HTML/CSS POUR HAL

Tutoriel

Licence CC : BY-NC-SA. CentraleSupélec - Pôle IST/Amandine Lustrement
Version mise à jour le 5/07/2019

1 Introduction 1
2 Le HTML 3
2.1 Fonctionnement des balises 3
2.2 Les attributs 3
2.3 Différenciation balises block et inline 4
2.4 Structure d'une page web 4
2.5 Le « modèle des boîtes » 4
2.6 Différentes balises pour différentes fonctions 5
2.7 Les balises HTML intéressantes dans HAL 7
3 Le CSS 11
3.1 Fonctionnement 11
3.2 Les balises CSS intéressantes dans HAL 14
3.3 Mieux utiliser le CSS 26
3.4 Quelques bonnes pratiques 27
3.5 Points de vigilance 28
4 Autres constructions spécifiques 30
4.1 Ajouter du multimédia 30
4.2 Faire des boutons 31
4.3 La création d'un fil Twitter 31
5 Se repérer dans HAL 33
5.1 Modifier le contenu HTML de la page 33
5.2 Modifier le CSS 34
5.3 Déposer des fichiers 35
5.4 Ajouter un logo / un bandeau 35
6 Construire son squelette : proposition de trame 36
7 Autres applications : HALtools et CV HAL 41
7.1 HALtools 41
7.2 CV HAL 42
8 Documents et liens utiles 43

1 Introduction

Ce tutoriel a pour objectifs :

- De permettre une personnalisation plus poussée des portails et collections HAL ;
- D'expliquer le fonctionnement du HTML et du CSS, et ainsi donner une plus grande liberté ;
- De proposer une trame de structuration, bien qu'il en existe beaucoup d'autres ;
- D'expliquer les principales balises (HTML) et propriétés (CSS) intéressantes dans ce cadre, ce qui implique une non-exhaustivité de l'ensemble des balises et propriétés qu'il peut exister en HTML et CSS ;

La méthode proposée ici a été retenue car elle :

- Permet de construire rapidement un squelette en utilisant l'interface proposée de base par HAL ;
- Peut être facilement mise à jour ;
- Semble offrir un bon rapport connaissances requises/possibilités de personnalisation ;

Elle est bien entendu discutable compte tenu des possibilités infinies de l'HTML et du CSS. Elle peut aussi simplement servir d'inspiration. Elle a été simplifiée au maximum pour demander le moins de connaissances possibles (nombre de balises, etc.). Par conséquent, suivant vos besoins, vous serez probablement amené à chercher par vous-mêmes de nouvelles balises et propriétés.

Elle n'est pas adaptée pour la construction de pages web « réelles », mais bien uniquement pour des portails et collections HAL.

Si vous souhaitez recevoir une formation complète sur le HTML/CSS, nous vous recommandons le cours dédié sur OpenClassroom (comptez au moins 20h de formation).

Ce tutoriel a été initialement créé pour être le support de l'atelier « Faîtes de votre portail/collection la prochaine égérie de HAL avec HTML/CSS » aux journées CasuHAL du 17-18 juin 2019, hébergées par l'Université de Rouen Normandie.

Depuis, il a été modifié et enrichi. Il ne contient pas les exercices faits lors de l'atelier (voir diaporama).

Besoins :

- Un éditeur de code (Exemples : NotePad++, Sublime Text) ;
- Un compte preprod HAL
- Différents navigateurs d'installés (recommandés à minima : IE/Edge, Firefox, Chrome)

HTML/CSS pour HAL : tutoriel. V.5/7/19. Licence CC : BY-NC-SA. CentraleSupélec - Pôle IST/A.Lustrement

De quoi allons-nous parler ?

Le HTML est une des trois briques indispensables au web créé par Tim Berners-Lee et Robert Cailliau.

Un fichier HTML structure à l'aide de balises déjà définies le contenu d'un document et permet de créer des liens (hyperliens) dedans pour relier les ressources entre elles. **Le CSS est arrivé plus tard, pour mettre en forme le HTML**. Pourquoi ? Car, à la base, une page HTML, ça ressemble à ça :

Un portail HAL sans CSS (les widgets n'apparaissent volontairement pas ici) :

Un portail HAL avec un CSS personnalisé :

Nous en sommes aujourd'hui aux versions **HTML5** et **CSS3**. Ces deux langages ont des syntaxes et des logiques bien distinctes que nous allons désormais voir.

2 Le HTML

Ici, l'objectif n'est pas de savoir écrire en HTML, mais de comprendre comment il est construit, et quelles balises correspondent à quel(s) élément(s). De cette manière, vous pourrez les repérer dans le code, et les personnaliser avec le CSS.

2.1 Fonctionnement des balises

Les balises en HTML sont toujours entre < >.

Elles fonctionnent le plus souvent par paires : il y a **une balise ouvrante < > et une balise fermante </ >, qui permettent de délimiter la zone**/le texte concerné(e).

Avec

```
<h1>Ceci est une balise de titre</h1>
<p>Ceci est une balise de paragraphe.</p>
```

s'affichera

Ceci est une balise de titre

Ceci est une balise de paragraphe.

Il existe des **balises auto-fermantes (« orphelines »)**, comme les images ou les sauts de ligne. Il s'agit en effet juste d'un élément qu'on va insérer à un endroit précis, il n'y a pas de zone à délimiter. Elle n'a donc pas de balise fermante, et peut ressembler, pour les images à : `<img />`. Leur contenu est précisé si besoin avec des attributs.

2.2 Les attributs

Les attributs sont ajoutés à l'intérieur d'une balise ouvrante ou d'une balise orpheline.

Ils apportent des précisions comme une adresse url, une source, parfois de la mise en forme (cf. CSS) ou une particularité (ouvrir le lien dans une nouvelle fenêtre ou non).

Quelques attributs que nous allons croiser :

href : indique une adresse web vers laquelle on va aller si on clique sur l'élément

```
<a href="monadresseweb.fr">Cliquez ici !</a>
```

src : indique l'adresse web de l'élément qui va apparaître (ex : image, vidéo)

alt : indique un texte alternatif si le contenu ne s'affiche pas

```
<img src="foret.png" alt="photo de forêt" />
```

style : permet d'inclure du CSS directement dans le code HTML

```
<p style="color:blue; font-size:1.3em;">
```

2.3 Différenciation balises block et inline

La plupart des balises HTML sont des balises de type « block » : elles **délimitent une zone**, et **se placent automatiquement les unes en-dessous des autres.** On peut insérer des balises inline dedans.

Les **balises inline s'insèrent obligatoirement dans une balise block**, **mais ne demandent pas de retour à la ligne** (comme l'indique leur nom !). C'est le cas par exemple de la balise `<a>` qui sert à indiquer un lien hypertexte. Elles ne peuvent pas contenir de balises block.

2.4 Structure d'une page web

Une page est généralement structurée de :

- `<head>` : contient des **métadonnées** sur la page qui ne sont pas visible par l'utilisateur. Par exemple : l'encodage, le lien vers la feuille CSS, le titre du site web… Bref, toutes les informations nécessaires à son identification et à son bon fonctionnement !

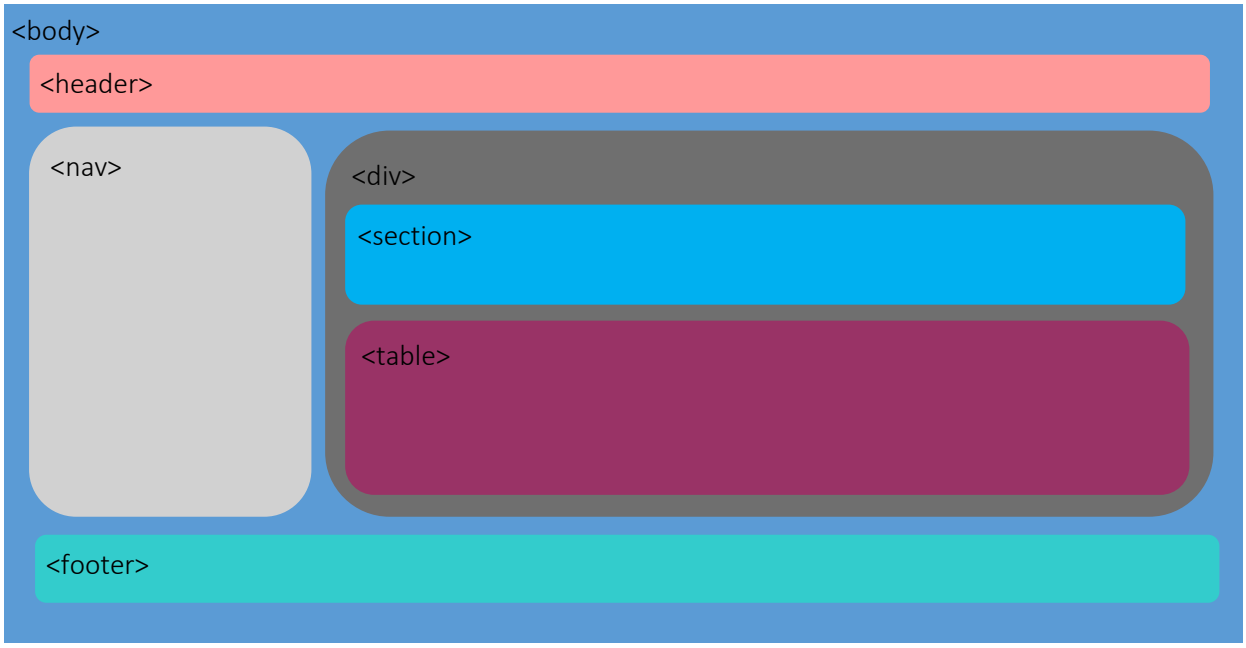
  Nous n'intervenons pas dessus : tout est déjà renseigné par HAL.

A ne pas confondre avec le `<header>`, qui est un en-tête visible sur votre page !

- `<body>` : le corps de la page. C'est là où nous allons travailler !
- `<footer>` : pied de page. Modifiable dans *Privilèges > Site web > Pied de page > Pied de page personnalisé*

2.5 Le « modèle des boîtes »

En HTML, on emboîte les balises les unes dans les autres. **Elles ne peuvent jamais se chevaucher :**



Il est donc impossible de faire ceci :

```
<h1>Ceci est une balise de titre
<p>Ceci est une balise de paragraphe.</h1></p>
```


HTML/CSS pour HAL : tutoriel. V.5/7/19. Licence CC : BY-NC-SA. CentraleSupélec - Pôle IST/A.Lustrement

2.6 Différentes balises pour différentes fonctions

Il existe une bonne centaine de balises HTML ! Heureusement, nous n'avons pas à toutes les connaître.

Ci-dessus, nous avons par exemple `<nav>` qui indique une zone pour placer le menu pour naviguer, `<header>` pour l'en-tête de la page, `<div>` pour délimiter une zone, `<section>` qui délimite un bloc avec du contenu, `<table>` pour un tableau et `<footer>` pour le bas de page.

Dans celles mentionnées ci-dessus, dans HAL, nous n'utiliserons que `<table>` !

Sachez que (presque) **tout est adaptable**. Toutes les balises n'ont pas à être utilisées, et il y a plusieurs façons d'utiliser les balises pour arriver à un même rendu (attention cependant aux bonnes pratiques).

Prenons l'exemple de la navigation.

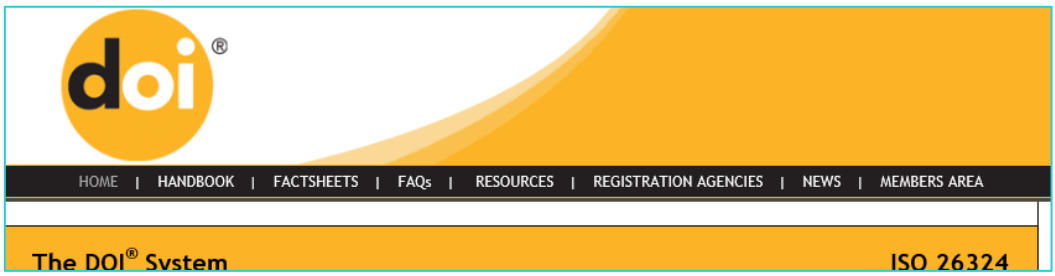
Cas n°1. Il existe la balise dédiée `<nav>`, qui pourrait ressembler à ça :

```
<nav>
  <ul>
    <li><a href="#Accueil">Accueil</a></li>
    <li><a href="#Apropos">À propos</a></li>
    <li><a href="#Contact">Contact</a></li>
  </ul>
</nav>
```

Ce qui donne, sans mise en forme :

- Accueil
- A propos
- Contact

Cas n°2 (doi.org) :



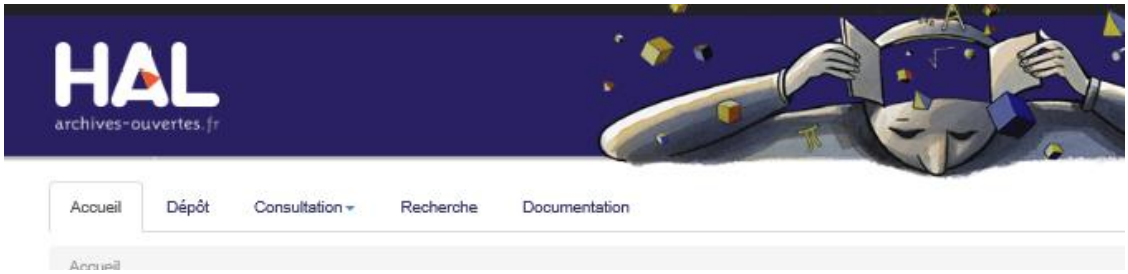
```
<table width="100%" border="0" cellpadding="0" cellspacing="0" id="navtable" align="
    <tr>
        <td width="34" height="26" bgcolor="#231f20"><img src="img/transparent.gif"
        <td height="26" bgcolor="#231f20" class="navtext">
            <span style="color: #acacac; padding-left: 10px; padding-right: 5px;">HC
            <a href="hb.html">HANDBOOK</a>  | 
            <a href="factsheets.html">FACTSHEETS</a>  | 
            <a href="faq.html">FAQs</a>  | 
            <a href="resources.html">RESOURCES</a>  | 
            <a href="registration_agencies.html">REGISTRATION AGENCIES</a>  |&r
            <a href="announce.html">NEWS</a>  | 
            <a href="idf-members/index.html">MEMBERS AREA</a>
        </td>
    </tr>
</table>
```

`<nav>` est remplacé par un tableau `<table>`. On a un tableau avec deux lignes : une pour le bandeau, une deuxième qui contient tous les mots de la navigation séparés par le caractère « | ».

Ce qui donne :

Bandeau doi.org
Handbook | Factsheets | FAQS | etc.

Cas n°3 (HAL) :




```
<div id="global-navigation" class="sidebar-nav" role="navigation">
<ul class="nav nav-tabs">
    <li class="active"><a href="/">Accueil</a></li>
    <li class=""><a href="/submit/index">Dépôt</a></li>
    <li class="dropdown "><a href="/section/list2" class="dropdown-toggle" data-toggle="dropdown">Consultation<
        <ul class="dropdown-menu">
            <li><a href="/browse/last">Les derniers dépôts</a></li>
            <li><a href="/browse/doctype">Par type de publication</a></li>
            <li><a href="/browse/domain">Par discipline</a></li>
            <li><a href="/browse/period">Par année de publication</a></li>
            <li><a href="/browse/structure">Par structure de recherche</a></li>
            <li><a href="/browse/portal">Les portails de l'archive</a></li>
            <li><a href="/browse/collection">Les collections</a></li>
        </ul>
    </li>
    <li class=""><a href="/search/index">Recherche</a></li>
    <li class=""><a href="https://doc.archives-ouvertes.fr" onclick="this.target='_blank'">Documentation</a></li>
</ul>
</div>
```

`<nav>` est remplacé `<div>`, avec une liste de puces à l'intérieur. Ce qui donne :

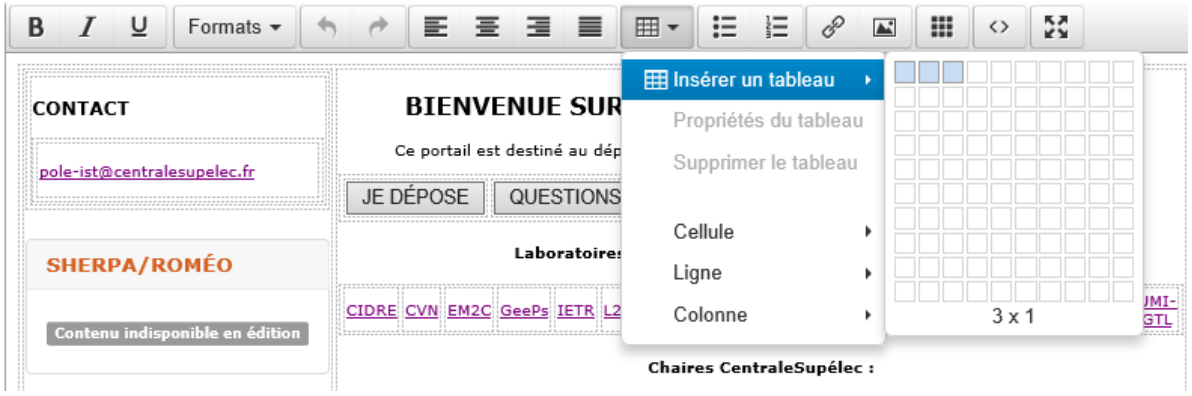
- Les derniers dépôts
- Par type de publication
- Par discipline
- Etc.

Nous obtenons donc des résultats similaires, pourtant ce ne sont pas les mêmes balises !

2.7 Les balises HTML intéressantes dans HAL

2.7.1 Le squelette / balises « conteneurs »

Un site internet se construit généralement avec des balises complétées par des classes ou id. **Dans HAL, la structure se fait plus facilement à partir de tableaux (`<table>`).**



Ce qui écrira dans le code HTML (uniquement le texte en blanc) :

```
<table> = début du tableau
 <tbody> = corps du tableau, par opposition à <thead> qui
correspond aux en-tête des colonnes de tableaux

      <tr> = début de la ligne
           <td>contenu de la cellule</td>
           <td>contenu de la deuxième cellule, qui va créer
une deuxième colonne</td>
           <td>contenu de la 3e cellule (3e colonne)</td>
      </tr> = fin de la ligne

 </tbody> = fin du corps du tableau
</table> = fin du tableau
```

Note : en réalité, quand on fait un tableau, préciser `<thead>` et `<tbody>` (ligne d'en-tête / corps du tableau) n'est pas obligatoire, mais HAL vous rajoute automatiquement `<tbody>` quand vous validez le code.

Ensuite, à la manière des poupées russes, vous pouvez insérer des tableaux dans des tableaux pour créer votre mise en page.

« - Pourquoi des tableaux dans des tableaux !? »

Tout d'abord, l'interface HAL est faite pour, ce qui permet de construire les grands éléments du site avec cette interface. Les tableaux sont simples à mettre en forme, et cela permet d'offrir une certaine liberté dans la disposition des contenus en demandant le moins de manipulations possibles.

2.7.2 Balises de texte

Les titres

Vont de h1 à h6, h1 étant le plus important, et h6 le moins important.

Point de vigilance : il ne pas confondre avec `<title>` qui est le titre de votre page, renseigné dans le `<head>` (métadonnées du site web).

Ces balises se mettent avec *Formats > Headings*

Le paragraphe

`<p> </p>`

Note : la balise `<br/>` pour le saut de ligne n'est pas prise en compte dans HAL. Pour sauter une ligne, il faut fermer le paragraphe actuel et en ouvrir un nouveau
(se fait automatiquement quand vous sautez une ligne dans l'interface HAL).

Les textes importants

Il existe plusieurs balises pour signaler un texte important, dont :

`<strong>` : correspond de base à la mise en forme « gras »

`<em>` : correspond de base à la mise en forme « italique »

Rappel : ces mises en forme sont juste là à titre indicatif. Vous pouvez passer les balises en non-gras, rose ou bleu, taille 8 ou 40… !

Les listes

On signale une liste en la plaçant entre les balises `<ol>` et `</ol>` pour une liste ordonnée (numérotée), et `<ul>` et `</ul>` pour une liste non numérotée.

Chaque élément de la liste est balisé avec `<li>` et `</li>`.

```
<h3>Quelques merveilleux chercheurs qui déposent sur HAL</h3>
  <ul>
    <li>Mr X</li>
    <li>Mr Y</li>
    <li>Mr Z</li>
  </ul>
```

Les liens

Un lien est signalé avec la balise `<a> </a>`. C'est une balise inline, c'est-à-dire qu'elle reste « sur la ligne » et ne force pas le retour à la ligne.

Dans la balise ouvrante, il faudra indiquer l'adresse à laquelle on veut accéder (cf. attributs). Entre les deux balises, on place le texte qui sera visible par l'utilisateur, et qui en cliquant dessus l'amènera à l'adresse indiquée dans la balise.

```
<a href="https://monadresse.fr">Le texte qui sera visible</a>
```

Attention à bien mettre les guillemets de chaque côté de l'adresse !

2.7.3 Les id et les classes

L'idée est ici de pouvoir donner des noms uniques à des balises génériques (a, p, table…), pour les appeler dans le CSS et les différencier pour leur mise en forme.

Concrètement, vous pouvez ainsi (exemples) :

- Créer des listes `<ul><li>` avec des puces différentes : une avec des carrés, l'autre avec des flèches…
- Avoir des cellules `<td>` avec un fond bleu, d'autres avec une police en vert fond transparent…
- Définir des dimensions précises pour absolument ***chaque*** élément
- Etc.

Id

Un id signale un élément unique dans votre code. Il ne peut apparaître qu'une seule fois (deux balises ne peuvent avoir le même id). Voici comment l'écrire en HTML (exemple pour une cellule de tableau) :

```
<balise id="nomdemonid">
```

```
<td id="encart-contact">
```

Class

Une classe peut être appliquée à plusieurs balises, même différentes.

```
<balise class="maclasse">
```

```
<td class="cellules-laboratoires">
<p class="cellules-laboratoires">
```

Attention : sachez que toutes les propriétés CSS ne sont pas applicables sur toutes les balises

Les règles de nommage (id et classes)

- Vous choisissez vous-même le nom
- Evitez les accents, caractères spéciaux, espaces…
- Vous pouvez combiner classe(s) et id pour une même balise HTML
- Attention à placer votre classe ou id dans la bonne balise (<table>/<td>, etc.)

2.7.4 Petit rappel des balises HTML indispensables

Construction du tableau		
<table>	</table>	*délimitation du tableau*
<td>	</td>	*cellule*
Texte		
		paragraphe
<a>		*lien*
Titres		
<h1>	</h1>	*titre le plus important*
<h2>	</h2>	
<h3>	</h3>	
<h4>	</h4>	
<h5>	</h5>	
<h6>	</h6>	*titre le moins important*
Mise en valeur du texte		
Créer des listes		
		liste numérotée
		liste non numérotée
		élément de la liste
<button>	</button>	*création d'un bouton*

Retrouvez également ces indications dans le mémo des balises, onglet HTML disponible ici

HTML/CSS pour HAL : tutoriel. V.5/7/19. Licence CC : BY-NC-SA. CentraleSupélec - Pôle IST/A.Lustrement

3 Le CSS

Vous avez probablement remarqué que lorsqu'on applique certaines balises en HTML, elles sont parfois mises en forme différemment. C'est le cas de `<h1>` ou `<h2>` qui délimitent des titres : ils apparaissent en beaucoup plus gros que les paragraphes `<p>`. Idem pour `<strong>` qui signale un élément important et affiche le texte en gras. **Cependant, il s'agit juste de repères visuels, et non pas d'une mise en forme réelle. Si vous voulez aller au-delà de la taille de police et du gras, il faut utiliser le CSS.**

3.1 Fonctionnement

Pour indiquer à l'ordinateur la mise en forme qu'il doit appliquer, nous devons indiquer pour chaque balise HTML les propriétés CSS à appliquer. Pour se faire, il existe trois manières :

1. En écrivant directement toutes les balises dans un fichier .css (distinct du fichier .html). On indique ensuite dans le fichier HTML où aller chercher les informations de mise en page.

C'est le contenu de ce fichier que nous copions dans la partie *Préférences > Apparence*. En utilisant un tel fichier, nous n'écrivons qu'une seule fois ce dont nous avons besoin pour chaque balise ! La syntaxe pour le fichier.css est :

```
Nom-de-la-balise-HTML {
propriété : valeur;
propriété2 : valeur1 valeur2;
}

Nom-de-la-balise-HTML-2 {
propriété : valeur;
propriété2 : valeur1 valeur2;
}
```

```
h1 {
color: blue;
border: solid 1px blue;
}

h2 {
color: red;
}
```

1. Noter le nom de la balise HTML.
2. Placer les propriétés entre accolades { }.
3. Après la propriété, ajouter deux points :
4. A la fin de chaque valeur, placer un point-virgule ;

HTML/CSS pour HAL : tutoriel. V.5/7/19. Licence CC : BY-NC-SA. CentraleSupélec - Pôle IST/A.Lustrement

La ponctuation doit être impérativement respectée, autrement cela génère des bugs !

Note : la bonne pratique veut qu'on passe à la ligne pour chaque nouvelle propriété, mais sachez qu'il est tout à fait possible de tout écrire sur une seule et même ligne. **C'est la ponctuation qui permet au robot de séparer et définir quelles propriétés vont avec quelle balise.**

```
h1 {color: blue; border: solid 1px blue;} h2 {color:red ;
font-size : 0.8em ; font-family : verdana, arial;} table
{border : solid 1px white ; border-radius : 2px 4px 2px 4px ;
border-bottom :none ;}
```

Difficilement lisible, non ?

Voici comment les deux fichiers, .css et .html, s'y retrouvent :

```html
<td style="width:20%;">
    <table class="module">
        <tbody>
            <tr>
                <td class="head-cellule">Lexique</td>
            </tr>
            <tr>
                <td>
                    <ul>
                        <li>Archive ouverte <a title="bulle d'information">&#x2139;</a></li>
                        <li>Creative Commons <a title="bulle d'information">&#x2139;</a></li>
                        <li>Libre accès <a title="bulle d'information">&#x2139;</a></li>
                        <li>Propriété intellectuelle <a title="bulle d'information">&#x2139;</a></li>
                    </ul>
                </td>
            </tr>
        </tbody>
    </table>

    <table class="module">
        <tbody>
            <tr>
                <td class="head-cellule">Questions similaires</td>
            </tr>
            <tr>
                <td>Vide</td>
            </tr>
        </tbody>
    </table>
```

```css
body {
    background-color: none;
    font-family: arial;
}

table {
    border-spacing: 0;
}

td {
    padding: 10px;
    vertical-align: top;
    text-align: justify;
}

p {
}

a {
    color: rgb(233,116,2);
    text-decoration: none;
}

strong {
    
}

em {
    
}

.module {
    border: solid 1px rgb(233,116,2);
    border-radius: 10px;
    width:100%;
    margin-bottom:30px;

}

.head-cellule {
    background-color: rgb(233,116,2);
    color: white;
    font-weight: bold;
    border-radius: 10px 10px 0 0;
}
```

2. Au sein des balises HTML avec l'attribut « style » : le HTML permet de préciser certaines choses grâce aux attributs (la mise en forme avec « style », le lien vers une image avec « src », etc.). Le gros problème de cette méthode est qu'elle alourdit considérablement le code et qu'il faut tout remettre dans chaque balise ! Si vous avez passé tous vos paragraphes <p> en texte rouge et que vous changez d'avis, vous devrez remodifier chaque balise <p>. Néanmoins, dans certains cas, nous serons amenés à utiliser cette méthode. La syntaxe pour l'écriture ressemble à la précédente :

```
<balise style="propriété: valeur ;">
```

```
<h1 style="color: blue ; border: solid 1px blue ;">
```

```
<h1 style="font-size:1.4em ; color:blue ;">Ceci est mon titre important<h1>

<h2 style="font-size:1.2em ; text-decoration: underline;">Ceci est mon titre moins important<h2>

<p>Voici mon premier paragraphe <a href="lien.fr" style="font-weight:bold;">avec un lien</a>.</p>

<p style="color:grey;">Et un autre paragraphe !</p>
```

3. En insérant la feuille de style dans la feuille HTML au sein de balise `<style>` dans `<head>` (non utilisé ici) ;

HTML/CSS pour HAL : tutoriel. V.5/7/19. Licence CC : BY-NC-SA. CentraleSupélec - Pôle IST/A.Lustrement

3.2 Les balises CSS intéressantes dans HAL

3.2.1 Dimensions

Uniquement pour les balises de type block !

Propriété 1 : `width`

Permet de définir la largeur d'un élément

- Valeur : valeur numérique comme px, cm ou %. Par défaut : auto

```
table {width: 80%;}
```

Note : il existe également min-width et max-width, qui prennent le pas sur width

Propriété 2 : `height`

Permet de définir la hauteur d'un élément

- Valeur : valeur numérique comme px, cm ou %. Par défaut : auto

```
table {height: 400px;}
```

Note : il existe également min-height et max-height, qui prennent le pas sur height

3.2.2 Couleur

Propriété : `color`

Valeurs :

- white, black, gray, silver, maroon, red, lime, green, yellow, olive, navy, blue, purple, fushia, aqua, teal

```
h1 { color: blue;}
```

- notation hexadécimale

```
h1 { color: #00FF44;}
```

```
h1 { color: #0F4;}
```

- notation RGB (red green blue)

```
h1 { color: rgb(204,90,206);}
```

Pour en savoir + : https://www.w3schools.com/colors/default.asp

Trouver un code couleur : http://www.code-couleur.com

3.2.3 Modifier son texte

Taille

Propriété : `font-size`

Valeurs :

- px, cm ou mm

```
h1 { font-size: 10px;}
```

- em

```
h1 { font-size: 1.2em;}
```

- ex (légèrement plus grand que em)

```
h1 { font-size: 1ex;}
```

- %

```
h1 { font-size: 80%;}
```

- xx-small, x-small, small, medium, large, x-large, xx-large

```
h1 { font-size: medium;}
```

Il est recommandé d'utiliser em, ex ou des %, qui vont s'adapter aux résolutions de chaque utilisateur

Police

Propriété : `font-family`

Il est possible de déclarer plusieurs polices. Pour les polices avec des noms composés, il faut utiliser des tirets. En lisant, l'ordinateur appliquera la première police qu'il voit. S'il ne l'a pas d'installée, il passera à la seconde, etc. Il est donc conseillé de mettre en dernier une police que tout système a de base, comme Arial.

```
h1 {font-family : Verdana, arial ;}
```

Vous pouvez utiliser une police plus originale que ce qui est proposé d'office. Pour se faire, il faut la déclarer avec **@font-face**. Plusieurs choses sont à prendre en compte :

- les droits d'auteur pour l'utilisation de la police
- il faut des formats différents suivant les navigateurs
- le navigateur va télécharger la police, ce qui est parfois assez lourd

```
@font-face {
font-family: 'nouvellepolice';
src: url('nouvellepolice.eot');
src:    url('nouvellepolice.eot?#iefix')    format('embedded-
opentype'),
     url('nouvellepolice.otf') format('truetype'),
```

```
     url('nouvellepolice.woff') format('woff'),
     url('nouvellepolice.svg#nouvellepoliceBold')
format('svg');
}
```

Pour télécharger des polices : https://www.dafont.com . Vous pouvez filtrer dans « plus d'option » pour avoir des polices non payantes.

Pour mieux comprendre le fonctionnement de @font-face :
Cours Open Classroom HTML/CSS - « formatez du texte »

Décorations autres

Propriété 1 : `font-style`

Valeurs :

- Italic
- Oblique
- Normal (par défaut)

```
h1 {font-style: italic;}
```

Propriété 2 : `font-weight`

Valeurs :

- Bold
- Normal (par défaut)

```
h1 {font-weight: bold;}
```

Propriété 3 : `text-decoration`

Valeurs :

- Underline : souligné
- Overline : ligne au-dessus
- Line-through : barré
- Blink : clignotant (attention aux navigateurs, tous ne l'acceptent pas)
- None : normal, par défaut

```
h1 {text-decoration: underline;}
```

Alignement

Propriété : `text-align`

Attention : ne fonctionne que sur les balises de type block.

Valeurs :

- Left
- Center
- Right
- Justify

```
h1 {text-align: center;}
```


HTML/CSS pour HAL : tutoriel. V.5/7/19. Licence CC : BY-NC-SA. CentraleSupélec - Pôle IST/A.Lustrement

3.2.4 Modifier son fond

Super-propriété : ***background***

Les super-propriétés, qu'est-ce que c'est ? Une super-propriété rassemble plusieurs propriétés d'un type similaire. Cela évite de multiplier les lignes.

- ⇨ Ici, `background` va regrouper `background-color`, `background-image`, `background-repeat`, `background-position`, etc. L'ordre dans lequel les valeurs sont renseignées importe peu.

Couleur

Propriété : `background-color`

Valeurs : cf. couleurs

```
td {background-color: #f9429e;}
```

Image

Propriété 1 : `background-image`

Indique l'adresse de l'image choisie comme fond. Valeur unique : **url(monimage.png)**

Le nom du fichier, doit être exactement le même que dans *Privilèges* > *Ressources*, avec son extension.

Il est possible d'héberger l'image ailleurs et de mettre l'adresse complète à la place du nom de fichier déposé sur HAL.

```
body {background-image: url(header_HAL_CS_2_80.jpg) ;}
```

```
body{background-image:
url(https://monsite.fr/header_HAL_CS_2_80.jpg);}
```

Propriété 2 : `background-attachment`

Valeurs :

- Fixed : fixe le fond, le texte glisse par-dessus quand on descend dans la page ;
- Scroll : l'image défile avec le texte. Par défaut.

```
body {
background-image: url(header_HAL_CS_2_80.jpg);
background-attachment: fixed;
}
```

Propriété 3 : `background-repeat`

Valeurs :

- Repeat : par défaut. Répète l'image en mosaïque ;
- No-repeat
- Repeat-x : répète uniquement à l'horizontale ;
- Repeat-y : répète uniquement à la verticale ;

```
body {
background-image: url(header_HAL_CS_2_80.jpg);
background-repeat: repeat-x;
}
```

Propriété 4 : `background-position`

A n'utiliser que si : `background-repeat: no-repeat;`

Permet d'indiquer où l'image de fond doit se placer. Valeurs :

- En pixels : la première valeur indique la position par rapport au bord gauche de la page, la seconde par rapport au bord supérieur

```
body {
background-image: url(header_HAL_CS_2_80.jpg) ;
background-repeat: no-repeat ;
background-position: 20px 40px ;
}
```

- Top, bottom, left, center, right. Ces valeurs peuvent être combinées

```
body {
background-image: url(header_HAL_CS_2_80.jpg) ;
background-repeat: no-repeat ;
background-position: top left ;
}
```

Exemples pour la superpropriété `background` :

```
body {background : url(header_HAL_CS_2_80.jpg) no-repeat 20px 40px ;}
```

```
body {background : url(header_HAL_CS_2_80.jpg) fixed ;}
```

```
td {background : #f9429e;}
```

Si vous déclarez plusieurs images, il faut les séparer par une virgule :

```
body {background : url(header_HAL_CS_2_80.jpg) no-repeat 20px 40px, url(image_supplementaire.jpg) no-repeat top right;}
```

En superposant plusieurs images, la première déclarée sera placée au-dessus des autres.

3.2.5 Bordures

Super-propriété : ***`border`***

⇨ Prend en compte `border-color`, `border-width` et `border-style`.

Définir la couleur : `border-color`

Valeurs : cf.couleurs

Définir l'épaisseur : `border-width`

Valeur : en px.

Une seule valeur d'indiquée applique la même bordure sur les 4 côtés. Pour appliquer une valeur différente à chaque côté, l'ordre est le suivant : haut – droite – bas – gauche.

```
td {border-width : 2px ;}
```

Pour une bordure de 2px en haut et en bas (rien sur les côtés) :

```
td {border-width : 2px 0 2px 0;}
```

Définir le type de trait : `border-style`

Valeurs :

- None : par défaut
- Solid : trait plein
- Double : trait double
- Dotted : pointillés
- Dashed : tirets
- Groove : rainuré
- Ridge : relief
- Inset : relief intérieur
- Outset : relief extérieur

```
td {
border-width : 1px;
border-style : dotted;
}
```

Attention : pour afficher une bordure, il est obligatoire d'indiquer une épaisseur avec border-width car la valeur par défaut est de 0 (soit rien !). Idem pour border-style car par défaut, la valeur est nulle.

Définir un angle : `border-radius`

Valeur : en px ou %

Une seule valeur d'indiquée applique le même angle aux **4 coins**. Pour appliquer une valeur différente à chaque côté, l'ordre est le suivant : haut/gauche – haut/droite – bas/droite – bas/gauche.

```
td {border-radius : 4px 4px 0 0;}
```

…donnera une bordure avec les deux angles du haut arrondis à 4px.

Attention : la propriété border-radius est la seule qui n'est pas inclue dans la super-propriété border

Exemple :

```
td {
border : solid 1px blue ;
border-radius : 4px
}
```

Il existe également des propriétés spécifiques pour chaque côté :

```
border-top
border-left
border-bottom
border-right
```

```
td {
border-top : solid 2px green ;
}
```

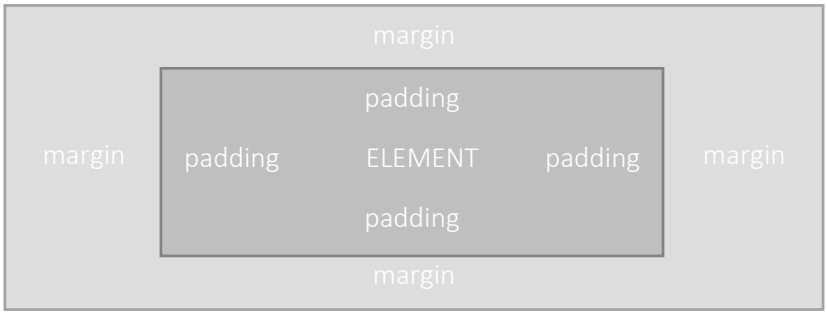
équivaut à

```
td {
border : 2px 0 0 0 solid green ;
}
```

Autre exemple :

```
td {
border-top : solid 1px blue ;
border-bottom : dotted 4px red ;
}
```

3.2.6 Les marges extérieures et intérieures



Super-propriété : margin
Propriétés : margin-top / margin-right / margin-bottom / margin-left

Utilisée pour définir la marge extérieure à l'élément.

Valeurs : en px ou « auto », dans l'ordre suivant : Haut, droite, bas, gauche. Une seule valeur applique la même marge à l'ensemble.

```
td { margin : 4px ;}
```

Propriété : padding
Propriétés : padding-top / padding-right / padding-bottom / padding-left

Utilisée pour définir la marge à l'intérieur de l'élément.

Valeurs : en px, dans l'ordre suivant : Haut, droite, bas, gauche. Une seule valeur applique la même marge à l'ensemble.

```
td {padding : 5px 0 5px 0 ;}
```

Propriété : border-spacing

Gère l'espace entre les cellules. S'applique uniquement à <table> et si on a indiqué avec border-collapse:separate (pour décoller les bordures des cellules et permettre ainsi d'avoir de l'espace entre elles).

On peut indiquer deux valeurs : la première sera pour l'espace horizontal, la deuxième pour l'espace vertical.

```
table {
border-collapse :separate ;
border-spacing : 2px ;
}
```


HTML/CSS pour HAL : tutoriel. V.5/7/19. Licence CC : BY-NC-SA. CentraleSupélec - Pôle IST/A.Lustrement

3.2.7 Les utiles

Gérer le texte qui dépasse

Propriété : `word-wrap`

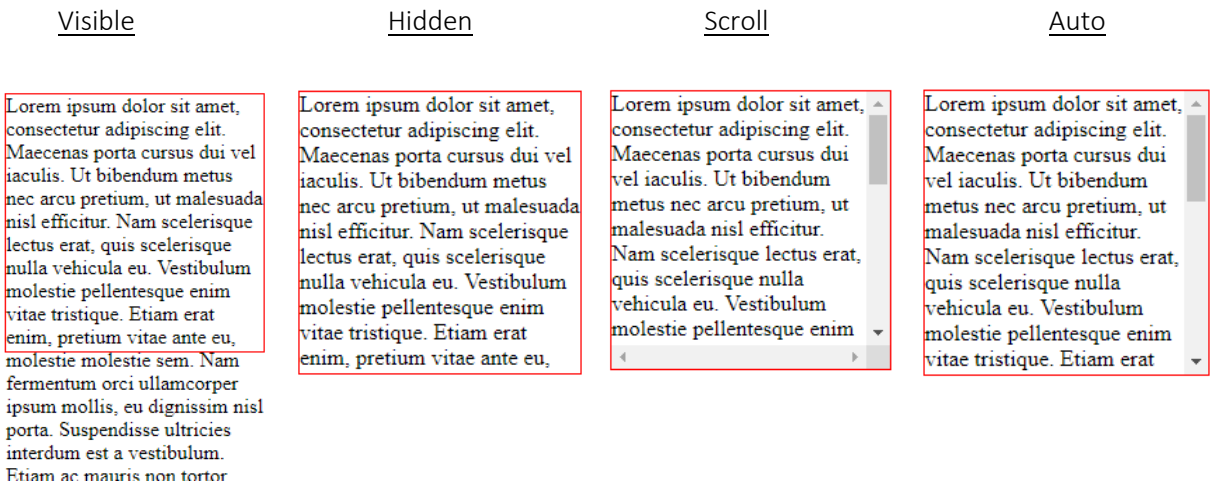
Permet de couper les mots qui sont trop longs et qui risquent de dépasser. Valeurs :

- Break-word
- Normal (par défaut)

Propriété : `overflow`

Sert à empêcher que le texte dépasse d'un bloc. Valeurs :

- Visible : par défaut. Le texte qui sort du bloc est visible ;
- Hidden : Le texte qui sort du bloc n'est pas visible ;
- Scroll : ajout de barres de défilement pour voir le texte qui dépasse, sans que celui-ci ne sorte du cadre ;
- Auto : le navigateur décide lui-même si mettre des barres de défilement est pertinent ou non. Cela permet d'adapter en fonction de la résolution de l'utilisateur. Dans les exemples ci-dessous, il a retiré la barre allant de droite à gauche :



Positionner des éléments

Nous avons vu que les balises block se placent systématiquement les unes en dessous des autres. Si nous voulons par exemple que des `<table>` se placent les unes à côté des autres, il faut changer leur propriété « block » en « inline » (qui, elles, se placent à côté). L'inverse est aussi possible !

Propriété : `display`

Valeurs :

- Inline
- Block
- Inline-block
- None : devient un élément non affiché, qu'on pourra faire réapparaître avec un autre langage par la suite

Propriété : `vertical-align`

Permet d'aligner verticalement le contenu d'une **cellule** (`<td>`).

Valeurs :

- Top
- Bottom
- Middle

```
td {vertical-align : middle ;}
```

Propriété : `margin`

Nous avons vu plus haut que `margin` permet de donner des marges extérieures. Il est parfois judicieux de l'utiliser pour aligner des éléments au centre.

```
body {
margin-left : 10% ;
margin-right :10% ;
}
```

```
#cellulespeciale { margin :auto ;}
```


HTML/CSS pour HAL : tutoriel. V.5/7/19. Licence CC : BY-NC-SA. CentraleSupélec - Pôle IST/A.Lustrement

3.2.8 Les pseudo-classes

Lorsque nous passons notre curseur sur un lien ou un bouton, nous pouvons remarquer que celui-ci change légèrement d'aspect. De même, il arrive qu'un lien soit d'une couleur différente après que nous ayons visité le site sur lequel il renvoie. Toutes ces mises en forme sont possibles grâce aux « **pseudo-classes** ».

D'un point de vue ergonomique, les pseudo-classes sont très importantes. En effet, l'utilisateur a besoin d'être informé visuellement qu'un élément est cliquable, et qu'il a correctement cliqué dessus le cas échéant.

On vient placer la pseudo-classe après la balise concernée, séparés par deux points :

```
balise :pseudoclasse {propriété1 : valeur ;}
```

Pour changer d'apparence quand on survole l'élément :

Pseudo-classe : `:hover`

```
a { color : blue ;}
a :hover {color :red ;}
```

Pour changer d'apparence quand on sélectionne (surligne) un élément :

Pseudo-classe : `:focus`

```
a { color : blue ;}
a :focus {color :red ;}
```

Dans les faits, il y a peu de différence entre :hover et :focus pour l'utilisation qu'on peut en faire dans notre cadre.

Pour changer d'apparence quand on clic sur l'élément :

Pseudo-classe : `:active`

```
a { color : blue ;}
a :active {color :white ;}
```

Pour changer l'apparence d'un élément déjà visité (de base est en bordeau/violet) :

Pseudo-classe : `:visited`

```
a { color : blue ;}
a :hover {color :green ;}
```

⇨ *Avec ces exemples, j'ai donc des liens affichés normalement en bleu, qui passent en rouge quand je les survole, en blanc quand je clique dessus, et deviennent verts après que j'ai cliqué (après les avoir visités).*

3.3 Mieux utiliser le CSS

3.3.1 Sélecteurs CSS

Il existe des sélecteurs CSS qui permettent de sélectionner rapidement certaines balises.

Pour sélectionner uniquement les liens dans les cellules qui ont la classe `.celltable1` :

```
.celltable1 a {
}
```

Si plusieurs éléments partagent beaucoup de caractéristiques communes, vous pouvez leurs attribuer les mêmes valeurs en une seule fois en séparant les balises par une virgule (ce qui ne vous empêche pas de reprendre la balise seule après et de lui attribuer des valeurs spécifiques):

```
.celltable1, .celltable2{
 background: #bfb8cd;
 color: #807f79;
}

.celltable1 {
border: 1px blue solid ;
}
```

Pour connaître tous les sélecteurs CSS :

http://easycss.fr/cibler-vos-elements-selecteurs-css/

3.3.2 Principes d'hérédité

Un élément enfant hérite des propriétés de l'élément parent.

Par exemple, en attribuant à `<body>` la police Verdana, tous les textes de votre page vont avoir la police Verdana. En effet, toutes nos balises sont contenues dans `<body>`.

Si vous appliquez la couleur bleue à toutes les `<table>` alors tous les textes contenus dans les `<td>` seront bleus également. (mais pas les textes qui ne sont pas hors des tableaux).

Pour en savoir + : l'article dédié sur Alsacreations https://www.alsacreations.com/tuto/lire/545-Comprendre-l-heritage-et-la-parente-des-styles-CSS.html

3.3.3 Principes de lecture (ordre)

La propriété CSS qui l'emportera sera toujours la plus précise ou celle écrite en dernier.

Exemples :

- Les propriétés d'un id l'emporteront sur les propriétés données à la balise générale ;

- Je définis `strong {font-weight : bold ;}` en haut de mon code et `strong {font-weight :normal ;}` en bas : mon texte `<strong>` ne sera pas en gras.

3.4 Quelques bonnes pratiques

3.4.1 L'indentation

Si vous utilisez un éditeur de code, indenter votre code permet de mieux s'y retrouver ! Il s'agit des tabulations qui permettent de décaler vos lignes pour qu'elles soient plus lisibles.

Sans indentation :

```
<h4>[Titre de section]</h4>
<ul>
<li>[Point 1]</li>
<li>[Point 2]</li>
<li>[Point 3]</li>
</ul>
```

Avec indentation :

```
<h4>[Titre de section]</h4>
	<ul>
		<li>[Point 1]</li>
		<li>[Point 2]</li>
		<li>[Point 3]</li>
	</ul>
```

3.4.2 Commentaires

Pensez à bien renseigner votre code grâce aux commentaires : cela permet de s'y retrouver plusieurs mois après, ou que quelqu'un d'autre puisse comprendre s'il doit reprendre votre travail ! Les commentaires apparaissent uniquement dans le code, et ne sont donc pas visibles par l'utilisateur.

En HTML :

```
< !--- J'ecris mon commentaire --->
```

En CSS :

```
/* J'ecris mon commentaire */
```

3.5 Points de vigilance

3.5.1 Attention aux différents navigateurs

Tous les navigateurs, et leurs différentes versions, ne reconnaissent pas toutes les balises ni propriétés CSS.

Pensez à tester votre page dans différents navigateurs pour voir le résultat : Edge, IE, Chrome, Firefox, Safari, Opera… Bien entendu, vous n'êtes pas obligé de tous les tester, c'est à vous de voir ! Le minimum semble être IE ou Edge + Chrome + Firefox, qui sont les plus utilisés dans l'hexagone.

Pour savoir si une balise est reconnue ou non, vous pouvez aller sur caniuse.com

Si votre propriété n'est pas reconnue par un navigateur, vous devrez probablement utiliser un « **préfixe vendeur** ». Il s'agit d'un préfixe qu'on place devant la propriété voulue. Par exemple, les vieilles versions de Mozilla Firefox ne prenaient pas en compte les bordures arrondies avec `border-radius`. Il fallait alors écrire `-moz-border-radius`, et c'était bon !

Pour tout savoir sur ces préfixes et leur utilisation, consultez l'article dédié sur Alsacreations : https://www.alsacreations.com/article/lire/1159-prefixes-vendeurs-css-proprietaires.html

3.5.2 Le responsive

Ordinateur, tablette, téléphone… Chaque appareil a un écran différent ! Par conséquent, il faut également penser à ce que le site s'adapte au mieux à l'écran : on dit alors que le site est « *responsive* ».

Avoir un site parfaitement *responsive* demande des compétences techniques bien plus pointues que ce qui est abordé ici. Pour rendre votre site le plus fluide possible, utilisez le moins possibles des valeurs fixes (px, cm, etc.) et privilégiez celles qui s'adaptent (%, em, ex, etc.). Vous pouvez également définir des valeurs minimales et maximales pour vos blocs.

3.5.3 HAL corrige

Sachez que si vous faîtes une erreur en modifiant votre code HTML, comme par exemple oublier de délimiter avec une balise fermante, HAL corrigera au mieux de lui-même quand vous enregistrez.

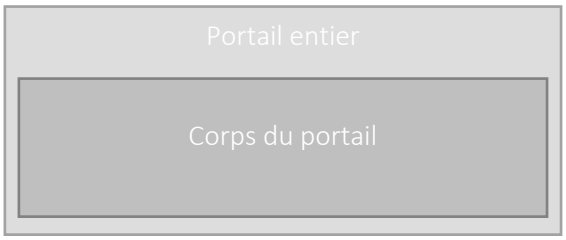
3.5.4 Bien choisir ses balises

Toutes les propriétés de mise en forme CSS ne s'appliquent pas à toutes les balises HTML. Par exemple, si vous voulez créer un bouton avec une image de fond, il vaut mieux mettre votre lien dans une cellule `<td>` ou utiliser `<button>` que de créer une bordure autour d'un lien (le lien `<a>` ne supportant pas background-image).

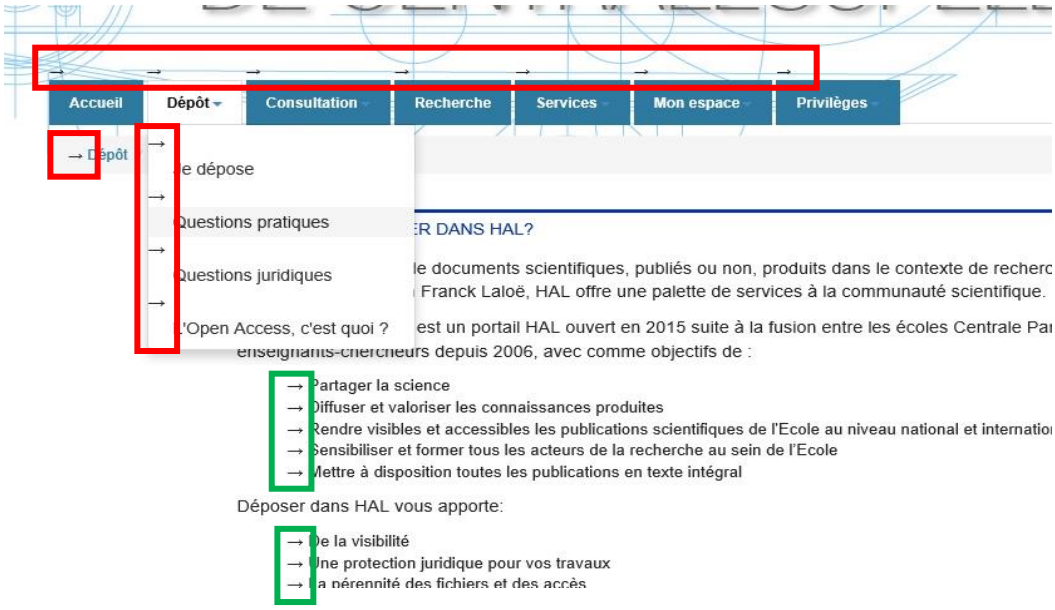
Idem, la propriété `border-spacing` ne s'adapte qu'à `<table>` et non pas aux `<td>`.

3.5.5 Le cadre HAL

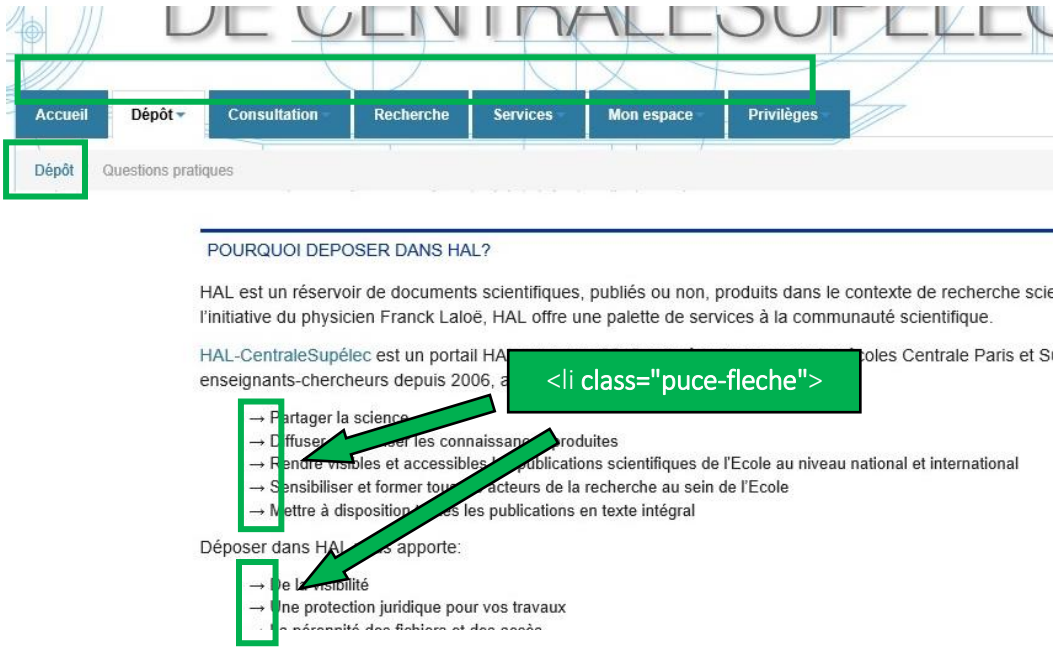
Nous travaillons dans **une partie de HAL.** Il faut prendre en compte le site tout entier (comprendre : les menus de navigation, où nous avons la main pour le CSS mais pas en HTML), car nous pouvons appliquer un nouveau style CSS à des balises déjà utilisées par ailleurs.



L'exemple des puces avec `<li>` est assez flagrant :



En différenciant mes puces du site HAL de mon portail avec une classe (il était aussi possible d'utiliser un sélecteur) :



4 Autres constructions spécifiques

4.1 Ajouter du multimédia

4.1.1 Image

Outre l'image en fond (`background-image`), vous pouvez tout à fait insérer des images sur votre portail, grâce à la balise HTML `<img/>`.

Attributs :

- Src : obligatoire, donne l'emplacement de l'image que l'on souhaite afficher ;
- Alt : recommandé, permet de donner une description textuelle si notre image n'arrive pas à s'afficher
- Width / height : pour définir les dimensions si besoin (en px, mais seuls les chiffres sont indiqués)

Vous pouvez définir des dimensions minimales et maximales pour votre image avec `width-min` et `width-max`, afin que le portail soit un minimum responsive.

```
<img src="adresse_de_l_image.jpg" alt="description de mon image" width="685" height="200" />
```

Pour en savoir + : tutoriel des images responsives
https://developer.mozilla.org/fr/docs/Apprendre/HTML/Comment/Ajouter_des_images_adaptatives_à_une_page_web

4.1.2 Vidéo

On insère une vidéo avec la balise `<iframe>`. Attributs :

- Src : URL (adresse web) ;
- Width/ height : dimensions (en px, mais seuls les chiffres sont indiqués)

```
<p><iframe src="https://www.youtube.com/embed/WnxqoP-c0ZE" width="400" height="220" frameborder="0" allowfullscreen=""></iframe></p>
```


HTML/CSS pour HAL : tutoriel. V.5/7/19. Licence CC : BY-NC-SA. CentraleSupélec - Pôle IST/A.Lustrement

4.2 Faire des boutons

Pour créer des boutons « animés », consultez l'article « Transitions CSS » sur Alsacreations : https://www.alsacreations.com/tuto/lire/874-transitions-css3-proprietes.html

4.2.1 Solution 1 : <button>

Il existe une balise HTML dédiée : `<button>`. Exemple :

Avec `:hover` (curseur placé sur le bouton du milieu):

4.2.2 Solution 2 : <td> + <a>

En plaçant un lien `<a>` dans un paragraphe `<p>` dans votre cellule `<td>`, vous pouvez également créer un bouton.

Avec `:hover` :

4.2.3 Solution 3 : <a>

Il est également possible de personnaliser les liens <a> en leur attribuant des bordures, un fond… (cf. exemple du site doi.org)

4.3 La création d'un fil Twitter

4.3.1 Solution 1 : le widget

HAL propose d'emblée un widget avec deux fonctionnalités :

- Afficher un fil Twitter ;
- Afficher un bouton pour s'abonner sur Twitter ;

Problème du widget **si vous insérez un fil** : la taille est difficilement maîtrisable, notamment en longueur !

Vous pouvez créer vous-même un fil Twitter et le cadre que vous souhaitez, avec les dimensions que vous voulez.

Sachez que vous pouvez créer un flux pour n'importe quel compte Twitter, pas forcément le vôtre.

4.3.2 Solution 2 : créer un encart et insérer un fil Twitter dedans

Etape 1 : **préparer la cellule** `<td>` dans laquelle vous allez placer votre fil. Si vous voulez lui donner une apparence différence par rapport au reste, attribuez-lui un id, comme par exemple `<td id="fil-twitter">`.

Etape 2 : rendez-vous sur https://publish.twitter.com/#

Etape 3 : **Indiquez le compte que vous souhaitez suivre**. Ex : https://twitter.com/HAL_CS_Publis

Etape 4 : choisissez « **embedded timeline** ». *L'autre option consiste à proposer un bouton, ce que fait très bien le widget de HAL, sauf si vous souhaitez le personnaliser*.

Etape 5 : allez dans « **set cusomization options** ». Vous pouvez définir directement les **dimensions de votre fil, son fond et la couleur des liens** (il est possible de mettre la couleur qu'on veut au format hexadecimal).

Etape 6 : cliquez sur **Update**, puis **Copy Code**.

Etape 7 : **collez le lien** dans votre encart dédié.

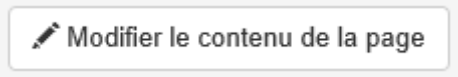
Votre fil est plus long que votre encart ? Si ce n'est pas fait automatique, pensez à `overflow:auto` pour avoir des barres de défilement si besoin.

Le fil est plus petit que votre cellule et votre bordure se met mal ? Insérer le dans une cellule dans votre cellule initiale, sans donner de dimensions à cette nouvelle cellule. Elle s'ajustera, ainsi que sa bordure, à son contenu.

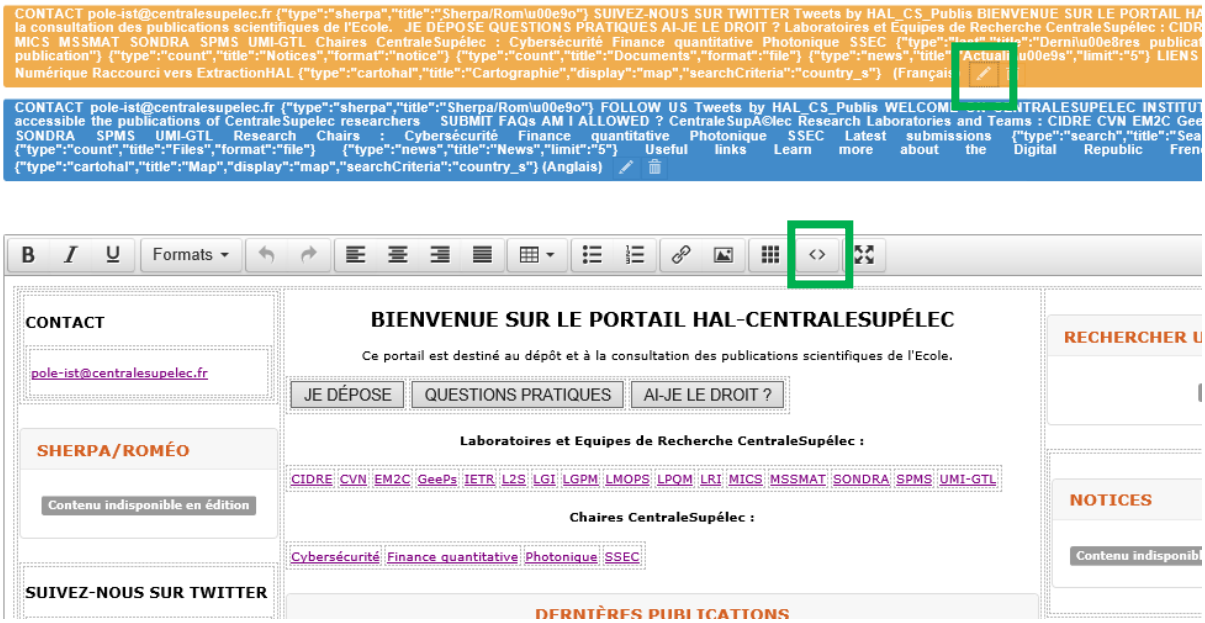
5 Se repérer dans HAL

5.1 Modifier le contenu HTML de la page

En bas de votre page d'accueil, cliquez sur le bouton « Modifier le contenu de la page »



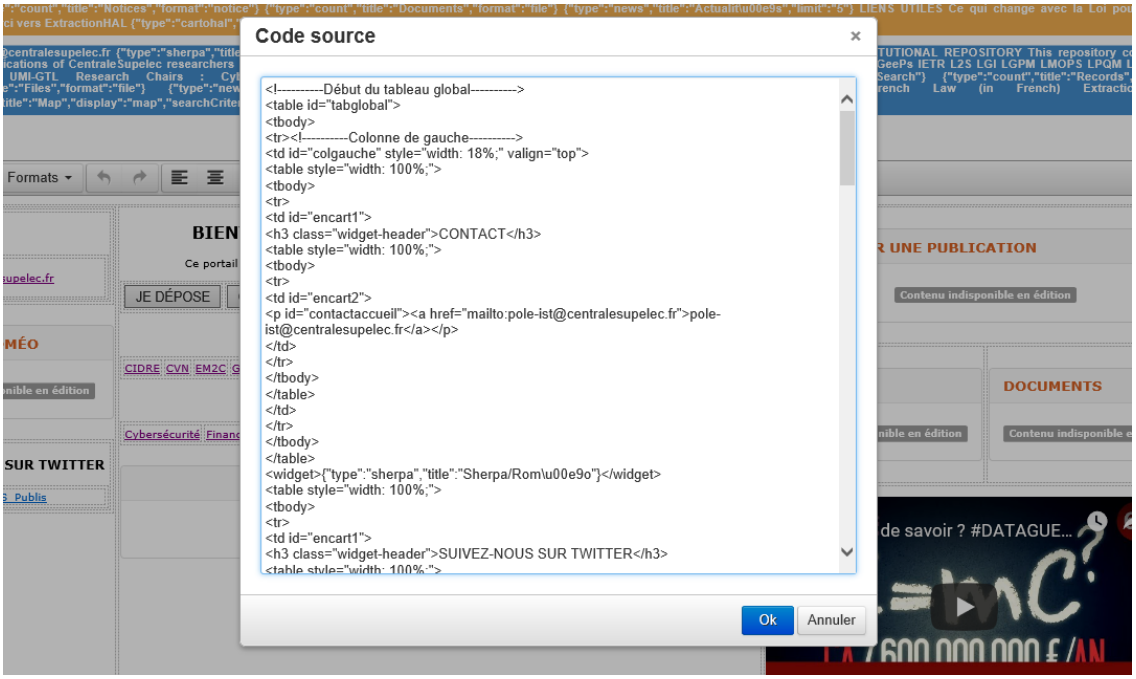
Cliquez ensuite sur le crayon de la version de la page que vous voulez modifier (français, anglais, ou plus si vous traduisez votre portail dans d'autres langues)



C'est ici que vous construisez votre squelette (tableaux, texte…)

HTML/CSS pour HAL : tutoriel. V.5/7/19. Licence CC : BY-NC-SA. CentraleSupélec - Pôle IST/A.Lustrement

Si vous appuyez sur le bouton <>, vous tombez sur le code brut HTML de votre page :

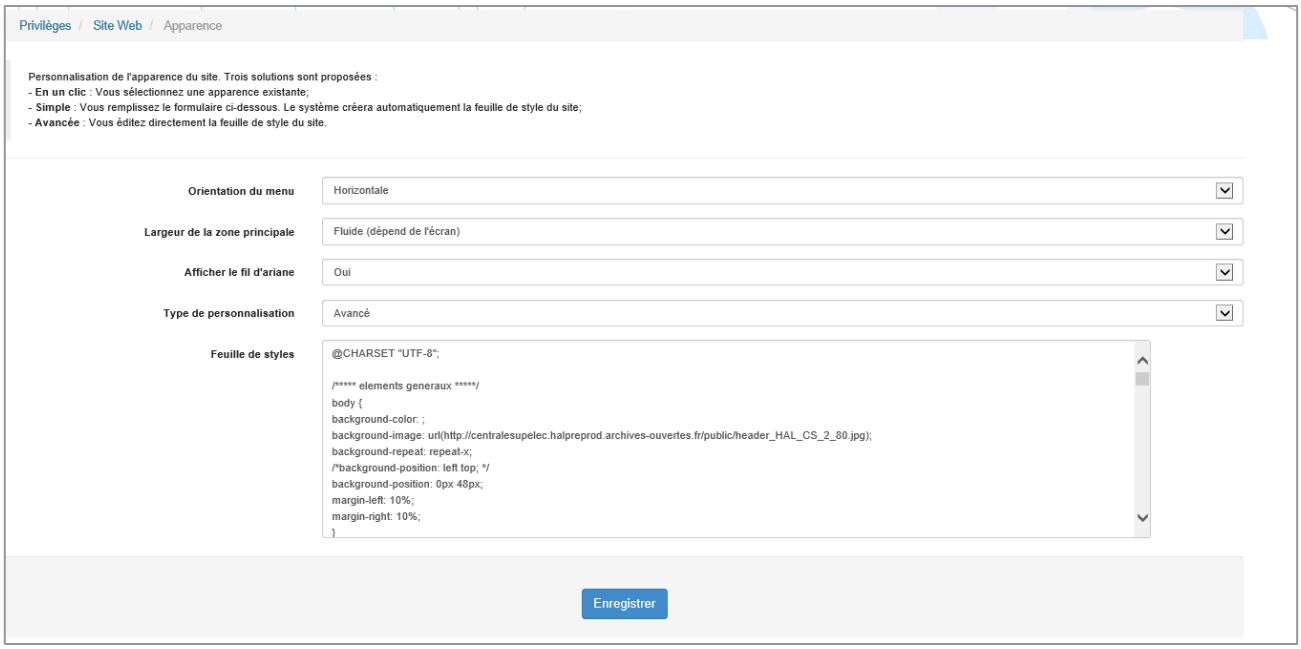


A cet endroit, vous pouvez ajouter des id et class aux balises qui en ont besoin.

C'est également ici que vous donnez l'attribut style="width : **%" pour définir les largeurs de vos colonnes principales.

5.2 Modifier le CSS

Allez dans *Privilèges > Site web > Apparence*



Votre code CSS sera à renseigner *« Feuille de styles »* (Paramètre : *Type de personnalisation > avancé*).

5.3 Déposer des fichiers

Par exemple : photo de l'université, de laboratoires...

Déposez dans *Privilèges > Site web > Ressources*

ATTENTION : Ne jamais supprimer le fichier « style.css » !

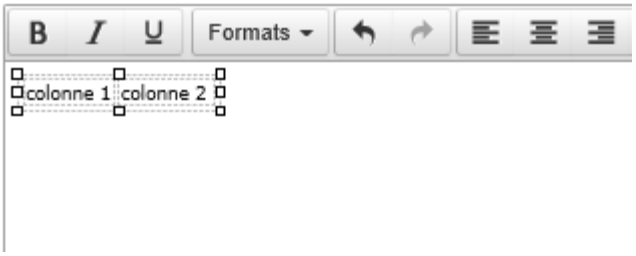
5.4 Ajouter un logo / un bandeau

Déposez votre (vos) image(s) dans *Privilèges > Site web > En-tête*

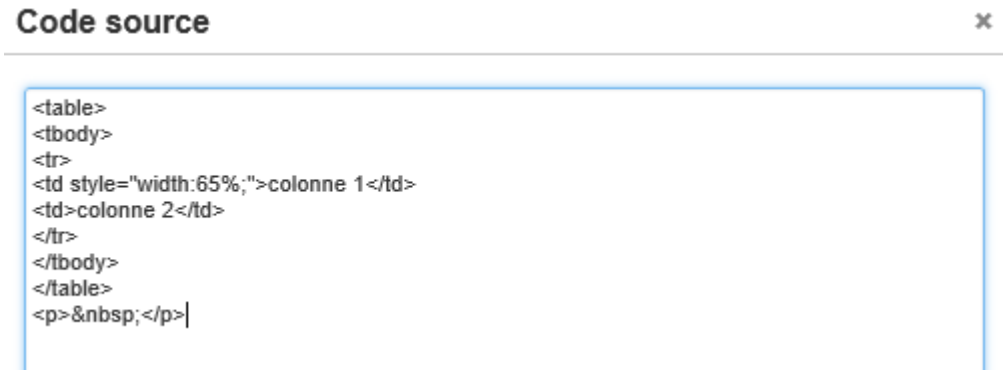
6 Construire son squelette : proposition de trame

1. **Rendez-vous sur votre page d'accueil**

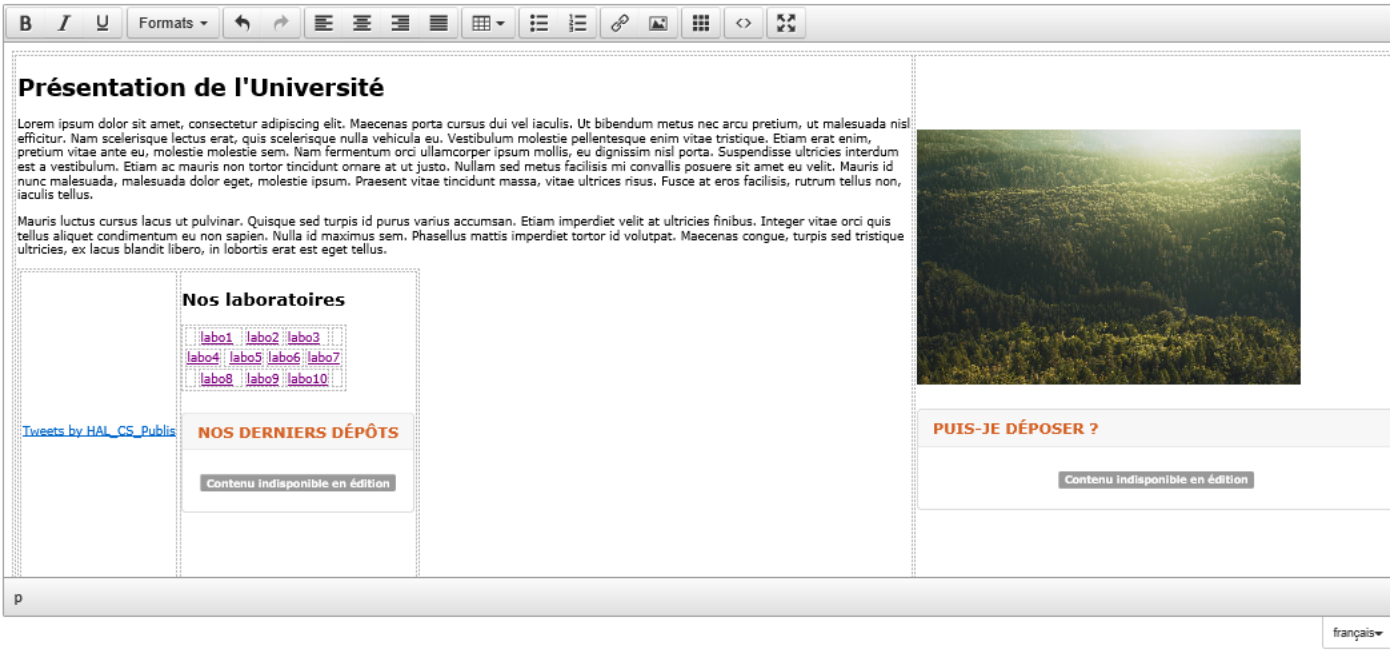
→ Créez un premier tableau d'une seule ligne, avec en nombre de cellules le nombre de colonnes que vous souhaitez (généralement 2 ou 3)



→ Allez dans le code (<>) et indiquer avec style="width=XX%" la largeur que vous souhaitez pour chacune d'entre elles



→ Placez votre texte ou au moins quelques éléments de repère (du même type que « colonne 1 » / « colonne 2 »), vos sous-tableaux, vos images, vos liens, vos widgets, les textes en gras/italiques et titres.



HTML/CSS pour HAL : tutoriel. V.5/7/19. Licence CC : BY-NC-SA. CentraleSupélec - Pôle IST/A.Lustrement

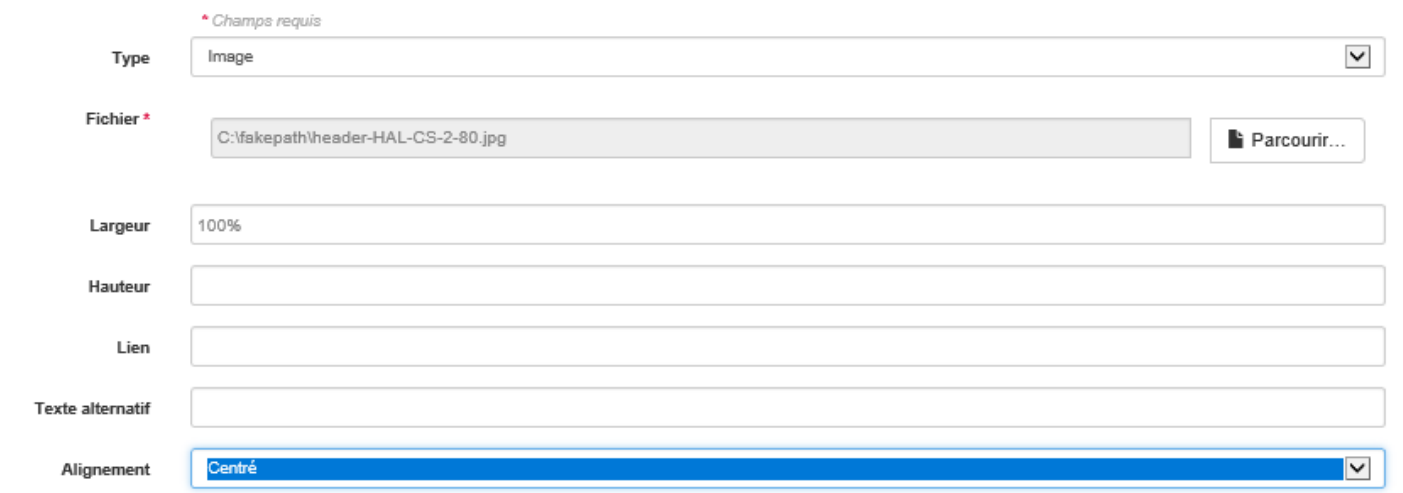
→ Copier votre code dans un éditeur de code pour :
1) le sauvegarder
2) travailler directement dans l'éditeur de code (ex : NotePad++) si vous le souhaitez, car c'est plus agréable visuellement (couleurs, détection des erreurs). Quand vous avez fait des modifications, vous pouvez le recoller le code dans HAL. Vous pouvez aussi ainsi facilement sauvegarder plusieurs versions différentes ou successives pour revenir en arrière si besoin.

Maintenant que votre contenu est prêt (ou presque), nous allons installer le bandeau.

2. Bandeau / Logo

→ Allez dans privilèges > site web > en-tête

→ Déposez votre bandeau, indiquez son éventuelle taille. Vous pouvez mettre des % pour la largeur (notamment s'il dépasse à droite, vous pouvez par exemple mettre 100%)



3. Premier CSS

Ca prend forme ! [Lien pour télécharger un CSS vierge]

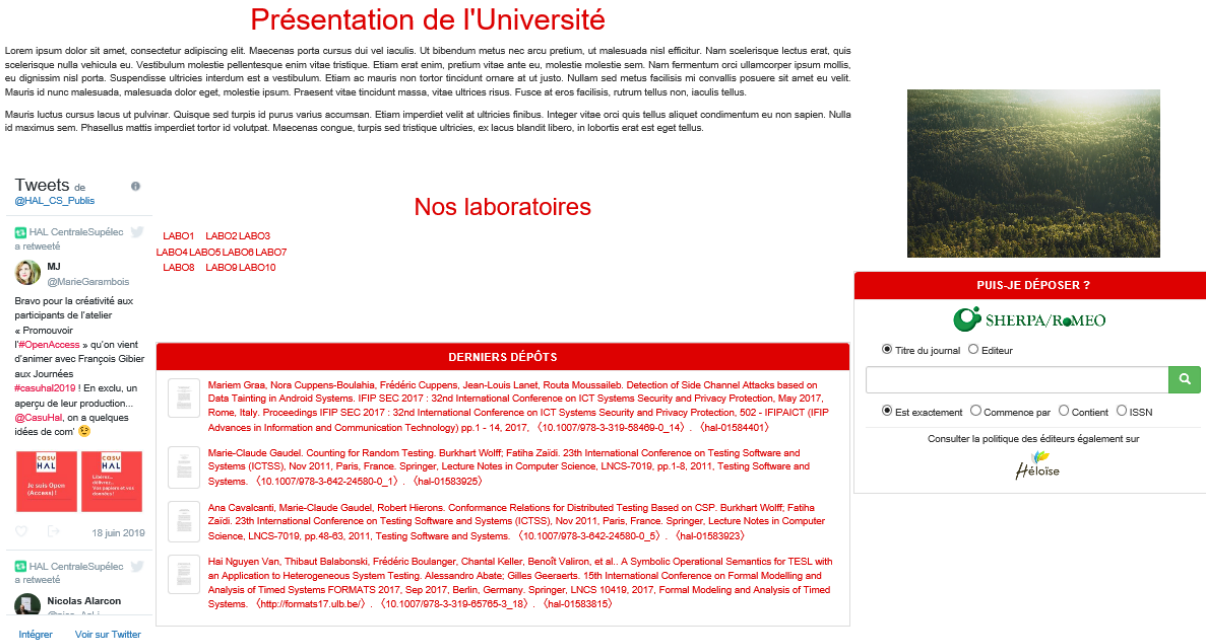
→ Vous pouvez changer le style des éléments les plus connus : liens (a), paragraphes (p)…

→ Mettez également en forme le menu de navigation

```
/***** navigation *****/
.nav-tabs > li > a {
        margin-right: ;
        line-height: ;
        border: ;
        border-radius: ;
        font-weight:
        background-color: ;
        color: ;
}

.nav-tabs > li > a:hover,
.nav-tabs > li > a:focus {
        background-color: ;
        color: ;
        border: ;
}

.nav-tabs > li.active > a,
.nav-tabs > li.active > a:hover,
.nav-tabs > li.active > a:focus {
        color: ;
        cursor: ;
        background-color: ;
        border: ;
        border-bottom-color: ;
}
```



→ C'est le moment de mettre en forme vos tableaux ! Indiquer des bordures et fonds pour voir l'effet que cela fait. Vous pouvez aussi indiquer des tailles…

→ **Notez tous les éléments qui auraient besoin d'être précisés autrement que par leur balise générique** (comprendre : ceux à qui l'on va attribuer des classes et des id)

Dans notre exemple, nous allons préciser :

- la cellule autour du Twitter (pour mettre une bordure) avec un id. Note : le Twitter est dans une cellule DANS la cellule car autrement la bordure rouge suivait mal son contour (elle s'adaptait à la colonne de droite qui contient les labos et les dépôts) ;
- un id au tableau du laboratoire (#tablabo) pour attribuer un border-spacing sans toucher les autres <table> ;
- une classe aux cellules de #tablabo où il y a du contenu, pour attribuer des images et une taille (100px)
- une classe aux cellules de #tablabo où il n'y a PAS de contenu, pour leur attribuer une taille afin que tout se mette bien en quinquonce (50px) ;

4. Préciser les éléments

→ Allez à la modification de page d'accueil, dans le code HTML, ou dans votre éditeur de code

→ Placez vos id et class

```
<h2>Nos laboratoires</h2>
<table id="tablelabo">
<tbody>
<tr>
<td class="cellvide"> </td>
<td class="cell-labo" colspan="2"><a href="https://hal-centralesupelec.archives-ouvertes.fr/">LABO1</a></td>
<td class="cell-labo" colspan="2"><a href="https://hal-centralesupelec.archives-ouvertes.fr/">LABO2</a></td>
<td class="cell-labo" colspan="2"><a href="https://hal-centralesupelec.archives-ouvertes.fr/">LABO3</a></td>
<td class="cellvide"> </td>
</tr>
```

→ Notez bien les noms !

→ Revenez au CSS et indiquez leur mise en forme

```
/*** laboratoires ***/

#tablelabo {
	border-spacing: 4px;
	margin-right: auto;
	margin-left: auto;
}

.cell-labo {
	border-radius: 8px;
	height: 100px;
	width: 100px;
	background : url(SPMS_2.jpg) center;
	text-align: center;
	vertical-align: middle;
	font-size:2em;
	margin: auto;
}

.cellvide {
	width: 50px;
}
```

Faites autant d'allers-retours que nécessaire ! Testez avec les différents navigateurs régulièrement.

Pensez également à sauvegarder votre code HTML et votre code CSS avec l'éditeur de code, au cas où.

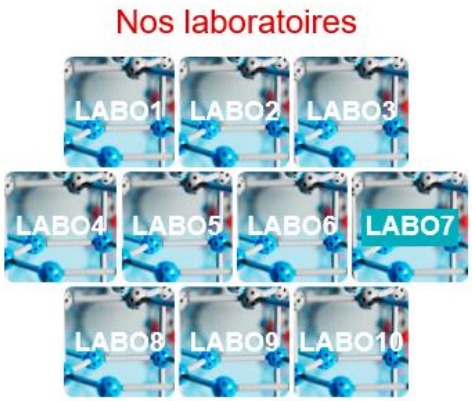
5. Boutons & cie

→ Une fois que vous êtes satisfaits, vous pouvez ajouter des pseudo-classes pour rendre votre site plus vivant !

```
#tablelabo a {
    color:white;
    padding: 3px;
    font-weight: bold;
}

#tablelabo a:hover,
#tablelabo a:focus {
    background-color:#00AEBE;
    text-decoration: none;
}

#tablelabo a:active {
    color:#DD0101;
    text-decoration: none;
}
```



7 Autres applications : HALtools et CV HAL

Maintenant que vous êtes des as du HTML/CSS (!), sachez que vous pouvez également personnaliser des listes des publications de vos laboratoires ou les CV HAL de vos chercheurs et enseignants-chercheurs (encore une fois, il y a des éléments sur lesquels nous n'avons pas la main !).

7.1 HALtools

Cet outil développé par l'INRIA vous permet d'insérer des listes de publications automatiques dans les sites web. Il y a la possibilité d'ajouter une feuille de style CSS personnalisée.

Exemple de CSS pré-défini sur HALtools :

2010

Journal articles

Study of droplet size distribution during an emulsification process using in situ video probe coupled with an automatic image analysis
Abir Khalil, François Puel, Yves Chevalier, Jean-Marc Galvan, Alain Rivoire, Jean-Paul Klein
Journal Chemical Engineering, 2010, 165 (3), pp. 946-957. ⟨10.1016/j.cej.2010.10.031⟩

A synthesis methodology for AMS/RF circuit reliability: Application to a DCO design
Pietro Maris Ferreira, Hervé Petit, Jean-François Naviner
Microelectronics Reliability, Elsevier, 2010, 51 (4), pp.765-772

Adapté aux couleurs CentraleSupélec :

2010

Study of droplet size distribution during an emulsification process using in situ video probe coupled with an automatic image analysis
Abir Khalil, François Puel, Yves Chevalier, Jean-Marc Galvan, Alain Rivoire, Jean-Paul Klein
Journal Chemical Engineering, 2010, 165 (3), pp. 946-957. ⟨10.1016/j.cej.2010.10.031⟩

A synthesis methodology for AMS/RF circuit reliability: Application to a DCO design
Pietro Maris Ferreira, Hervé Petit, Jean-François Naviner
Microelectronics Reliability, Elsevier, 2010, 51 (4), pp.765-772

Comment procéder ?

L'INRIA met à disposition d'autres feuilles CSS et une feuille de style générique, à l'adresse : https://iww.inria.fr/hal/aide/spip.php%3Farticle152&lang=fr.html

Pour reconnaître les balises à modifier en CSS, allez sur https://haltools.archives-ouvertes.fr > Créer sa page web et remplissez les champs qui vous concernent.

Cliquez ensuite sur le lien rouge généré, qui vous ouvrira une nouvelle page.

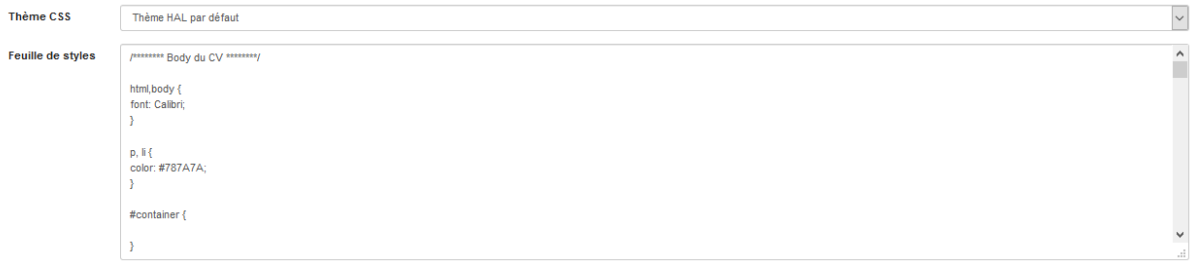
Faîtes clic droit et « afficher la source » (IE) / « afficher le code source de la page » (Chrome) / « code source de la page » (Firefox) : la structure HTML va apparaître, et vous pourrez comparer avec votre feuille CSS.

Une fois votre feuille CSS prête, déposez-la en ligne (sur un site dont l'url doit commencer par https://) et indiquez son adresse dans « Ou css privé ».

7.2 CV HAL

Note : la manipulation est également expliquée ici : https://doc.archives-ouvertes.fr/identifiant-auteur-idhal-cv/personnaliser-lapparence-de-son-cv/

Etape 1 : **choisissez un thème CSS**. Ce qui est important ici est de choisir où vous souhaitez que les widgets apparaissent (colonne de gauche ou de droite).



Etape 2 : **Récupérez la feuille CSS vierge** à l'adresse https://github.com/CCSDForge/HAL/blob/master/CVThemes/cv-tpl.css et **complétez ce qui vous intéresse.**

Pour savoir quelles balises correspondent à quels élements, vous pouvez faire comme pour HALtools : clic droit et « afficher la source » (IE) : la structure HTML va apparaître, et comparez avec votre feuille CSS.

Vous pouvez supprimer les éléments qui ne vous intéressent pas pour alléger la feuille, ou même simplement copier dès le début dans HAL celles qui vous intéressent uniquement.

8 Documents et liens utiles

Memo balises HTML et CSS :
Lien Sharepoint
Cours HTML/CSS d'OpenClassroom
Apprenez à créer votre site web avec HTML5 et CSS3
Pour se former sur des points précis
Alsacreations.com
MDN web docs (Mozilla)
W3schools (en anglais)
Connaître les spécificités des navigateurs
Caniuse.com
Pour récupérer les couleurs
Code-couleur.com

HTML/CSS pour HAL : tutoriel. V.5/7/19. Licence CC : BY-NC-SA. CentraleSupélec - Pôle IST/A.Lustrement