

1 Explications et modalités de remise du projet

Ce projet est à faire en monôme ou en binôme. Les groupes de plus de 2 étudiants sont strictement interdits.

L'objectif de ce projet est de mettre en œuvre l'ensemble des aspects que vous avez vus dans ce module : XHTML, CSS et javascript.

Le thème de votre site web est laissé à votre libre choix. Si vous manquez d'idée, vous pouvez, par exemple, reprendre votre 3PE¹ et réaliser un site web le présentant. L'évaluation de votre projet ne tiendra pas compte du thème mais du respect (ou du non-respect) des contraintes techniques qui vous sont décrites par la suite.

On attend un site original, i.e. réalisé pour ce projet et non pas recopié ailleurs.

L'ensemble devra être remis sur UPdago sous forme d'un fichier archive au format *zip* ou *tar.gz* à l'exclusion de tout autre.

Le nom de l'archive sera **IMPÉRATIVEMENT** composé de vos noms de famille en minuscules séparés par un underscore, par exemple *dupont_dupond*, suivi des extensions classiques, *.tar.gz* ou *.zip* à l'exclusion de toute autre.

Le désarchivage devra créer, dans le dossier courant, un dossier s'appelant : *PROJET_nom1_nom2*. Dans l'exemple précédent, ce répertoire s'appellerait *PROJET_dupont_dupond*. Ces directives sont à respecter **SCRUPULEUSEMENT**.

Nous nous réservons le droit de convoquer un ou plusieurs groupes à un oral pour des précisions sur leur travail, ce qui peut conduire à ce que les membres d'un binôme n'aient pas la même note au final.

Ce projet donnera lieu à une note sur 20.

Pour mener à bien ce projet, vous devez vous répartir **intelligemment** le travail. Le projet permet, entre autres, de mettre en œuvre l'ensemble des compétences que vous êtes supposés avoir acquis à la fin de l'UE. Il est donc important que chacun fournisse du travail dans chacune des technologies : HTML / CSS / Javascript.

Pour ceux qui travailleront en binôme, faites clairement apparaître dans chacun de vos fichiers, le nom du contributeur principal via un commentaire.

2 Généralités

Les langages utilisés sont (sauf mention explicite du contraire) obligatoirement XHTML-1.0-strict, CSS-2.1 et javascript 5.1.

Vos fichiers doivent passer avec succès l'épreuve des validateurs du W3C, autrement dit sans erreur ni warning. Pour les fichiers CSS, une solution de vérification est d'utiliser l'extension "*Web Developer*" de Firefox. Pour les fichiers XHTML, une solution de vérification est d'utiliser l'extension "*HTML*

1. ou celui d'un des membres du binôme si vous ne l'avez pas fait ensemble

Validator” de Firefox. Mais dans les deux cas, une vérification finale sur le site du W3C est une bonne idée.

Il est interdit d'utiliser des programmes de génération de sites ou de reprendre un site existant.

Voici l'arborescence que doit respecter votre site Web :

```
PROJET_nom1_nom2/
|-- Contenu
|   |-- vos fichiers HTML
|-- Images
|   |-- vos fichiers image
|-- Styles
|   |-- vos fichiers CSS
|-- Javascript
|   |-- vos fichiers javascript
|-- index.html
```

3 HTML

a. TOUS les fichiers HTML

Dans le DOM de chaque fichier HTML, le corps du document a très exactement un seul fils, qui est une division.

Le reste de la page (titres, paragraphes, listes, images...) est inclus dans cette division².

b. Fichier index.html

Le fichier `index.html` est le point d'entrée sur votre site. Il doit contenir³ a minima (pas forcément dans cet ordre) :

- Un titre de niveau 1 qui est le titre de votre site. Ce titre doit être repris dans la barre de titre du navigateur ;
- Une division qui contient un texte présentant le contenu du site web ;
- Un menu qui permet d'accéder à l'ensemble des pages du site : il s'agit juste d'une liste dont chaque élément est un lien vers une des pages du site ;
- Une image qui sert de bandeau de bas de page, autrement dit une image plus longue que haute, qui lorsqu'on clique dessus permet d'envoyer un mail aux auteurs (vous donc).

c. Fichiers HTML dans Contenu

Il s'agit ici de la partie statique des fichiers HTML, vous vous occuperez de la partie dynamique lorsque vous arriverez au javascript.

- Le répertoire **Contenu** contient au moins 4 fichiers HTML, je les nommerai ici `page1.html`, `page2.html`, `page3.html`, et `page4.html`, mais vous pouvez les nommer comme bon vous semble, tant que leurs contenus respectifs respectent les contraintes indiquées ci-dessous. Si vous attribuez d'autres noms à vos fichiers, vous indiquerez en commentaire dans votre fichier de quelle page il s'agit.
- Chaque fichier HTML contient au moins :
 - un logo de votre choix
 - un titre de niveau 1
 - deux titres de niveau 2,
 - 2 ou 3 images (par exemple des flèches), qui permettent de passer à la page précédente (si elle existe), à la page suivante (si elle existe) et au fichier `index.html` (lui, il existe!),
 - le même bandeau de bas de page que dans le fichier `index.html` (qui permet également d'envoyer un mail aux auteurs du site).
 - La structure des 4 pages (autrement dit leur DOM) doit présenter des similitudes.

2. Cette structuration est nécessaire pour répondre de manière satisfaisante aux questions de la partie javascript

3. dans la division principale.

On pourrait imaginer, par exemple, que la division principale de chaque page contienne d'abord une division englobant les images de navigation, puis un titre de niveau 1, puis une division contenant elle-même un titre de niveau 2 (et autre chose propre à chaque page), puis une seconde division contenant un nouveau titre de niveau 2 (et autre chose propre à chaque page), puis le bandeau. Ceci n'est qu'un exemple et vous pouvez choisir de structurer vos pages différemment.

Dans chaque page, on trouvera également du texte faiblement mis en valeur et du texte fortement mis en valeur dans différents éléments (`<div>`, `<p>`, `<td>`, ``, `<option>`, `<label>`...).

- Le fichier `page1.html` contient essentiellement du texte. Dans cette page, devra apparaître au moins un bloc de citation et une (ou plusieurs) listes.
- Le fichier `page2.html` contient, outre du texte, des données chiffrées, structurées dans différents tableaux. A chaque tableau est associée une image (un graphique par exemple).
- Le fichier `page3.html` contient un formulaire décomposé en plusieurs parties. Il peut, par exemple, s'agir d'un quizz en lien avec votre site web. Le formulaire doit contenir plusieurs éléments de saisie de ligne de texte, un ensemble de cases à cocher, un ensemble de boutons radio, une liste déroulante à choix multiples, une zone de texte et des boutons de remise à zéro et d'envoi du formulaire. Pour tester (lorsque vous êtes dans les salles machines de l'Université), vous mettrez dans l'attribut *action* du formulaire la valeur `http://10.16.21.1/~salayran/traitement.php`.

Outre la validité du formulaire d'un point de vue HTML, chaque champ rempli par l'internaute doit pouvoir donner lieu à un traitement sur le serveur (autrement dit, les attributs "name" doivent être correctement renseignés).

- Le fichier `page4.html` contient des listes, et des listes de listes avec des liens sur des sites web. Cette page peut, par exemple, constituer la "*webographie*" de votre site.
- Si vous ressentez le besoin d'ajouter d'autres fichiers HTML pour compléter votre site, ceux-ci peuvent utiliser la norme HTML5 si vous le souhaitez. Vous devez le préciser.

4 Style CSS

Le choix des couleurs et de la plupart des autres valeurs de propriétés de style est laissé, sauf mention explicite du contraire, à votre appréciation.

Vous ne devez utiliser ici que les sélecteurs définis en CSS-2.1.

Dans vos fichiers de style doivent apparaître chaque type de sélecteurs au moins une fois, ainsi que des combinaisons de sélecteurs :

- sélecteurs d'éléments HTML
- sélecteurs via *id* et *class*
- sélecteurs d'enfants
- sélecteurs de descendants
- sélecteurs via des pseudo-classes
- sélecteurs via des pseudo-éléments
- sélecteurs via les attributs (autre que *id* et *class*)

Avant chaque jeu de règles, indiquez en commentaire la signification du sélecteur ou des sélecteurs au(x)quel(s) il s'applique et, pour chaque propriété, indiquez à côté en commentaire quelle est son utilité.

Par exemple,

```

/*****
* Première lettre du deuxième item de toute liste *
*****/

```

```
li:first-child + li:first-letter {
    color : red;                /* écrit en rouge */
}
```

- Les deux prochaines questions vous demandent de concevoir deux styles pour vos pages, un style principal et un style alternatif. Chaque style se décline en une version "écran" et une version "impression", chacune définie dans un fichier CSS propre. Si certaines règles sont partagées par les versions écran et impression d'un même style, vous devez les écrire une seule fois dans un troisième fichier de style que vous inclurez dans les deux fichiers de style demandés. Autrement dit, ici, vous avez *a minima* besoin de 4 fichiers de style et plus probablement 6, par exemple :

screen_mainstyle.css, print_mainstyle.css, common_mainstyle.css,
screen_altstyle.css, print_altstyle.css, common_altstyle.css.

- Toutes vos pages HTML sont liées à deux fichiers de style commun pour *l'affichage dans le navigateur* (un style principal et un style alternatif) qui gèrent chacun :
 - la position de votre logo : fixe en haut à droite de la page pour le style principal, au choix pour le style alternatif;
 - l'apparence des titres;
 - l'apparence des divisions et paragraphes (dans un des styles au moins, on distinguera les divisions des paragraphes en utilisant des indentations différentes pour les deux);
 - l'apparence des liens (visités ou pas);
 - la couleur de fond (autre que blanc) et la couleur d'écriture par défaut. Dans le style alternatif, on choisira des couleurs avec un fort contraste et une taille de police de caractères plus importante;
 - l'apparence et le positionnement des flèches de navigation
 - l'apparence et la position du bandeau de bas de page (dans un des styles, vous ajouterez une bordure et utiliserez les marges externes pour positionner votre bandeau)
 - l'apparence des textes faiblement et fortement mis en valeur (cette apparence doit varier selon l'endroit où se trouvent ces textes)
- Toutes vos pages HTML sont liées à un fichier de style commun pour *l'impression* qui gère :
 - la position de votre logo : fixe en haut à droite de chaque page imprimée pour le style principal, au choix pour le style alternatif;
 - la couleur d'écriture par défaut;
 - l'apparence des divisions et paragraphes;
 - l'apparence des titres;
 - l'apparence des liens;
 - l'apparence des tableaux (jouez sur les marges internes des cases - *padding*);
 - l'apparence des listes;
 - la disparition du bandeau de bas de page et des flèches de navigation.

Dans le style alternatif, toutes les couleurs utilisées sont des niveaux de gris et les caractères sont écrits plus gros que la normale.

- Chaque fichier est, en outre, associé pour *l'affichage dans le navigateur* à un fichier de style qui lui est propre (pas de style alternatif ici). Vous avez donc 5 autres fichiers de style à écrire. En cas de conflit avec le style commun à toutes les pages, ce doit être le style propre de la page qui l'emporte. Ces fichiers gèrent :
 - pour `index.html` :
 - un positionnement absolu pour l'ensemble des éléments présents;
 - l'apparence du titre de votre site;
 - l'apparence du menu, affiché horizontalement avec des puces qui sont des images;
 - l'apparence des liens lorsque la souris les survole.

- pour `page1.html` :
 - la présentation du texte sous forme de deux colonnes (en utilisant la propriété *display* et sans toucher à la position) ;
 - l'apparence d'un bloc de citation.
- pour `page2.html` :
 - le style de chaque tableau et de son image associée.
- pour `page3.html` :
 - le style de chaque partie du formulaire qui possède notamment une image en fond ;
 - l'apparence de chaque item du formulaire, apparence qui doit changer lorsqu'il a le focus, et lorsqu'on clique sur le label associé ;
 - l'apparence des titres qui doit être différente de celle des titres des autres pages.
- pour `page4.html` :
 - l'apparence des liens : les liens de la page ne doivent pas tous avoir la même apparence. Cela peut, par exemple, dépendre de l'importance que vous leur donnez.

5 Javascript

Attention : dans les questions suivantes, on n'utilisera jamais l'attribut *innerHTML*⁴.

- a. Modifiez votre formulaire pour que certaines vérifications de validité de données soient réalisées au fur et à mesure de la saisie par l'utilisateur du formulaire (par exemple lorsqu'un champ de saisie de texte perd le focus). En cas d'erreur de l'internaute, quelque chose lui indiquera la nature du problème : changement de style, message dans la page, message dans une fenêtre d'alerte... Vous pouvez également ajouter un bouton dans votre formulaire pour tester d'un coup la validité de l'ensemble du formulaire.
- b. Écrivez une fonction qui permet d'ajouter dynamiquement à la page `index.html`, à l'emplacement de votre choix, un paragraphe contenant l'heure de connexion de l'internaute à votre page sous la forme `hh :mm :ss` (avec éventuellement un petit message explicatif). Autrement dit, il s'agit d'une fonction qui modifie le DOM. Modifiez votre fichier `index.html` pour que cette fonction soit appelée à la fin du chargement de la page.

Pour cela vous utiliserez un objet de type *Date*. L'instruction `var d = new Date()` stocke dans la variable *d* un objet contenant toutes les informations sur la date et l'heure au moment de l'exécution de l'instruction. Ces informations peuvent être obtenues en utilisant sur *d* les méthodes des objets de type *Date*, par exemple :

- `d.getHours()` ;
- `d.getMinutes()` ;
- `d.getSeconds()`.

Pour construire la fonction répondant à cette question, vous aurez besoin d'écrire (au moins) une fonction javascript annexe :

- une fonction qui prend un entier d'au plus deux chiffres en paramètre et produit une représentation de ce nombre sur exactement deux chiffres en ajoutant, si nécessaire, un zéro dominant. Par exemple, si on lui fournit le nombre 9 elle donne la chaîne `"09"` et si on lui donne 12 elle fournit `"12"`.
- c. Écrivez les fonctions nécessaires afin que dans chaque page de votre site, un bouton soit dynamiquement ajouté (en haut à droite de la page) après le chargement de la page qui permette d'intervertir le style principal et le style alternatif pour le style commun à toutes les pages.
- Pour ce faire, on peut manipuler la balise `<link />` comme n'importe quelle autre balise. On peut notamment récupérer un objet javascript lui correspondant, après lui avoir affecté un

4. Si vous ne savez pas ce que c'est, pas de souci, vous n'avez justement pas à l'utiliser.

identifiant, via `document.getElementById(...)`. On peut ensuite modifier via javascript la valeur de ses attributs *rel* et *title*.

- d. Écrivez les fonctions nécessaires afin que dans chaque page de votre site, un menu apparaisse, après le chargement de la page, proposant des liens sur tous les titres h2 de la dite page. Ce menu aura une position fixe en haut à gauche.

Ce menu sera une division contenant une liste de liens vers les titres h2 de votre page.

Pour obtenir ce résultat, il faut :

- modifier le DOM en ajoutant une division au début de la page (avant la division principale) contenant une liste ;
- récupérer l'ensemble des titres `<h2>` de la page via `document.getElementsByTagName("h2")` ;
- pour chaque titre créer un item d'une liste contenant un lien pointant vers ce titre et l'ajouter à la liste incluse dans la division.
Attention, si vos titres h2 n'ont pas d'identifiant, vous allez devoir en ajouter (soit dynamiquement via votre script, soit directement dans le fichier HTML) ;
- adapter le style (toujours en javascript) pour que la division principale de la page occupe une largeur plus petite que celle de la page (par exemple 70 ou 80%) et que la division contenant le menu ait une position fixe, à la position souhaitée.