

Projet Doc Web

Ce travail sera évalué et comptera pour 1/2 de la note de contrôle continu de l'UE.

Objectifs :

Réalisation d'un mini site web d'au moins 5 pages (HTML) illustrant un **thème imposé**. Les cinq pages auront la même structure générale et partageront donc la même feuille de style.

Structure des pages

La structure des pages devra être composée de 4 blocs :

- un bloc de menu ;
- un bloc de titre ;
- un bloc de contenu ;
- un bloc d'auteurs.

Ces 4 blocs seront associés aux identifiants **#menu**, **#titre**, **#contenu** et **#auteurs**. Ils seront disposés selon votre choix, ci dessous quelques exemples (non exhaustifs) de dispositions possibles. Ces dispositions correspondent aux différents modèles de structure fournis dans l'archive *projetInfoDoc25.zip* disponible sur eureka. Ces modèles sont très simples, ils ne sont fournis qu'à titre d'exemples basiques et devront obligatoirement être **améliorés et personnalisés**.

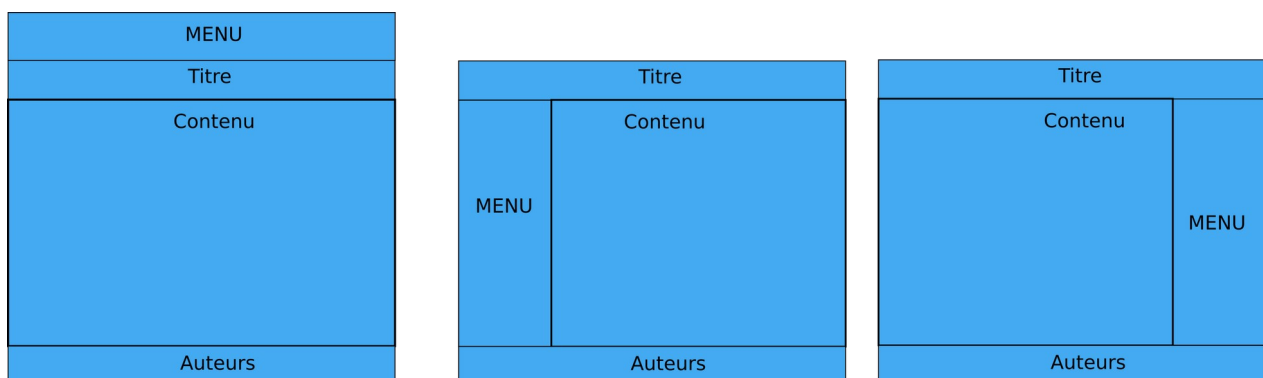


Illustration 1: Structures schématiques de pages

Contenu des blocs

- Le bloc de titre indiquera le nom du site et le titre de la page.
- Le bloc de contenu détaillera les informations de la page courante et sera organisé en sections introduites par des titres. Ces sections seront accessibles **directement** à l'aide de liens internes sur les titres introductifs.
- Le bloc de menu affichera des liens permettant d'accéder à toutes les pages du site. Pour chaque page du site, des liens hypertextes permettront d'accéder **directement** à chaque section.
- Le bloc d'auteurs permet d'identifier les auteurs du projet.

Le mini site sera constitué d'une page d'accueil définie dans un fichier **index.html**, elle sera associée à au moins 3 pages de contenu et à une page recensant les sources informationnelles utilisées.

Architecture du site

L'ensemble de vos fichiers devra être structuré de la façon suivante (en adaptant le nommage des fichiers et en veillant à nommer vos répertoires et fichiers **sans espace et sans caractères diacritiques**) :

- ProjetDocWeb
 - Markdown
 - 📄 index.md
 - 📄 fichier1.md
 - 📄 fichier2.md
 - 📄 ../..
 - HTML
 - 📄 index.html
 - 📄 fichier1.html
 - 📄 fichier2.html
 - 📄 ../..
 - Images
 - 📄 image1.jpg
 - 📄 image2.png
 - 📄 ../..
 - CSS
 - 📄 nomÉtudiant.css

Thème du site

Le site produit portera sur **une série (TV, streaming, animée, live action, ...)**

Planification du travail

Étape 1 : Prototypage de la structure des pages HTML, le contenu des blocs de menu, de titre, de contenu et d’auteurs seront alimentés à l’aide de Markdown et transformé en HTML avec pandoc ; pandoc permettra également d’intégrer un squelette de document XHTML ; le contenu sera alimenté avec du texte de type « Lorem Ipsum » ;

Étape 2 : réalisation de la feuille de styles CSS et association au prototype produit lors de l’étape 1 ;

Étape 3 : alimentation du site avec le véritable contenu et finalisation du projet. Il n’est pas nécessaire de rédiger le contenu mais toutes les sources utilisées pour alimenter le site devront être citées dans la page dédiée à cet effet.

Attendus (en binôme)

Une archive compressée au format **ZIP** contenant l’ensemble des fichiers Markdown, HTML et CSS devra être déposée sur EUREKA **avant 18:00 le 09 mai 2024**.

Un seul dépôt par binôme est nécessaire.

Code Pandoc

Template et sous-templates de la structure de page utilisé dans le site

Code Markdown

Les prototypes des pages réalisés en Markdown devront être fournis, un fichier (format txt) indiquera les commandes pandoc utilisées pour générer les fichiers HTML du site (et éventuellement des sous-templates).

Code HTML Final

Le code HTML produit avec Pandoc devra être valide au format **xHTML1.0 strict**. Il devra obligatoirement intégrer des images, un ou plusieurs tableaux de données et des listes.

Code CSS

Le code CSS devra mettre en œuvre des sélecteurs de types, des sélecteurs de classe, des sélecteurs d’identifiant ainsi que des sélecteurs contextuels. Les propriétés de polices, de textes, de listes, de tableaux et de boîtes devront être utilisées. Les modèles fournis en exemple peuvent être réutilisés moyennant une réelle personnalisation. L’utilisation des box-layout ou flex-layout est un plus.