

1 Contexte du projet

Ce projet est à faire en **binôme**. Votre partenaire n'a pas besoin d'être dans votre groupe de TD même si cela peut s'avérer plus pratique pour le travail collaboratif. Par ailleurs, les étudiants des groupes 4 et 5 (NSI) auront un supplément à faire pour la partie script. Tout binôme qui compte au moins un étudiant d'un de ces groupes aura cette partie supplémentaire à effectuer.

Pour chacun des fichiers produits, vous devez indiquer le nom du contributeur principal dans un commentaire. Si vous estimez que le travail sur le fichier a été équitablement réparti, vous indiquez vos deux noms.

Un **oral** sera organisé à la fin du semestre pour que vous nous présentiez votre projet.

Le travail est collectif mais la notation individuelle : autrement dit, deux membres d'un binôme peuvent avoir des notes différentes.

Ce projet va servir de fil rouge tout au long de l'UE et vous permettra de mettre en œuvre l'ensemble des aspects traités dans ce module : XML, XHTML, CSS et typescript/javascript.

Objectif « pratique » du projet : concevoir et réaliser un site web autour d'un catalogue présentant une « collection » de votre choix ¹.

Le thème de votre collection doit être un sous-thème de l'un des thèmes suivants :

- des œuvres (livres, films, tableaux, musiques, ...)
- des personnes / personnages (artistes, scientifiques, héros de roman, sportifs, ...)
- des jeux (Jeux de société, jeux vidéo, ...)
- des lieux (Pays, villes, lieux remarquables, ...)

Vous pouvez, par exemple, choisir de présenter vos joueurs de football préférés, ou les films vainqueurs aux Oscars les dernières années, les jeux auxquels vous jouez sur votre console, etc.

Plusieurs groupes peuvent choisir le même thème, mais chacun doit rendre un travail original ².

Un exemple est partiellement montré sur UPdago. Il s'agit d'une « collection » de lauréats du prix Turing.

Différentes aides pourront être ajoutées au fur et à mesure de vos questions. Donc n'hésitez pas à en poser ! Cela aidera tout le monde.

2 Originalité du travail vs fraude

Évidemment, le travail rendu doit être original. Autrement dit, il ne doit pas être similaire au travail d'un autre groupe (de cette année ou d'une année précédente) ni à un site web existant. Pour les redoublants, il vous est demandé de faire un nouveau projet, ne serait-ce que parce que le projet est une aide pour acquérir les compétences pour valider l'UE. Accessoirement, l'énoncé diffère partiellement de celui de l'an passé.

En cas de copie, il s'agit de fraude voire de plagiat et vous risquez le conseil de discipline avec toutes les conséquences possibles. À noter que le plagiat est tout aussi illégal dans le milieu professionnel et les conséquences également peu agréables.

1. Évidemment, la collection choisie doit rester dans les limites de la légalité et de l'éthique.

2. On y revient dans un instant.

La limite est parfois ténue entre l'inspiration et le plagiat. Si vous avez un doute, n'hésitez pas à poser la question.

La réalisation de ce projet comporte deux aspects : la technique « informatique » (structuration des pages du site, styles, scripts...) et le contenu (texte et images).

Pour le deuxième, vous avez le droit de reprendre des textes et des photos récupérés sur le web (ou ailleurs) MAIS vous devez absolument citer vos sources et bien les référencer.

Notamment, vous devez être conscients que, si vous récupérez des images sur le web, elles sont soumises à droit d'auteur et ont probablement été publiées sous une licence ou une autre. Comme le site que vous fournirez n'a pas vocation à être publié sur Internet, la seule chose que nous demandons ici est que pour chaque image qui ne vous appartient pas, une info bulle³ indique l'url du site sur lequel vous avez pris l'image en question. Si vous avez inséré une image que vous avez réalisée vous-même, indiquez-le également dans l'info-bulle.

Une page de votre site sera également dédiée à référencer l'ensemble de vos sources dont les images.

D'autre part, les images présentes sur un site web, lorsqu'elles ne sont destinées qu'à l'affichage sur l'écran, n'ont pas besoin d'être dans une résolution très fine. Il existe divers outils gratuits pour redimensionner des images. N'hésitez pas à demander. Votre site doit avoir un poids raisonnable.

3 Travail attendu

Dans cette UE, vous allez découvrir les technologies de bases pour le Web côté client ((X)HTML, CSS, Typescript/Javascript).

Il existe des bibliothèques logicielles et des outils qui permettent de simplifier l'utilisation de ces technologies et la création de sites web. Vous ne devez pas les utiliser ici. Le but est de comprendre comment tout cela fonctionne, non de produire de manière efficace un site professionnel. Les outils et bibliothèques évoluent tout le temps, les principes fondamentaux, eux, ne changent pas.

Un site web est classiquement composé de plusieurs fichiers de diverses natures (HTML, CSS, Typescript / Javascript...) rangés dans des dossiers (ou répertoires).

Attention, il ne doit pas y avoir d'accents dans les noms de vos fichiers.

Voici l'arborescence de répertoires / dossiers et fichiers que doit respecter votre site Web. Une archive vous est fournie sur UPdago qui contient cette arborescence. Commencez par télécharger l'archive en question et extrayez-en l'arborescence de répertoires.

Un fichier `index.html` est présent dans chacun des répertoires. Vous ne devez pas les modifier. Les fichiers HTML que vous créerez seront tous suffixés par `.xhtml` pour que lorsque vous les passerez au validateur <https://validator.w3.org/nu/>, leur syntaxe soit vérifiée avec l'analyseur XML.

Pensez à renommer le dossier racine avec vos noms.

```
SITE_nom1_nom2/
|-- Content
|   |-- Collection
|       \-- vos fichiers HTML décrivant la collection
|   |-- Forms
|       \-- vos fichiers HTML contenant des formulaires
|   |-- Miscellaneous
|       \-- autres fichiers HTML (par exemple, votre bibliographie)
|-- Images
|   \-- vos fichiers image
|-- Code
```

3. attribut *title* de la balise ``

```
| |-- Javascript
|     \-- vos fichiers javascript
| |-- Typescript
|     \-- vos fichiers typescript
|-- Styles
|     \-- vos fichiers CSS
|-- XML
|     \-- votre DTD
|     \-- vos fichiers XML
\-- index.html
\-- README.txt : toutes les informations que vous jugerez utiles
```

Si vous en ressentez le besoin, vous pouvez créer des sous-dossiers supplémentaires.

Tous vos fichiers doivent passer avec succès l'épreuve du validateur (HTML⁴ et CSS⁵).

Un fichier `autoeval.xhtml` vous est fourni. Téléchargez-le et mettez-le dans le dossier `SITE_nom1_nom2/Content/Miscellaneous`. Récupérez également le fichier `autoeval.css`, rangez-le dans le répertoire `SITE_nom1_nom2/Styles`. Outre le style précédent, appliquez également à la page `autoeval.xhtml`, le style général de votre site.

Comme montré en cours, ce fichier contient un tableau HTML vous permettant de réaliser une auto-évaluation de votre travail. Chaque ligne correspond à un des éléments demandés dans le projet. L'intitulé de l'objectif est rappelé dans la deuxième case de la ligne. La troisième case contient un curseur représentant l'avancement de votre projet sur ce point-là. Vous devez renseigner **directement dans le code HTML** la valeur de l'attribut `value` de l'élément `input` correspondant à ce curseur avec une valeur entre 0 et 100 représentant le taux d'achèvement de la tâche. Idéalement vous devriez être à 100% partout à la fin du projet mais si vous n'avez pas fini, ne le cachez pas et indiquez un taux qui vous paraît réaliste. La dernière case de la ligne vous permet de saisir un commentaire (toujours directement dans le fichier HTML) sur votre travail. Notamment si le travail n'est pas abouti, expliquez à cet endroit ce qui manque et les difficultés rencontrées.

4 Planning prévisionnel⁶

Vous devez rendre votre travail régulièrement (cf calendrier ci-dessous).

À noter que, jusqu'à la fin de votre projet, vous pouvez toujours modifier / compléter le travail déjà effectué. Il ne sera noté qu'à la fin.

Le respect des livraisons intermédiaires entrera néanmoins en compte dans la notation⁷.

Lorsque vous travaillez en binôme : un seul des deux remet le travail proprement dit, le second remet juste un fichier en rappelant le nom de son binôme.

1. Choix du thème et constitution des groupes sur UPdago : lundi 17 février 2025, 19h00
2. Remise sur UPdago de la structure XML des fiches (cf section 5) la DTD, un exemple de fiche par étudiant, l'arbre de chacune des deux fiches restreint aux nœuds éléments : vendredi 21 février 2025, 19h00

4. <https://validator.w3.org/nu>

5. <http://jigsaw.w3.org/css-validator/>

6. Vous aurez noté le mot « prévisionnel » : si nécessaire, le planning peut être modifié pour s'adapter à votre avancement.

7. Il peut arriver qu'il vous soit difficile de respecter une date de livraison pour une raison ou une autre, la bonne manière de procéder est alors de me contacter AVANT la date limite pour m'informer de vos difficultés. Il est alors possible (mais pas certain) que vous ayez droit à un délai supplémentaire. Si vous ne demandez pas, vous n'aurez pas de délai supplémentaire et vous serez pénalisé.

3. Remise d'une archive sur UPdago contenant l'ensemble du site statique (cf section 6 sans ce qui est demandé 6.4) avec son style (cf section 7) (sans les formulaires) : **vendredi 28 mars 2025, 19h00**
4. Remise d'une archive sur UPdago contenant l'ensemble du site statique avec les formulaires : **vendredi 18 avril 2025, 19h00**
5. Remise du site final (avec les formulaires et les interactions javascript) et la grille d'auto-évaluation complétée : **lundi 12 mai 2025, 19h00**

Réalisation de l'archive :

Vous pouvez rendre une archive au format tar.gz, tar.bz2 ou zip à l'exclusion de toute 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 .tar.gz, .tar.bz2 ou .zip

Le désarchivage devra créer, dans le dossier courant, un dossier s'appelant : SITE_nom1_nom2. Dans l'exemple précédent, ce répertoire s'appellerait SITE_dupont_dupond.

Ces directives sont à respecter SCRUPULEUSEMENT.

Si vous travaillez sous windows, réalisez un clic droit sur le dossier à compresser, vous trouverez dans le menu contextuel ce qu'il faut pour réaliser une archive au format zip.

Si vous travaillez sous Linux (ou sous MacOS), ouvrez un terminal et utilisez la commande `tar cvzf nom1_nom2.tar.gz SITE_nom1_nom2` ou bien `tar cvjf nom1_nom2.tar.bz2 SITE_nom1_nom2`.

5 XML

Il s'agit ici de créer une description XML des éléments de votre collection et la DTD correspondante.

Pour chaque élément de votre collection, vous devez lister les informations pertinentes pour les définir puis structurer ces informations en différents niveaux.

Dans l'exemple de la recette de cuisine vu en cours, la recette contient une liste d'ingrédients qui contient elle-même un ou plusieurs ingrédients. La recette correspond au premier niveau, la liste d'ingrédients est définie au deuxième niveau, chacun des ingrédients au troisième niveau.

Quelques contraintes :

- Chaque fiche doit posséder
 - au moins 3 niveaux autrement dit la balise racine doit avoir des petits-enfants,
 - une suite d'éléments obligatoires (éventuellement présents en plusieurs exemplaires) avec un ordre prédéfini,
 - un ou plusieurs éléments facultatifs,
 - des éléments contenant d'autres éléments,
- Plusieurs éléments doivent disposer d'un ou plusieurs attributs obligatoires ou non. Il doit y avoir au moins un attribut de type ID dans la DTD. Il doit y avoir aussi des attributs pour lesquels la liste des valeurs possibles est contrainte.

FICHER A RENDRE SUR UPDAGO : pour le premier jalon, le répertoire XML de votre site web doit contenir la DTD, un exemple d'une fiche de votre collection au format XML **par étudiant** et pour chaque fichier XML, le dessin de l'arbre correspondant restreint aux nœuds élément.

Les fichiers XML doivent évidemment être conformes à votre DTD.

6 HTML

Vos fichiers HTML doivent absolument respecter la norme selon laquelle ils ont été écrits. Pour mémoire, c'est la ligne de votre fichier qui commence par "`<!DOCTYPE`" qui indique quelle est la

norme HTML que vous avez choisi d'utiliser dans votre fichier.

Ici vous utiliserez la dernière version de la norme (décrite sur <https://html.spec.whatwg.org/>).

Vous utiliserez pour tous vos fichiers HTML l'extension `.xhtml`.

Vous devez valider la conformité de votre fichier avec le validateur : <https://validator.w3.org/nu/>.

Le fichier `document_minimal.xhtml` disponible sur UPdago vous fournit un squelette de page HTML avec le DOCTYPE adéquat ainsi que quelques éléments obligatoires pour que tout se passe bien.

Dans l'archive qui vous est fournie avec l'architecture du site, sont présents plusieurs fichiers `index.html` : vous ne devez pas modifier ces fichiers.

6.1 Le fichier `index.xhtml`

Vous devez écrire un fichier `index.xhtml` pour servir de point d'entrée sur votre site. Il doit se situer à la racine de votre site. Il contient un titre de niveau 1, un (petit) texte de présentation de votre site, une ou plusieurs images, des liens vers une ou plusieurs page(s) de votre site, un lien (texte ou image) permettant d'envoyer un email aux auteurs du site (à vous donc).

6.2 Fiches de la collection

Dans la partie XML, vous avez structuré les informations des éléments de votre collection. Vous allez maintenant créer un site web, autrement dit des fichiers HTML, pour présenter votre collection. Chaque élément de votre collection sera décrit par un texte structuré dans un fichier HTML. Les documents HTML présentant les éléments de votre collection auront tous la même structure, qui s'inspirera de votre structure XML.

Par exemple, pour la collection des lauréats du "prix Turing", chaque fiche XML contient un élément `<personaldata>` qui a (au moins) un attribut indiquant s'il s'agit d'un homme ou d'une femme, et au moins deux éléments enfant qui sont un nom `<name>` et un prénom `<firstname>` du lauréat. On peut, à partir de là, décider que chacune des pages HTML présentant une fiche débutera par un titre de niveau 1 contenant "M" ou "Mme" selon qu'il s'agit d'un homme ou d'une femme suivi du prénom et du nom. On peut ainsi établir une correspondance entre les informations structurées dans le XML et leur présentation sous forme de texte structuré (sous forme de titres, paragraphes, divisions...) dans la page (X)HTML.

Chaque étudiant doit écrire au moins 2 fiches. Autrement dit, votre site aura au moins 4 fiches.

Attention, encore une fois, les fiches doivent toutes respecter la même structure (ce sera indispensable pour attribuer les styles ultérieurement).

Les fiches doivent également contenir au moins une image, et des liens vers d'autres fiches (par exemple, les fiches précédente et suivante), ainsi qu'un lien vers la page d'accueil de votre site `index.xhtml`.

Pour réaliser ces pages HTML vous avez deux possibilités équivalentes en terme de notation :

- Vous écrivez directement les pages HTML (2 par étudiant) de sorte qu'elles respectent toutes la même organisation que vous aurez décidée,
- Une fois toutes les fiches écrites au format XML, vous écrivez une feuille de transformation XSLT et vous l'utilisez pour transformer vos fiches XML en pages HTML. Si vous choisissez cette manière de procéder, vous devez rendre la feuille de style XSLT.

6.3 Autres informations

Le répertoire `Miscellaneous` contient au moins deux fichiers HTML.

- a. Le fichier `authors.xhtml` dresse un portrait chinois des auteurs. Vous n'êtes évidemment pas obligés de donner des informations véridiques vous concernant. Les informations sont affichées

dans un tableau : une colonne par auteur du site, dont l'entête est le prénom de l'auteur, et chaque ligne répond à une question, du type "si j'étais" dont voici la liste :

- un mot : il s'agit juste ici de donner un mot qui sera inclus dans un élément `span` ;
- un avatar : une image vous représentant, si vous n'avez pas d'avatar, il existe différents sites permettant d'en générer gratuitement ;
- un lieu géographique : une carte représentant un lieu de votre choix. Cette carte doit être au format SVG et incluse directement dans votre code HTML. Pour obtenir une telle carte, vous pouvez utiliser le site <https://www.openstreetmap.org/>, chercher le lieu qui vous intéresse, choisir "Partager" dans les icones qui apparaissent à droite, dans le format Image choisir SVG au lieu de PNG, et jouer avec l'échelle pour que l'image récupérée ne soit pas trop volumineuse ;
- une devise : une citation (élément `q`) d'une phrase, d'un proverbe, d'une devise ;
- un objet inutile : objet réel ou imaginaire parfaitement inutile, son nom sera dans un élément `span` ;
- un emoji : l'emoji sera saisi via son code UTF-8. Vous pouvez trouver les codes des emojis définis dans UTF-8 ici : https://www.w3schools.com/charsets/ref_emoji_smileys.asp

b. le fichier `sources.xhtml` référence l'ensemble des sources bibliographiques que vous avez utilisées pour votre site. Les informations sont structurées sous la forme d'une liste de listes respectant l'organisation suivante :

- Sources des images : contient une liste des images utilisées (à l'exception de celles éventuellement générées par une IA) avec un lien sur l'image dans votre site et un lien vers le site d'où provient l'image. Si vous êtes propriétaires de certaines de vos images, vous l'indiquez également ;
- Sources des textes : contient une liste des sites dont vous avez utilisé des textes pour vos fiches (à l'exception des textes éventuellement générés par une IA). Si vous avez écrit vous-mêmes tous les textes, vous l'indiquez également ;
- Documentation technique : une liste des sites que vous avez utilisés comme référence technique (à l'exception d'IA) ;
- IA utilisées : une liste des IA utilisées avec, pour chacune, la liste des utilisations (avec une référence sur la ou les pages du site concernées) et le prompt qui a permis de générer la réponse.

6.4 Formulaires

Le site web comprendra, dans le répertoire **Formulaire**, deux formulaires différents.

Un formulaire peut, à votre convenance :

- permettre de renseigner les différents éléments nécessaires pour créer une nouvelle fiche
- proposer un quizz sur votre collection
- offrir un livre d'or aux internautes visitant votre site

Si vous avez une autre idée pour le contenu, n'hésitez pas à demander.

Comme on travaille côté client sans une vraie base de données, il n'est pas possible d'aller au bout de l'utilisation du formulaire (autrement dit, de créer de nouvelles fiches, de stocker les informations du livre d'or, de mémoriser les résultats du quizz...)

Cependant, pour vérifier que les informations du formulaire sont correctement gérées, vous brancherez le formulaire sur la page de traitement suivante : <http://voca2.univ-poitiers.fr/traitement.php> (accessible seulement depuis le réseau de l'université ou via EduVPN).

7 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 imagination.

- Votre site doit être associé à deux styles : un style principal et un style alternatif. Le style alternatif doit être approprié pour quelqu'un qui aurait des difficultés de vision, vue basse et problème avec les couleurs. Des idées sur comment prendre en compte ce genre de problématique en CSS sont accessibles ici <https://www.w3.org/TR/CSS-access> mais rien ne vous empêche de proposer d'autres approches. Pour les problèmes de visions des couleurs, il peut être judicieux d'utiliser des niveaux de gris.

Vos styles n'ont pas besoin d'être complètement différents et s'ils partagent des règles communes, elles ne doivent pas être dupliquées mais mises dans un fichier de style commun. Autrement dit, vous devez avoir deux styles différents mais il sera probablement judicieux d'avoir au moins 3 fichiers de style. Un qui contient les règles communes aux deux styles et un fichier qui caractérise chacun des styles.

Il peut également être malin d'avoir des fichiers de style spécifique à une page. Par exemple, le fichier `index.xhtml`, et les fichiers présentant les fiches de votre collection ne contiennent a priori pas de formulaire, ils n'ont donc pas besoin de connaître les règles de styles appliqués aux formulaires.

Le découpage de vos deux styles en plusieurs fichiers est laissé à votre libre choix. La seule contrainte est qu'on ne veut pas de duplication de code (autrement dit une même règle dans plusieurs fichiers). Vous indiquerez dans un fichier texte README dans le répertoire **Styles** quels fichiers sont associés à quel style.

- Dans vos fichiers de style doit 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 d'éléments adjacents
 - sélecteurs via des pseudo-classes
 - sélecteurs via des pseudo-éléments
 - sélecteurs via les attributs (autre que *id* et *class*)

Pour utiliser des sélecteurs basés sur la structure de vos pages vous exploiterez le fait que les fiches de la collection ont des structures identiques.

Vous complèterez le fichier `styles.xhtml` fourni dans le répertoire **Styles** en indiquant pour chacun des types de sélecteurs demandés un exemple apparaissant dans un de vos fichiers de style. Pour chaque exemple, vous indiquerez dans le tableau fourni, le nom du fichier de style qui le contient, la ligne où on peut le trouver, et le sélecteur en question.

8 Typescript / javascript

Attention : dans les questions suivantes, quand il s'agira de modifier la structure de la page web, on utilisera les fonctions de manipulation de l'arbre de la page.

Comme indiqué précédemment, on n'utilisera pas de bibliothèques (JQuery ou autres).

8.1 ... sur vos propres pages

On vous demande d'ajouter les éléments dynamiques suivants.

Vous pouvez vous inspirer de ce qui a été fait dans les TPs autour de javascript.

- a. Faites en sorte que la date et l'heure s'affichent à un endroit de votre choix sur chaque page de votre site, lorsque celle-ci a fini de se charger.
- b. Ajoutez un lien dans la page d'accueil qui amène sur une fiche de la collection choisie au hasard. Chaque fois que l'on clique sur le lien, la page qu'il désigne (autrement dit la valeur de l'attribut `href` du lien) change de manière (pseudo-)aléatoire mais désigne toujours une page de votre collection.
- c. Ajoutez sur chaque page un bouton à l'endroit de votre choix et faites en sorte que lorsqu'il est cliqué, la page s'affiche avec le style alternatif que vous avez créé. Si la page s'affiche avec le style alternatif, alors un clic sur le bouton restaure le style initial de la page.
Pour rappel, le style alternatif est celui que vous avez lié à votre page avec un élément `link` dont la valeur de l'attribut `rel` est `"alternate stylesheet"`.
Une manière de faire cela est d'utiliser la collection `document.styleSheets` dont chaque case correspond à une feuille de style (le premier élément de la collection `document.styleSheets[0]` correspond à la feuille de style incluse par le premier élément `link`, etc.). Pour activer ou désactiver un style, on peut utiliser l'attribut `disabled`. Lorsqu'il vaut `true` le style est désactivé, lorsqu'il vaut `false` le style est appliqué sur la page.

8.2 ... dans les formulaires

Le but de ces questions est de rendre votre formulaire plus interactif.

- a. Ajoutez un champ caché dans chacun de vos formulaires dont la valeur est automatiquement mise à jour avec des informations sur le navigateur utilisé, le système d'exploitation sur lequel il s'exécute, et la langue. Lorsque cette information est envoyée sur le serveur, elle est associée au nom `userAgentInformation`;
- b. Modifiez votre formulaire pour que certaines vérifications de validité des 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 (événement `onblur`)). En cas d'erreur de l'internaute, la nature du problème lui sera indiquée par un changement de style, un message dans la page, un message dans une fenêtre d'alerte ou autre. Vous pouvez également ajouter un bouton dans votre formulaire pour tester d'un coup la validité de l'ensemble du formulaire;
- c. [Pour les étudiants des groupes de TD 4 et 5] Ajoutez un bouton à vos formulaires de sorte que lorsqu'on clique sur ce bouton, apparaît sur la page un récapitulatif des informations saisies par l'utilisateur dans le formulaire, précédé par la date et l'heure courantes.

8.3 ... dans la page `autoeval.xhtml`

Pour la suite, notez que lorsqu'on déplace le curseur d'un `input` de type `range`, c'est un événement de type `input` qui est déclenché (cela correspond à l'attribut `oninput`).

- a. L'élément `input` d'identifiant `average` doit contenir la moyenne des valeurs des éléments `input` de type `range` utilisés pour évaluer votre travail.
La valeur de cet élément est mise à jour lorsqu'on charge la page ou lorsqu'on modifie la valeur d'un des curseurs présents dans la page.
- b. Dans la deuxième colonne du tableau, les couleurs du label présent dans chaque case dépendent de la valeur de l'élément `input` présent sur la ligne. Si cette valeur est 0, le texte est écrit en blanc sur fond rouge. Si cette valeur est 100, il est écrit en blanc sur fond vert. Pour toute autre valeur (entre 0 et 100), il est écrit en blanc sur fond orange. Les couleurs doivent être présentes au chargement de la page et modifiée dès que le curseur de sa ligne est déplacé.
- c. [Pour les étudiants des groupes de TD 4 et 5] Au chargement de la page, pour chaque `input` de type `range`, un `span` est ajouté dynamiquement dans la même case du tableau. Il affiche la valeur de l'attribut `value` suivi du caractère `%` avec le même code couleur que précédemment. Lorsque le curseur est déplacé la valeur affichée dans le `span` est mise à jour.