture d'un fichier dans le navigateur se fait avec le préfixe `private.localhost/`.

6. Copier le fichier `html4.html` précédent en `php1.php`.

- a. Au dessus des articles dans la bande centrale, placer un article occupant toute la largeur de la bande. Le texte de l'article sera réduit à la ligne "Bonjour Le Monde!" affichée par php (echo).
- b. Dans la partie head de la page, écrire un script php définissant une variable \$couleur initialisée à "red". Immédiatement après ce script, définir une nouvelle classe de style se contentant de spécifiant comme couleur de fond la valeur courante de \$couleur. Modifier la partie body pour que la bande soit effectivement de cette couleur<sup>1</sup>.
- c. Modifier le script précédent pour y définir : une variable entière \$index initialisée à 0, un tableau \$palette de quatre noms de couleurs ("red", "blue"...). Après ces variables, initialisez \$couleur au contenu de \$palette de position \$index.
- d. Créer un fichier choix.html affichant une liste déroulant permettant de choisir une valeur parmi 0, 1, 2, 3, 4. La valeur choisie sera envoyée par la méthode GET à php1.php, et sera la valeur d'initialisation de la variable \$index avant l'affichage de la page. Voyez-vous ce qu'il faut modifier dans l'adresse du navigateur pour modifier cette valeur ? Que ce passe-t-il si la valeur entrée est incorrecte ?
- e. Modifier enfin les deux pages pour vous servir cette fois de la méthode POST, et notez les différences.

## IV) Révision de MySQL

Rappelons que l'accès à la base de données mabase sur votre serveur sql local peut se faire avec le code php suivant, en remplaçant "utilisateur" par votre nom :

```
try {
    $bdd = new PDO ('mysql:host=localhost; dbname=mabase',
                    'utilisateur', '');
} catch (PDOException $e) {
    print "Error!: " . $e -> getMessage () . "<br/>";
    die ();
}
```

La réponse à une requête requeteSQL peut être obtenue par :

```
$reponse = $bdd -> query ("requeteSQL");
```

La suite des éléments de la réponse peut être affichée par :

```
foreach ($reponse as $row) {
    echo $ligne['nom'] . "<br>";
};
$reponse -> closeCursor ();
```

La documentation complète de la classe PDO est ici :

<https://www.php.net/manual/en/class.pdo.php>.

### 7. On souhaite créer une base de données pour un restaurant végétarien.

---

1. Un élément peut être de plusieurs classes de style. Si plusieurs effectuent des modifications distinctes d'un même attribut, la modification prise en compte sera celle effectuée par la dernière classe ayant modifié cet attribut dans la suite des définitions de classes. Il faut donc que l'import de la feuille de style **html4.css** précède la définition de ce nouveau style.

- a. A l'aide de `mysql`, créer une base de données **restaurant** comprenant deux tables, **legumes** et **recettes**. La table **legumes** comprendra deux champs texte **nom** et **saison**. La table **recettes** aura trois champs **nom**, **cuisson** et **ingredient**, la cuisson étant le temps de cuisson en minutes, l'ingrédient étant un unique légume. Remplir cette base avec au moins quatre légumes et deux recettes.
  - b. Dans une page `sql1.php`, se connecter à la base de données et afficher :
    - tous les légumes ;
    - toutes les recettes ;
    - les recettes avec la saison de leur légume ;
    - le nombre de recettes.
8. On souhaite à présent étendre la base précédente pour y inclure des données sur les clients du restaurant.
- a. Créer une page `sql2.html` comprenant un formulaire de satisfaction client. Ce formulaire se composera, outre un bouton **envoyer**, d'un champ pour le login et deux menus déroulants permettant de donner des notes entre 0 et 2. La première note évaluera l'accueil du client, la deuxième la qualité des plats. Ce formulaire sera envoyé à la page `sql3.php`.
  - b. Dans `mysql`, créer dans la base **restaurant** deux tables supplémentaires pour stocker les résultats de l'enquête de satisfaction : une table **utilisateurs** comprenant un `id`, un `login`, et une table **notes** comprenant un `id`, et trois champs `notes`. Ajouter des clefs à ces tables, et faites en sortes que l'`id` ait un incrément automatique.
  - c. Dans `sql3.php`, lorsque des données sont envoyées par le formulaire de la page `sql2.php`, les entrer dans la base de données. Si un utilisateur existe déjà dans la table **utilisateur**, son mail (qui a pu changer) sera mis à jour. Sinon, il sera ajouté à cette table en recevant un nouvel identifiant. Dans tous les cas, ses notes seront ajoutées à la table des notes.
  - d. Dans la page `sql3.php`, après l'entrée des données, afficher :
    - le nombre d'utilisateurs différents,
    - la moyenne globale des notes de chaque catégorie.Mettre tout cela en forme grâce à du `css`.
  - e. Quand les données sont envoyées grâce au formulaire, avant de les insérer dans la base de données, vérifier qu'elles sont bien formées : pas de champ vide, l'adresse mail est bien de la forme : `xxxx @ xxxx.xxxx` (vérifier ceci à l'aide de `php` et d'expressions régulières).