05A - Interfaces Web

**Normes d’intégration web**

[1 - Structure de dossiers et fichiers](#_s716nsk786im)

[1.1 - Nomenclature des dossiers et fichiers](#_to1qvya4vipl)

[1.2 - Structure de site web](#_6ciyxsidhf0h)

[1.3 - Chemins relatifs](#_an3snowahofu)

[2 - Code HTML et CSS](#_w3jp8j69ui8v)

[2.1 - Nomenclature des attributs class et id](#_uqqbopirrfml)

[2.2 - Sémantique HTML](#_o65mnd72s6m7)

[Titres](#_am1uopizhzx6)

[Listes](#_gyxtowblmnru)

[Balises d’emphase](#_2ghudnwv6p6e)

[Balises génériques](#_qdwgojapn7x6)

[Balises de structure de contenus](#_yl0skalivokq)

[Métadonnées](#_bv0qc67ilvxr)

[2.3 - Feuille de style (style.css)](#_jz9aaajla0o0)

[2.4 - Validation du code](#_gout0obherac)

[3 - Intégration d’images](#_na7epcb6fiaj)

[3.1 - Formats d’images](#_n9iuo1sbbcvd)

[3.2 - Optimisation de la taille](#_qkesvnkdxzsv)

## 1 - Structure de dossiers et fichiers

### 1.1 - Nomenclature des dossiers et fichiers

Le nom des **dossiers** et **fichiers** d’un site web doit respecter les règles suivantes.

* Le nom doit être **significatif** et en **français.**
* Le nom doit être composé uniquement de **lettres minuscules**.
* Le nom **ne doit pas** contenir d’**accent** ou autres **caractères spéciaux**.
* Le nom peut contenir des **chiffres,** mais ne doit **pas débuter** par un chiffre.
* Si le nom comporte plus d’un mot, les **mots** doivent être **séparés par un trait d’union** “-”.

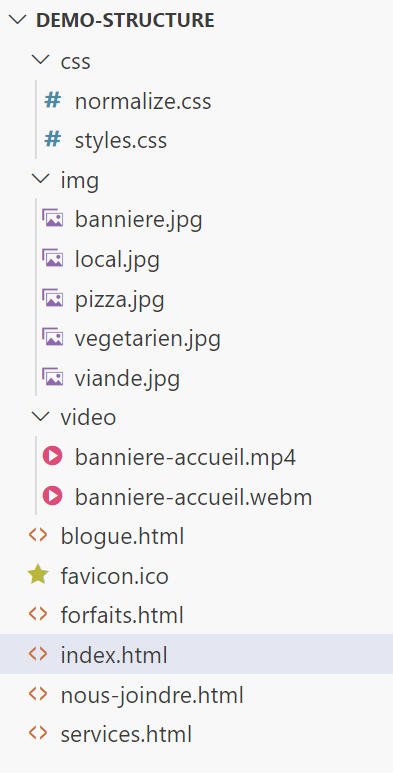
### 1.2 - Structure de site web

La structure des dossiers et fichiers doit respecter les règles suivantes.

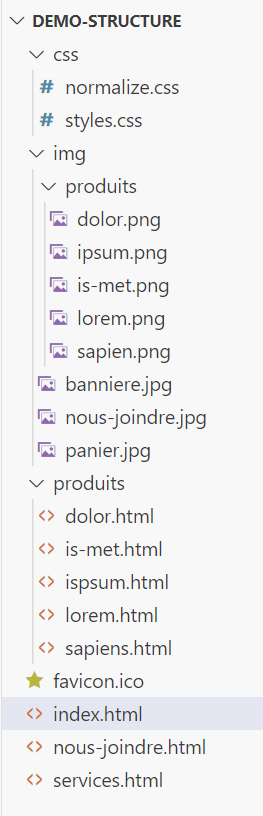
* Le nom du **dossier du site web** doit être **significatif**. **Important :** Ne pas utiliser le nom “nom-du-projet” du gabarit de base comme nom de dossier.
* Le fichier de page de démarrage (la page d’accueil du site) doit toujours être nommé **index.html** et se trouver à la racine du dossier de projet.
* Les **autres** fichiers de **pages du site** peuvent aussi se trouver à la racine du dossier du projet. Par contre, si le site comporte des pages de même nature (exemple : des produits), il est préférable de regrouper celles-ci dans un **sous-dossier.**
* Les **feuilles de style** se trouvent dans un **sous-dossier** nommé “**css**”.
* Les fichiers d’**images** se trouvent dans un **sous-dossier** nommé “**img**”.
* Les **fichiers audio** se trouvent dans un **sous-dossier** nommé “**audio**”.
* Les **fichiers vidéo** se trouvent dans un **sous-dossier** nommé “**video**”.
* Le site comporte un **favicone** nommé **favicon.ico** placé à la racine du site.
* Le **dossier** ne doit pas inclure de **fichiers non utilisés** (exemple : des images).

**Exemples de structure de site web**

Structure simple (site de quelques page)



Structure plus complexe (plusieurs pages)



### 1.3 - Chemins relatifs

Les **chemins** entre les **fichiers** du site (image, liens internes, feuille de style css, etc.) doivent toujours être **relatifs** et écrits sous la forme suivante (exemples) :

* <img src="**img/web.png**" alt="image web">
* body{

background-image: url(**../img/web.png**); } *(accès à un dossier supérieur)*

Formes à **éviter** (peut avoir été causé par l’utilisation d’une extension dans VS Code) :

* <img src="/**img/web.png**" alt="image web">
* <img src="./**img/web.png**" alt="image web">

## 2 - Code HTML et CSS

### 2.1 - Nomenclature des attributs **class** et **id**

Le nom donné à une valeur d’attribut **class** ou **id** dans le code HTML doit respecter les règles suivantes.

* Le nom doit être **significatif** et en **français,** sauf exceptions pour quelques concepts vus dans le cadre du cours. Exemple : *container*.
* Le nom doit être composé uniquement de **lettres minuscules**.
* Le nom **ne doit pas** contenir d’**accent** ou autres **caractères spéciaux**.
* Le nom peut contenir des **chiffres,** mais ne doit pas débuter par un chiffre.
* Si le nom comporte plus d’un mot, les **mots** doivent être **séparés par un trait d’union**“-”. **Note** : les abréviations de mots sont permises.
* Un nom de **class** doit être **général** car il doit pouvoir être utilisé dans différentes pages et différents contextes. Par exemple, une classe de style appliquée sur un bouton “Ajouter au panier” ne devrait pas être nommée “btn-ajout-panier”. On devrait plutôt utiliser un nom “bouton” ou “btn-action”, car ce style pourrait éventuellement être utilisé dans d’autres contextes. Exemples :



***Astuce d’optimisation*** *: Il est possible d’utiliser plusieurs valeurs dans l’attribut* ***class*** *d’un même élément, afin de garder les classes le plus réutilisable possible.*

* Un nom de **id** doit être **spécifique** car la valeur ne peut pas être utilisée plus d’une fois dans une page. Exemple : <nav id="nav-principale">.
* Un **id** doit être appliqué aux éléments suivants, dans le but de les identifier clairement (pas nécessairement pour leur appliquer un style particulier) :
  + <nav>
  + <section>
  + <form>

Quelques **exemples** valides de valeurs de class et id :

* **class**="btn"
* **class**="btn-action"
* **class**="sect-fonce"
* **class**="boite-contenu fond-fonce" (avec 2 valeurs séparées par un espace)
* **id**="sect-a-propos"
* **id**="nav-principale"
* **id**="nav-pagination"
* **id**="banniere-accueil"
* **id**="btn-ajout-panier"

### 

### 2.2 - Sémantique HTML

#### Titres

* La balise <h1> doit être utilisée **une seule fois** par page (sauf dans de très rares exceptions qui ne devraient pas être abordées dans le cadre du cours).
* Une page web doit **obligatoirement** contenir un **titre principa**l (balise <h1>)
* Les balises de **titres secondaires** <h2><h3>... <h6> peuvent être répétées et doivent être utilisées pour **hiérarchiser les contenus** d’une page.
* Un niveau de titre secondaire ne peut pas être “sauté” dans une page. Par exemple, on ne devrait pas retrouvé dans une page le niveau <h4> s’il n’y aucun titre <h3>

#### Listes

* La balise <ul> doit être utilisée pour une liste d'éléments sans ordre particulier.
* La balise <ol> doit être utilisée pour une liste d'éléments ordonnée. Exemple : une liste d’étapes de recette.
* Toutes formes de liste doit utiliser les balises <ul> avec <li> (ou <ol> avec <li>). L’utilisation de ces balises ne doit pas être limitée seulement à une liste à puces standard. Quelques exemples : Une liste de liens dans une navigation, une liste de produits dans une boutique en ligne, une liste de clients, etc.
* Une balise <ul> ou <ol> ne doit pas avoir d’autres éléments que <li> comme enfants directs.
* Une balise <ul> ou <ol> ne peut pas être un enfant d’une balise <p>.

#### Balises d’emphase

* Un paragraphe de quelques phrases devrait utiliser au moins une **balise d’emphase** sur un ou des mots.
* La balise <strong> est utilisée pour mettre de l’emphase sur un **mot important**, en **lien direct** avec le **sujet d’un paragraphe** ou d’une **section** de contenu.
* La balise <em> est utilisée pour mettre de l’emphase sur un **mot important** d’une **phrase**.

#### Balises génériques

* La balise générique <div> (sans valeur sémantique) doit seulement servir à regrouper le contenu afin qu'il puisse être facilement stylé à l'aide de l'attribut **class** (ou **id**). Il faut, autant que possible, limiter son utilisation. [Référence](https://developer.mozilla.org/fr/docs/Web/HTML/Element/div)
* L’usage de la balise générique <span> (sans valeur sémantique) doit être utilisée seulement pour appliquer une **distinction graphique** dans une **portion de texte**.

#### Balises de structure de contenus

* Balises de structure **obligatoires** :
  + **<header>** : une page doit avoir au moins une balise <header> pour l’en-tête de celle-ci. *Note : Cette balise doit aussi être utilisée dans d’autres contextes. Par exemple, une entête d’article de blogue.* [Référence](https://developer.mozilla.org/fr/docs/Web/HTML/Element/header)
  + **<nav>** : une page doit avoir au moins une balise <nav> pour la navigation principale du site. *Note : Cette balise doit aussi être utilisée dans d’autres contextes. Par exemple, une navigation secondaire, une pagination, un fil d’ariane, etc.* [Référence](https://developer.mozilla.org/fr/docs/Web/HTML/Element/nav)
  + **<main>** : une page doit avoir une seule balise <main> délimitant le contenu principal distinctif d’une page.
  + **<footer>** : une page doit avoir au moins une balise <footer> pour les informations de bas d’une page. *Note : Cette balise doit aussi être utilisée dans d’autres contextes. Par exemple, les informations sur l’auteur, dans le bas d’un article de blogue.* [Référence](https://developer.mozilla.org/fr/docs/Web/HTML/Element/footer)
* Autres balises de structure à utiliser **lorsque justifié** :
  + **<section>** : l’élément <section> est utilisé pour regrouper des éléments appartenant à un même thème. Une section doit généralement débuter par un titre secondaire (<h2> à <h6>). Exemple : différentes sections dans le contenu principal <main>, différentes sections dans le bas de page <footer>, regroupement des informations de contact, différentes section d’un article, etc. [Référence](https://developer.mozilla.org/fr/docs/Web/HTML/Element/section)
  + **<article>** : une balise <article> permet de définir un contenu dit “autonome” dans une page. C’est-à-dire un groupe d’éléments qui pourrait être utilisé dans un autre contexte que celui du site. Exemples: un article de journal, une publication dans un blogue, une fiche de produit, une fiche de recette, etc. [Référence](https://developer.mozilla.org/fr/docs/Web/HTML/Element/article)
  + **<address>** : une balise <address> est utilisée pour regrouper les informations de contact pour une personne, un groupe de personnes ou une organisation. [Référence](https://developer.mozilla.org/fr/docs/Web/HTML/Element/address)
  + **<aside>** : partie d'un document dont le contenu n'a pas un rapport direct avec le contenu principal du document. Ces contenus sont fréquemment présents sous forme de barre latérale dans la page. [Référence](https://developer.mozilla.org/fr/docs/Web/HTML/Element/aside)
  + **<time>** : L'élément HTML <time> permet de représenter une donnée temporelle: une heure, une date ou une durée ([référence](https://developer.mozilla.org/fr/docs/Web/HTML/Element/time)). L’utilisation de son attribut datetime permet l’indexation de cette information par les moteurs de recherche (voir les [valeurs valides](https://developer.mozilla.org/fr/docs/Web/HTML/Element/time#valeurs_valides)).

#### Métadonnées

Les métadonnées suivantes sont obligatoires dans chaque page du site :

* La **langue** de la page : <html lang="fr">
* L’encodage de **caractères** UTF-8 : <meta charset="utf-8">
* Le **titre** de la page, sous la forme suivante :

<title>Nom de la page - Nom du site</title>

* La **description** de la page :

<meta name="description" content="Description de la page">

* La zone de d’affichage ***viewport*** :

<meta name="viewport" content="width=device-width, initial-scale=1">

### 2.3 - Feuille de style (style.css)

* **Une seule** feuille de **styles personnalisées** (style.css) doit être créée par projet.
* Toutes les propriétés CSS appliquées doivent se trouver dans la feuille de style. Vous ne pouvez pas utiliser l’attribut style dans les fichiers HTML.

Exemple à **ne pas faire** : <h1 style="color: red;">Nous joindre</h1>

* Le **nom d’une classe** de style (**class**) et la règle de style qui s’y rattache doit être pensée dans un optique de réutilisation dans un autre contexte (voir aussi la section [2.1 - Nomenclature des attributs class et id](#_uqqbopirrfml) à ce sujet).
* Les règles de styles doivent être **regroupées** par **catégorie** à l’aide de commentaires. Ces **catégories** doivent être **classées** dans un **ordre logique** : les **règles** plus **générales** se trouventen **début** de la feuille de style et les règles plus **spécifiques** vers la **fin**. Vous pouvez utiliser les **catégories proposées** dans le fichier **style.css** du **gabarit de site** (nom-du-projet.zip) distribué dans le cadre du cours.

### 2.4 - Validation du code

Le code doit être **validé** à l’aide des outils suivants.

* Validation du code **HTML** : <https://validator.w3.org/#validate_by_input>
* Validation du code **CSS** : <https://jigsaw.w3.org/css-validator/#validate_by_input>

**Note** :

* Le code ne doit **absolument** pas avoir d’erreur.
* Le code ne doit **idéalement** plus avoir d’avertissement. En d’autres termes, si l’avertissement peut être corrigé facilement, vous devez le faire. En cas de doute, contactez le professeur.

## 3 - Intégration d’images

### 3.1 - Formats d’images

Seuls les **formats d’images** suivants peuvent être utilisés. *Note : Se référer au* ***chapitre concernant les images*** *pour connaître les particularités des formats* :

* jpg
* png
* gif
* svg
* avif
* webp

**Nomenclature** : La nomenclature des fichiers d’images doit respecter les règles listées à la section [1.1 - Nomenclature des fichiers](#_to1qvya4vipl) de ce document.

### 3.2 - Optimisation de la taille

* La **taille des images** doit être optimisée afin de minimiser le poids de chargement de celles-ci. La **largeur** d’une image ne doit **pas excéder le double** de la largeur qu’elle occupe dans la page.