# Création de tableaux HTML

Rédigé par: Jean-Luc Colson

Date première rédaction: Janvier 2020

SUIVI DES MODIFICATIONS A LA FICHE

| Série | Date | Page(s) modifiée(s) | Raison |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

# Contents

Créer des tableaux avec l'élément HTML table ..... 4
Définition et utilité des tableaux HTML ..... 4
Les éléments constitutifs essentiels d'un tableau HTML ..... 4
Créer un tableau simple en HTML ..... 5
Structurer un tableau HTML ..... 8
Ajouter une ligne d'en-tête à un tableau HTML ..... 8
Création d'un tableau structuré en HTML ..... 9
Fusionner des cellules entre elles avec colspan et rowspan ..... 10
Ajouter une légende à un tableau HTML avec caption ..... 11
Utiliser les éléments col et colgroup pour préparer la mise en forme des colonnes d'un tableau ..... 12
Mettre en forme un tableau HTML avec du CSS ..... 14
Mettre en forme un tableau : attributs HTML vs CSS ? ..... 14
Les propriétés border-collapse et border-spacing ..... 15
Contrôler la largeur des colonnes avec table-layout ..... 18
Définir l'affichage des cellules vides d'un tableau avec empty-cells ..... 19
Choisir où s'affiche la légende d'un tableau avec caption-side ..... 20
Aligner le contenu des cellules d'un tableau avec vertical-align ..... 21

# Créer des tableaux avec l'élément HTML table

Un tableau en HTML représente un ensemble organisé de données. Pour créer un tableau en HTML nous allons utiliser l'élément table qui signifie « tableau » en anglais.

Les tableaux sont une notion importante du HTML et il est important que vous sachiez comment les créer. Dans cette leçon, nous allons voir comment créer un tableau simple.

## Définition et utilité des tableaux HTML

Les tableaux en HTML vont nous permettre de présenter des données de manière organisée et sous une certaine forme pour les structurer et les rendre compréhensibles pour les navigateurs, moteurs de recherche et utilisateurs.

Historiquement, de nombreux développeurs et web designer se sont servis à tort des tableaux pour mettre en forme de pages web. En effet, à l'époque, utiliser des tableaux semblait être un bon moyen de contrôler l'affichage et le placement des différents éléments d'une page en les plaçant dans des cellules.

Dans cette lignée, de nombreux attributs HTML permettant de modifier / mettre en forme un tableau avaient été créés.

Cependant, cette utilisation est à bannir aujourd'hui. En effet, je vous rappelle que le rôle du HTML est et a toujours été de structurer du contenu et de lui donner du sens, pas de mettre en forme ledit contenu (ce qui est le rôle du CSS).

Ainsi, vous comprendrez également que pour mettre en forme visuellement un tableau, nous n'utiliserons pas les attributs HTML (qui sont en grande majorité dépréciés) mais plutôt des propriétés CSS.

## Les éléments constitutifs essentiels d'un tableau HTML

Un tableau est un ensemble de lignes et de colonnes. L'intersection entre une ligne et une colonne est une cellule de tableau.

Pour créer un tableau fonctionnel en HTML, nous allons devoir utiliser à minima 3 éléments :

- Un élément table (« tableau » en français) qui va définir le tableau en soi ;
- Des éléments tr, pour « table row » ou « ligne de tableau » en français qui vont nous permettre d'ajouter des lignes dans notre tableau ;
- Des éléments td, pour « table data » ou « donnée de tableau » en français qui vont nous permettre d'ajouter des cellules dans nos lignes et ainsi de créer automatiquement de nouvelles colonnes.

L'élément HTML table va représenter le tableau en soi. Cet élément est composé d'une paire de balises ouvrante <table> et fermante </table> au sein desquelles nous allons placer les différents autres éléments de notre tableau. Les éléments tr et td sont également représentés sous la forme d'une paire de balises avec leur contenu entre les balises.

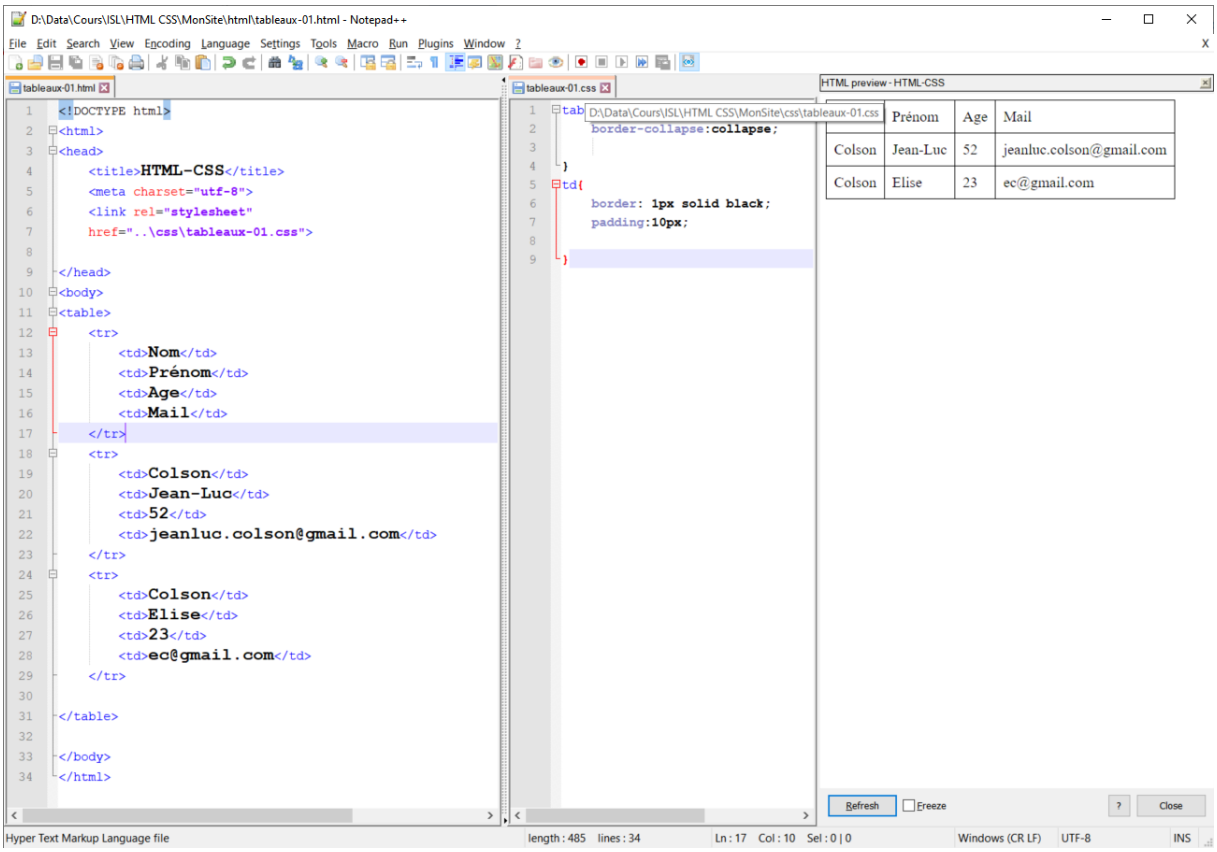
## Créer un tableau simple en HTML

Pour créer un tableau en HTML, il y a une chose que vous devez bien comprendre qui est que les tableaux HTML vont être créés ligne par ligne.

A chaque fois que nous voudrons ajouter une ligne dans notre tableau, nous utiliserons un nouvel élément tr.

Nous allons ensuite pouvoir ajouter autant d'éléments td au sein de chacune de nos lignes. Chaque élément td va représenter une cellule dans le tableau.

Par exemple, pour créer un tableau en HTML contenant 3 lignes contenant chacune 4 cellules (c'est-à-dire un tableau de 3 lignes, 4 colonnes), nous utiliserons notre élément table qui va contenir 3 éléments tr et chaque élément tr contiendra 4 éléments td comme ceci (ne tenez pas compte de la mise en forme pour le moment, nous en reparlerons plus tard) :



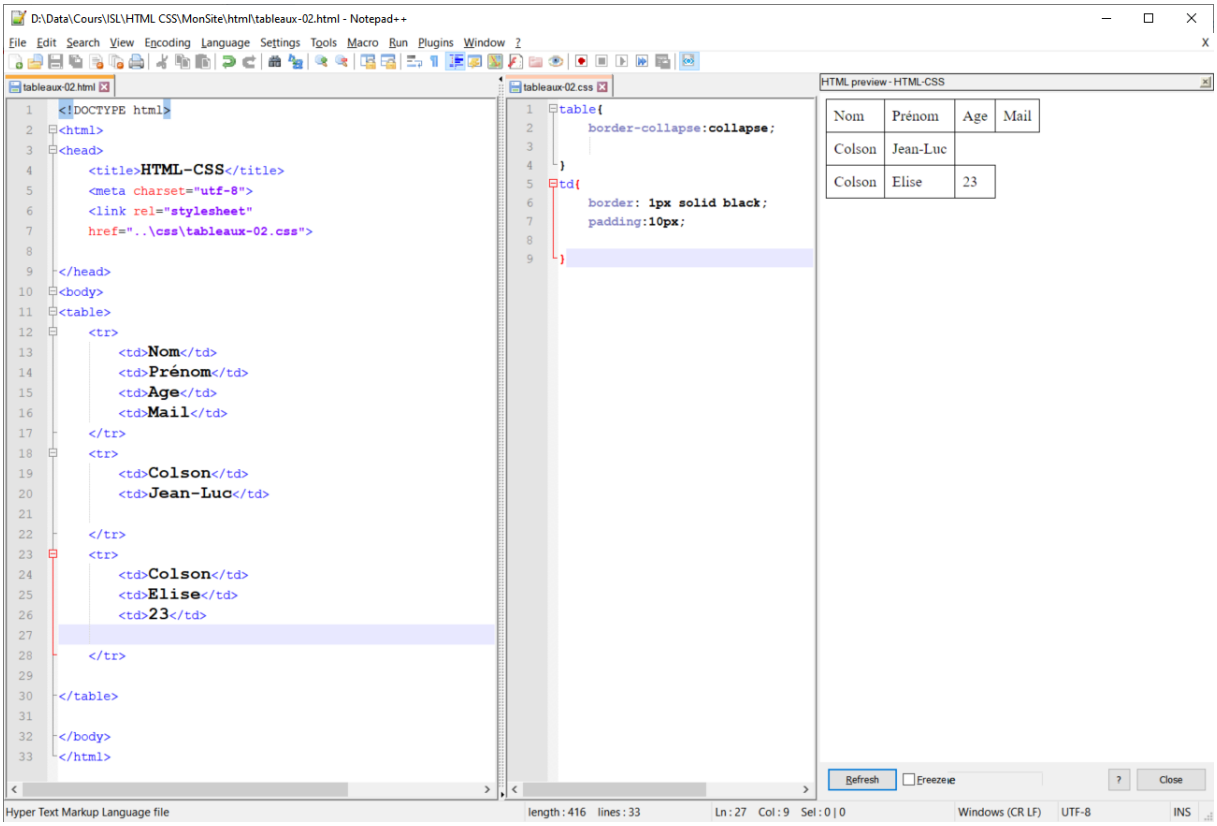
Nous venons ci-dessus de créer notre premier tableau entièrement en HTML. Retenez bien la syntaxe et particulièrement l'ordre d'imbrication des éléments avec nos éléments td à l'intérieur de nos éléments tr : on va utiliser un nouvel élément td dès que l'on va vouloir ajouter une cellule (et donc une colonne) au sein d'une même ligne.

Si vous essayez de ne recopier que le code HTML présenté ci-dessus et de l'afficher, vous risquez d'avoir un objet qui ne ressemble pas à l'idée que l'on se fait d'un tableau. C'est tout à fait normal puisque j'ai déjà ici mis sommairement notre premier tableau HTML en forme en utilisant des propriétés CSS.

Retenez cependant qu'au sens du HTML, et même sans mise en forme CSS, l'objet créé est bien un tableau. J'attire votre attention ici sur le fait que le rôle du HTML n'est que de structurer du contenu, c'est-à-dire le rendre intelligible pour les navigateurs et moteurs de recherche.

Notez par ailleurs que différentes lignes peuvent contenir un nombre différent de cellules même si cela est considéré comme une mauvaise pratique pour des raisons évidentes de sémantique et de lisibilité. Dans le cas où cela arriverait, la largeur de notre tableau HTML (i.e son nombre de colonnes) serait égale au nombre de cellules de la ligne contenant le plus de cellules.

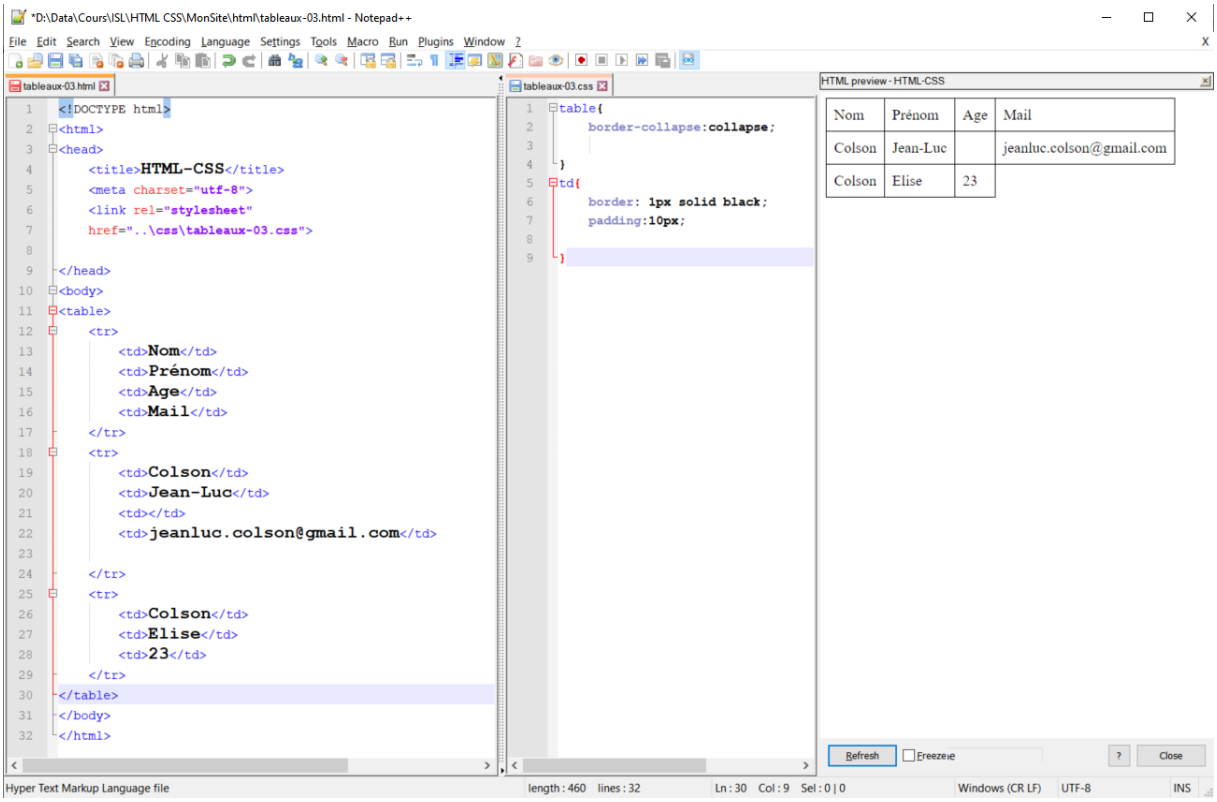
Regardez plutôt l'exemple suivant où on a créé un tableau HTML de 3 lignes avec une première ligne contenant 4 cellules, une deuxième ligne contenant seulement 2 cellules, et la dernière ligne contenant 3 cellules :




Encore une fois, retenez qu'il est considéré comme une mauvaise pratique de créer des lignes de tableau avec un nombre de cellules différent dans chacune. En pratique, nous attribuerons toujours

un même nombre de td à nos différentes lignes. Si on souhaite laisser une cellule vide, nous nous contenterons d'écrire <td> </td> sans rien écrire entre les balises.

Imaginons par exemple que nous voulions laisser la cellule « age » vide pour la ligne relative à « Jean-Luc » Pour cela, il suffit de créer un élément td pour son âge et de ne mettre aucun contenu dedans.



# Structurer un tableau HTML

Dans la leçon précédente, nous avons créé un premier tableau très simple. Ici, je vous rappelle que les tableaux sont des objets HTML qui servent à structurer et à donner du sens à des contenus en les organisant.

Souvent, lorsque nous choisirons d'organiser de l'information dans un tableau, cette information sera hiérarchisée et on voudra que notre tableau possède une ligne d'en-tête dans laquelle on va préciser le type de données attendues dans chaque colonne voire une légende, etc.

Dans cette nouvelle leçon, nous allons découvrir de nouveaux éléments et attributs qui vont nous servir à structurer nos tableaux HTML pour donner un maximum de sens à chaque donnée.

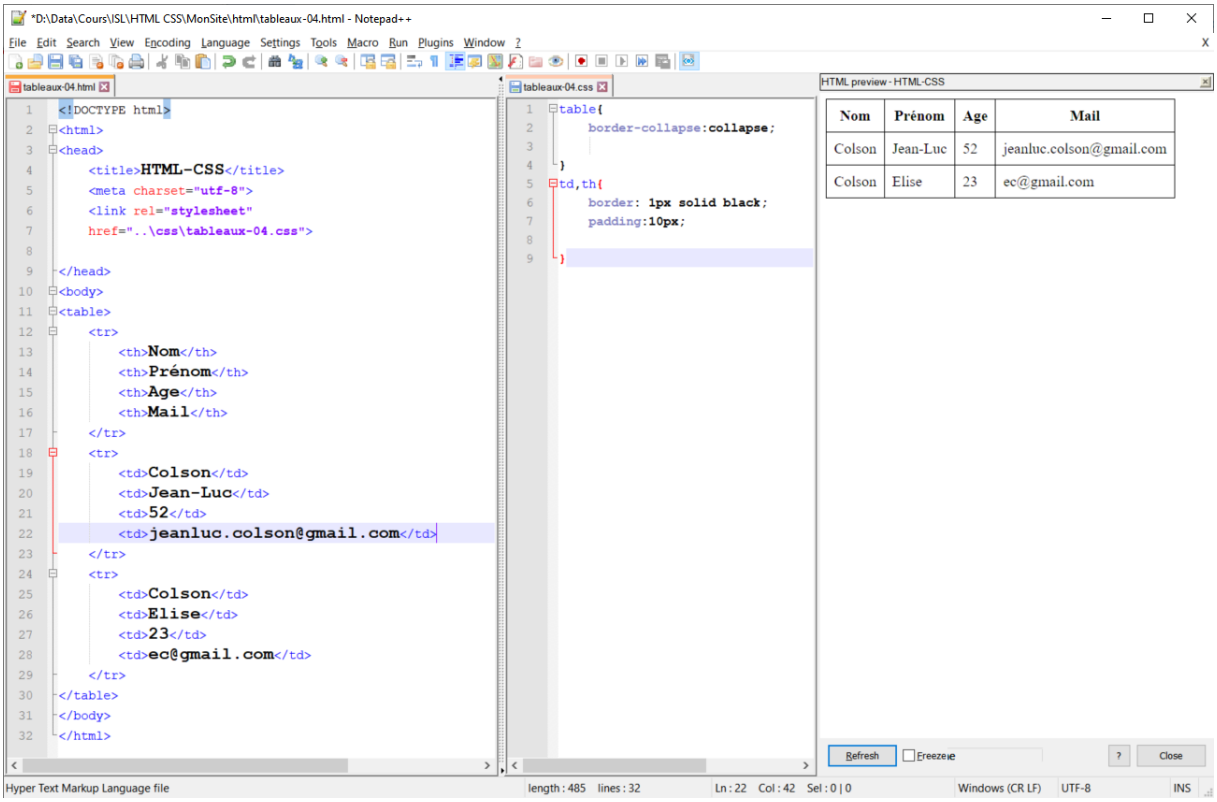
## Ajouter une ligne d'en-tête à un tableau HTML

Très souvent, les tableaux vont posséder une ligne d'en-tête dans laquelle on va donner des informations au lecteur sur le type des données qui seront renseignées dans chaque colonne.

Cette ligne d'en-tête est différente des autres lignes puisqu'elle ne contient pas le même type d'informations que les autres lignes de notre tableau : elle sert simplement à indiquer quelles informations vont figurer dans le tableau. Nous allons ainsi pouvoir différencier cette ligne des autres en HTML à l'aide d'un élément particulier.

Pour créer une ligne d'en-tête en HTML, nous allons cette fois-ci devoir utiliser l'élément th, pour « table head » ou « en-tête du tableau » en français à la place de nos éléments td dans notre première ligne.

On peut appliquer un style particulier en CSS à cette ligne afin de bien la démarquer des autres. Par défaut, le texte au sein des éléments th s'affichera en gras dans la plupart des navigateurs.




Si vous avez ajouté des bordures autour de chaque cellule de votre tableau en CSS, pensez bien ici à appliquer ces mêmes bordures aux éléments th en plus des éléments td pour avoir des bordures autour des cellules de la ligne d'en-tête également.

## Création d'un tableau structuré en HTML

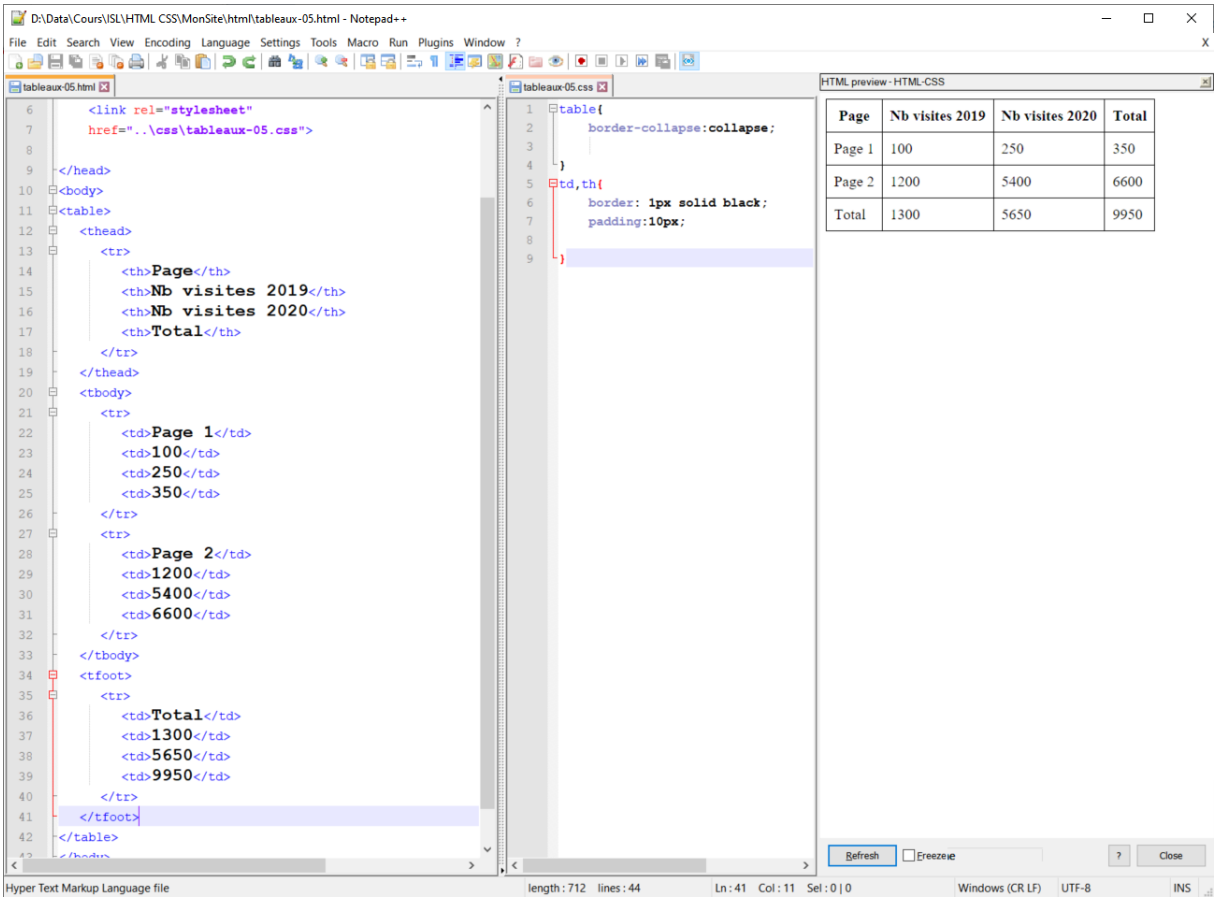
Si vous devez créer des tableaux assez longs, il peut être judicieux de commencer à les structurer en les subdivisant en plusieurs parties.

On peut distinguer trois parties dans un tableau : le haut du tableau contenant généralement la ligne d'en tête, le corps du tableau contenant les informations de notre tableau en soi et le pied du tableau servant à calculer des totaux ou à rappeler les informations d'en tête si votre tableau est vraiment long.

Pour définir chaque partie d'un tableau, nous disposons d'un élément HTML spécifique :

- thead pour la tête du tableau ;
- tbody pour le corps du tableau ;
- tfoot pour le pied du tableau.

Voyons immédiatement comment utiliser ces nouveaux éléments judicieusement.



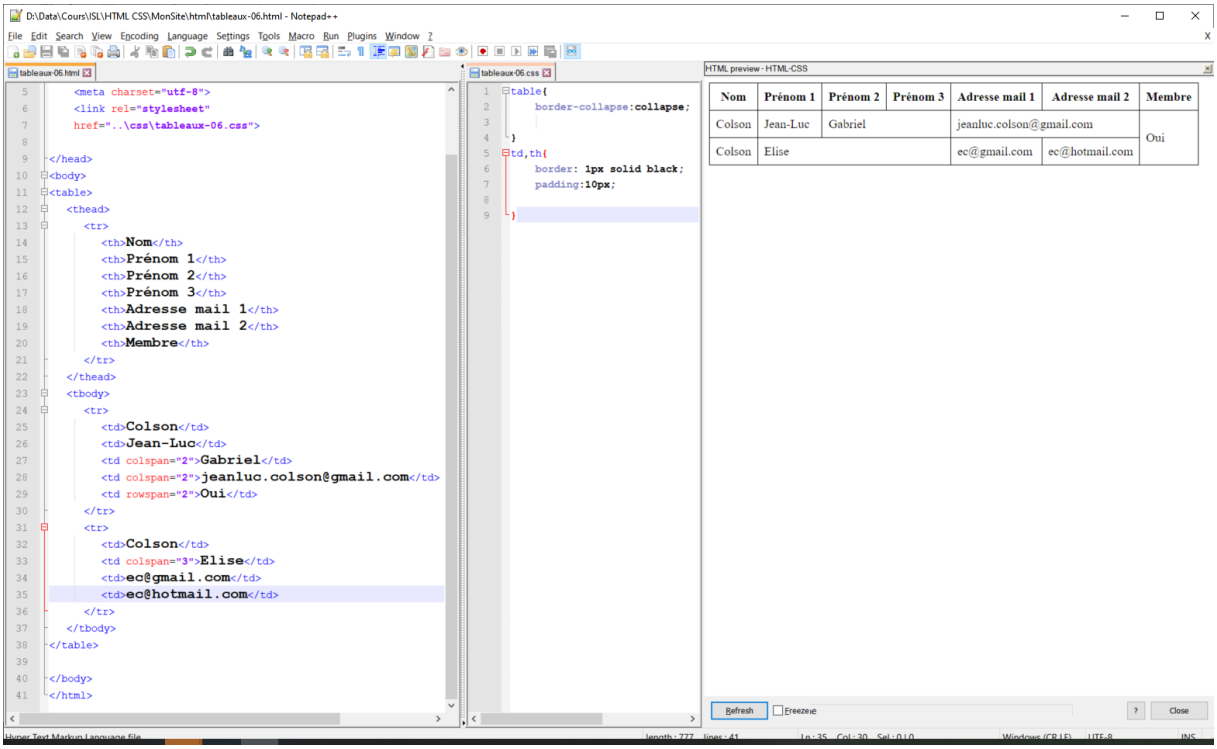

Vous pouvez par ailleurs noter ici une chose intéressante : il est strictement équivalent de mentionner un élément tfoot avant ou après un élément tbody. En effet, même si le tfoot est déclaré avant le tbody, son contenu s'affichera tout de même après dans le rendu final.

Cette curiosité nous vient du HTML4 dans lequel il était interdit de déclarer un élément tfoot après un élément tbody ; il fallait absolument le mentionner avant. Cette règle totalement contre intuitive a été supprimée avec le HTML5.

## Fusionner des cellules entre elles avec colspan et rowspan

Le CSS, bien que puissant aujourd'hui, ne nous permet pas encore de tout faire et certains attributs HTML ne sont pas encore dépréciés. C'est notamment le cas des attributs colspan et rowspan qui vont nous permettre de fusionner plusieurs cellules adjacentes d'une même ligne ou d'une même colonne.

*Ces attributs vont prendre en valeur le nombre de colonnes ou de lignes qu'une cellule doit couvrir, c'est-à-dire le nombre de colonnes ou de lignes qu'une cellule doit occuper ou encore sur lesquelles elle doit s'étendre*. Essayons donc de fusionner plusieurs cellules entre elles dans un tableau HTML grâce à ces attributs :

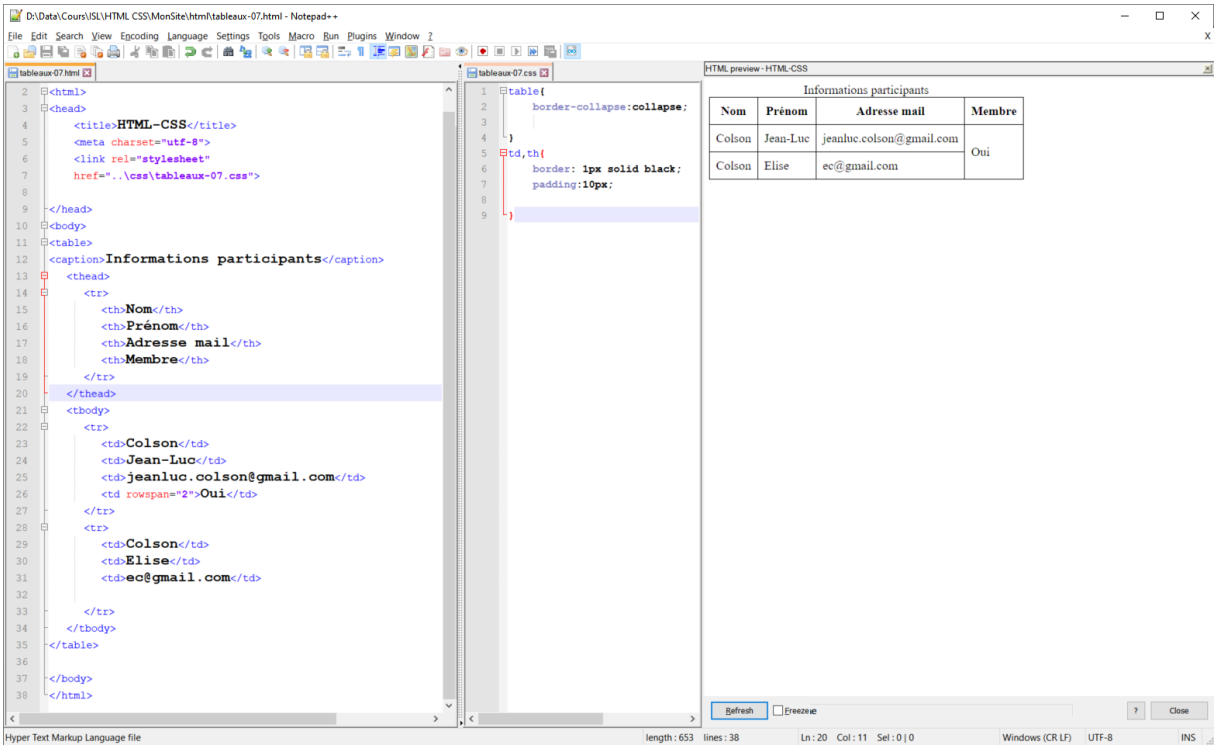



Notez bien ici qu'il faudra bien réfléchir à la construction de votre tableau lorsque vous utilisez ces attributs. Par exemple, si on définit qu'une cellule doit couvrir 3 lignes, il faudra omettre un élément td pour les deux lignes supplémentaires sur lesquelles la cellule s'étend et idem. pour une cellule couvrant l'espace relatif à plusieurs colonnes.

## Ajouter une légende à un tableau HTML avec caption

On va également pouvoir ajouter un titre ou une légende à notre tableau afin d'indiquer ce que contient notre tableau.

Pour ajouter une légende, nous allons utiliser l'élément HTML caption. Cet élément est très simple d'utilisation, mais il faut respecter une règle : il doit être inséré immédiatement après la balise ouvrante de l'élément table.




## Utiliser les éléments col et colgroup pour préparer la mise en forme des colonnes d'un tableau

Nous avons vu précédemment qu'en HTML nous créons les tableaux ligne par ligne avec des éléments tr. Cela fait qu'il est très simple de mettre en forme en CSS les différentes lignes d'un tableau HTML, en utilisant pas exemple une pseudo-classe :nth-child() que nous étudierons plus tard dans ce cours.

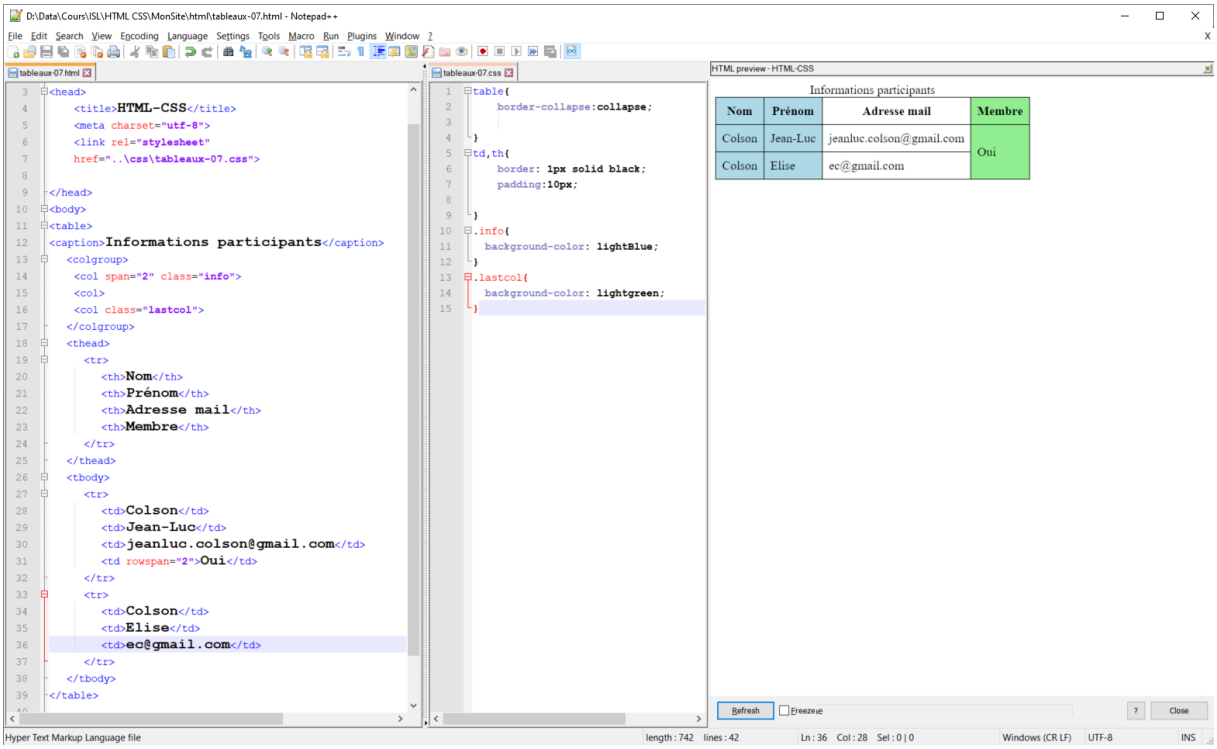
Cependant, cela est beaucoup plus compliqué lorsque l'on souhaite mettre en forme une colonne d'un tableau HTML puisque les cellules représentées par nos éléments td qui créent automatiquement ces colonnes sont dispersées au sein des différentes lignes.

Pour régler ce problème de mise en forme, le HTML nous offre deux éléments qu'on va pouvoir intégrer dans nos tableaux : les éléments col et colgroup.

L'élément colgroup va représenter un groupe d'une ou plusieurs colonnes dans un tableau qu'on va pouvoir ensuite mettre en forme en CSS. *Cet élément est représenté par une paire de balises et doit être un enfant direct de l'élément table, précédé éventuellement uniquement par un élément caption.* Cela signifie qu'il doit être inséré avant tout élément thead, tbody, tfoot, and tr au sein de l'élément table.

Au sein d'un élément colgroup, nous allons pouvoir insérer autant d'éléments col qu'il y a de colonnes dans notre tableau. Chaque élément col représente une colonne de notre tableau. Cet élément ne sert également qu'à la mise en forme et est représenté par une unique balise orpheline <col>.

Pour représenter plusieurs colonnes de notre tableau avec un seul élément col, on peut utiliser un attribut span qui va prendre en valeur le nombre de colonnes de notre tableau que l'élément col doit représenter.




Ici, notre tableau possède quatre colonnes. On veut appliquer une couleur de fond pour toutes les cellules des trois premières colonnes d'un coup.

On utilise donc un élément colgroup dans lequel on place deux éléments col. On passe un attribut span ="3" à notre premier élément col afin que cet élément représente les trois premières colonnes de notre tableau.

Nous n'avons alors plus qu'à appliquer une couleur de fond à cet élément col pour ajouter un fond à chacune des cellules des trois premières colonnes de notre tableau. Pour le cibler facilement, on lui attribue donc une class. Il est aussi possible de définir une autre couleur de fond pour une autre colonne.

# Mettre en forme un tableau HTML avec du CSS

Les tableaux sont des entités très particulières en HTML et des propriétés CSS ont donc été créées spécialement pour les mettre en forme et les personnaliser.

Dans cette leçon, nous allons découvrir ces différentes propriétés et apprendre à les utiliser.

## Mettre en forme un tableau : attributs HTML vs CSS ?

S'il est vrai que le HTML sert à structurer le contenu et à lui donner du sens, nous créons des sites pour des visiteurs humains et donc nous voulons que ces sites soient compréhensibles pour eux et agréables à consulter.

Pour cela, nous devons généralement mettre en forme notre contenu HTML brut afin de mieux le présenter. Pour les tableaux, comme pour la majorité des éléments HTML, nous utilisions à l'époque des attributs de mise en forme HTML.

Les attributs les plus communs utilisés avec les tableaux étaient les suivants :

- align pour définir l'alignement du tableau dans la page ;
- bgcolor pour définir la couleur de fond du tableau ;
- border pour définir la taille d'une bordure autour du tableau ;
- cellpadding pour définir l'espace entre le contenu d'une cellule et sa bordure ;
- cellspacing pour définir l'espace entre deux cellules ;
- char pour aligner le contenu au sein d'une ligne à partir d'un certain caractère ;
- charoff pour définir le nombre de caractères à partir desquels le contenu doit être aligné depuis le caractère défini par l'attribut char ;
- frame pour définir les côtés du tableau sur lesquels tracer une bordure ;
- rules pour définir la manière dont les traits doivent apparaître au sein du tableau ;
- summary pour définir un texte servant à résumer le tableau au cas où celui-ci ne pourrait pas être affiché ;
- valign pour aligner verticalement le contenu des cellules au sein d'une ligne.

Ces attributs sont tous néanmoins tous dépréciés et vous ne devriez jamais les utiliser pour mettre en forme vos tableaux HTML. A la place, nous utiliserons plutôt le CSS qui nous offre aujourd'hui de très bonnes alternatives et un bon support pour la mise en forme de nos tableaux.

Ainsi, nous allons pouvoir par exemple ajouter des bordures autour de chaque cellule de notre tableau en appliquant la propriété border à nos éléments td puis fusionner les différentes bordures

en une pour harmoniser l'ensemble en appliquant un border-collapse: collapse à notre élément table.

Nous allons encore pouvoir modifier la couleur de fond de notre tableau ou d'un élément particulier de notre tableau ou aligner son contenu en CSS.

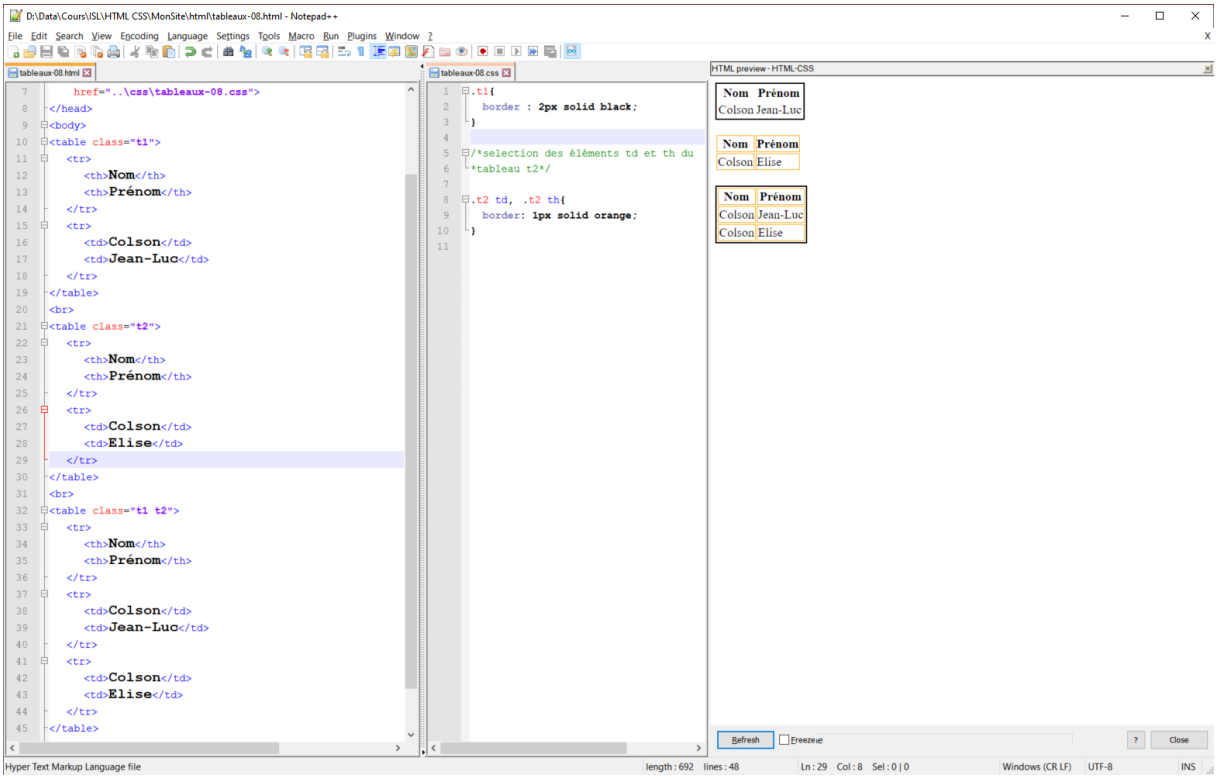
Dans cette leçon, nous allons voir les propriétés CSS suivantes liées aux tableaux :

- border-collapse ;
- border-spacing ;
- caption-side ;
- empty-cells ;
- table-layout ;
- vertical-align.

## Les propriétés border-collapse et border-spacing

La création de bordures dans les tableaux semble complexe à priori. En effet, en appliquant une propriété border à notre élément table, nous allons créer une bordure autour de notre tableau mais pas entre chaque cellule de celui-ci.

Au contraire, en appliquant une bordure à chaque élément th ou td, c'est-à-dire à chaque cellule de notre tableau, les bordures autour de chaque cellule ne vont pas se coller entre elles, ce qui n'est généralement pas le comportement voulu.

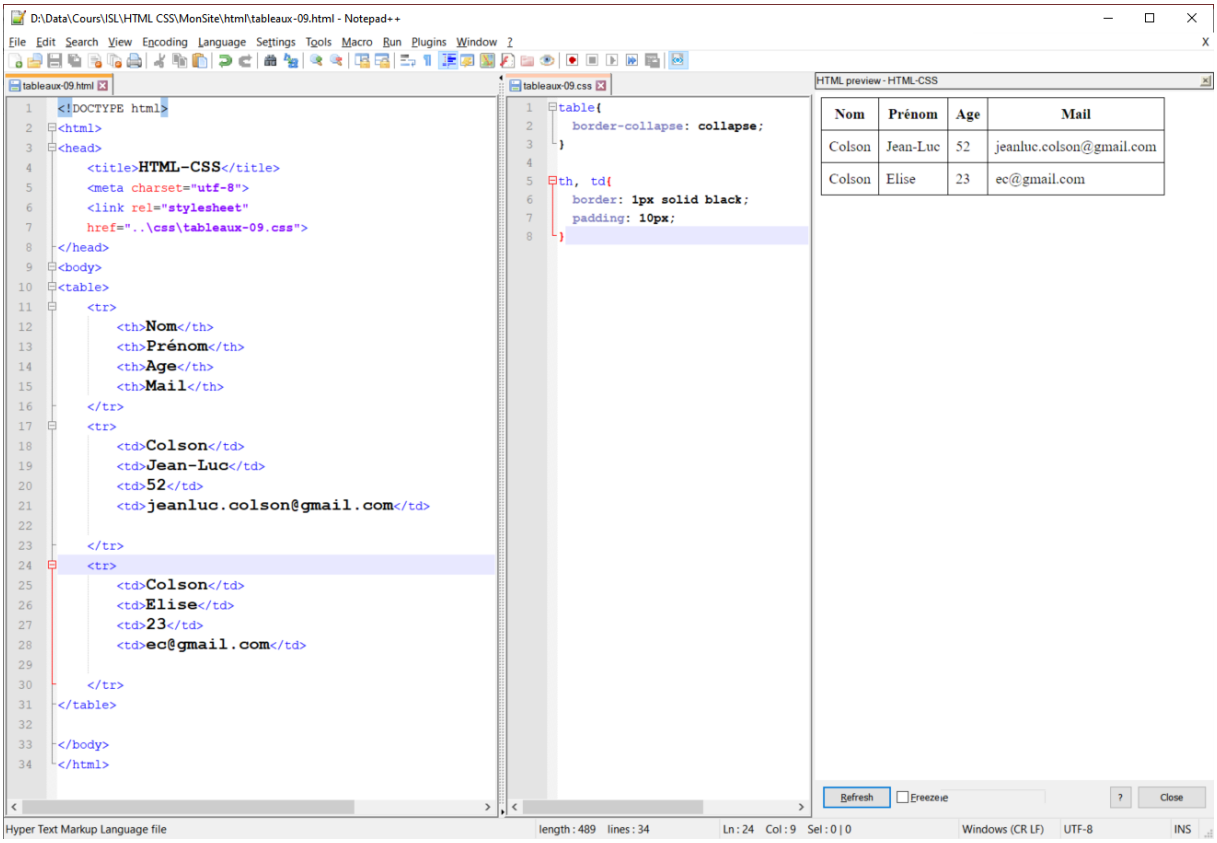



Les propriétés border-collapse et border-spacing ont été créées pour répondre à ce problème.

La propriété border-collapse va nous permettre de choisir si on veut faire fusionner les bordures des différentes cellules de notre tableau ou pas. On va pouvoir lui passer l'une de ces deux valeurs :

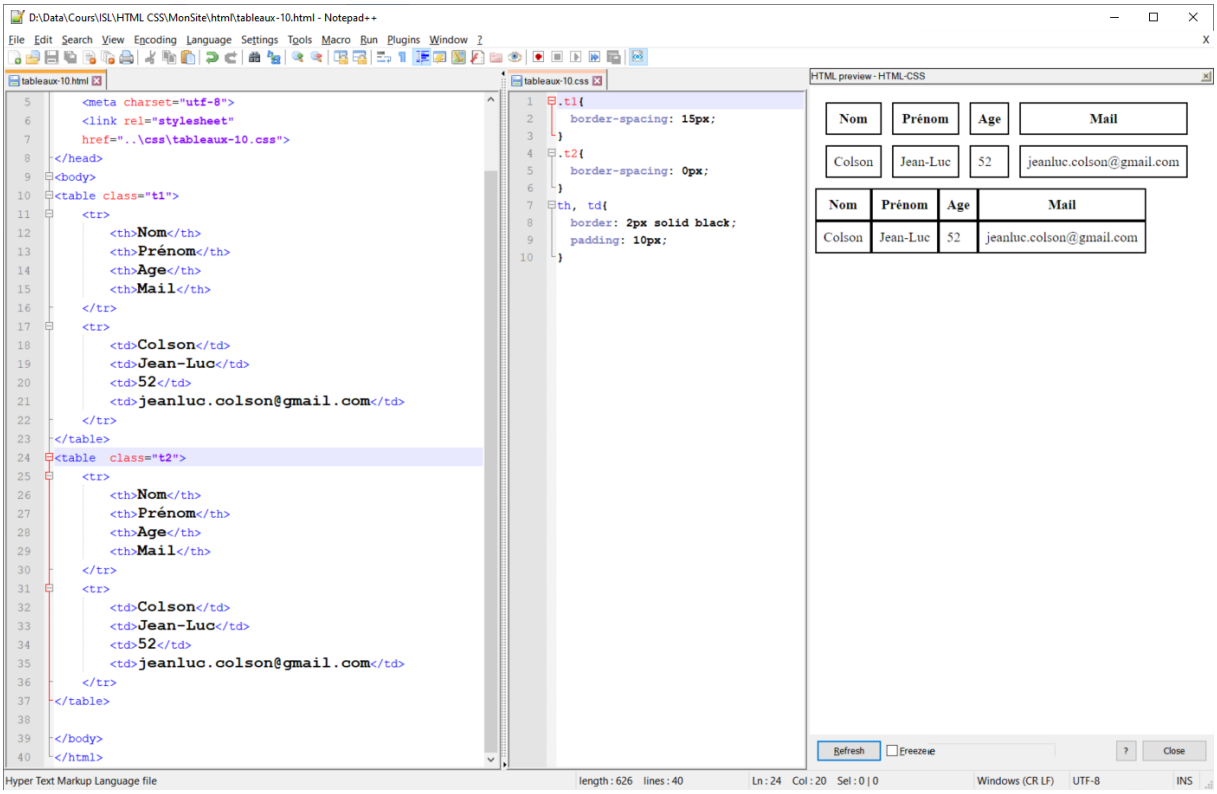
- separate : Valeur par défaut ; les bordures seront séparées et distinctes ;
- collapse : Les bordures adjacentes vont être fusionnées.

Nous allons donc généralement appliquer border à nos éléments th et td ainsi que la propriété border-collapse avec sa valeur collapse à notre élément table.




Dans le cas où l'on souhaite conserver des bordures séparées, on va tout de même pouvoir gérer la distance entre chaque bordure grâce à la propriété border-spacing.

Cette propriété va prendre une distance en valeur (en px, cm, em, rem, etc.) qui représente l'espace entre deux bordures adjacentes. Il faudra à nouveau l'appliquer à l'élément table.



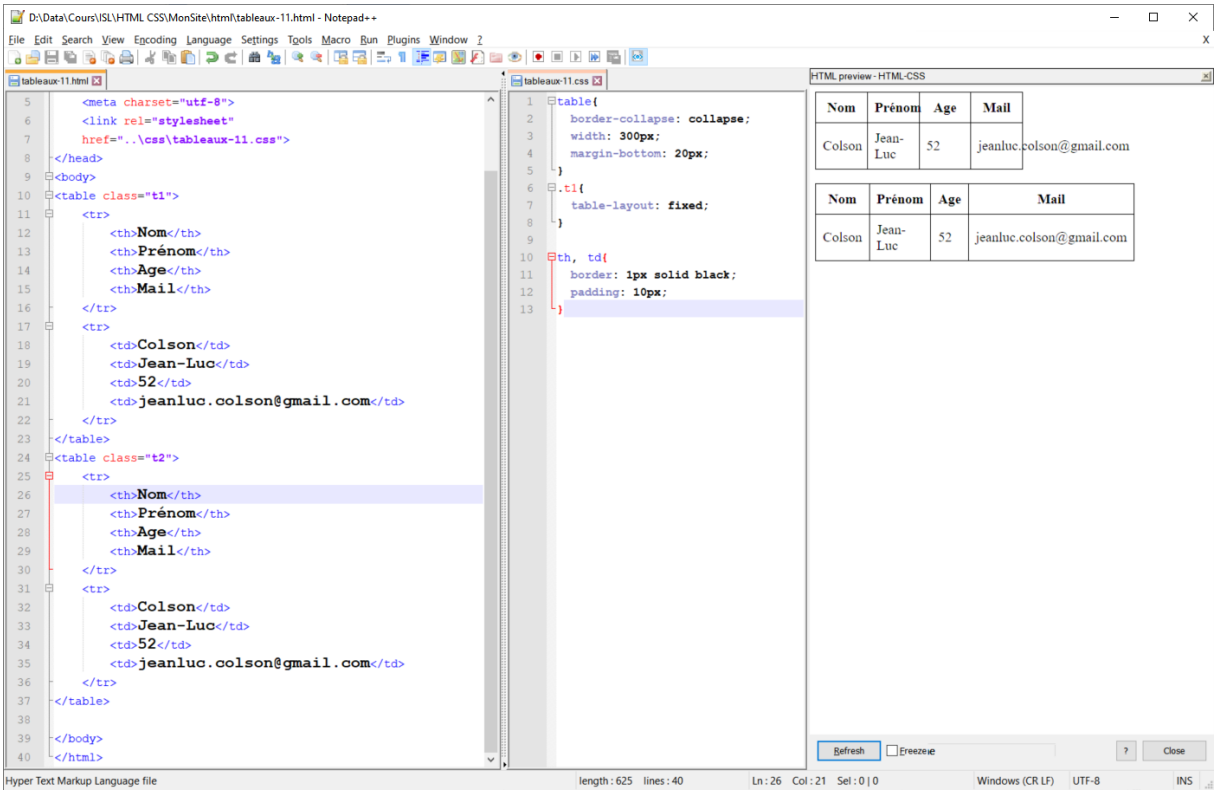

Notez qu'indiquer un border-spacing : 0px ne va pas fusionner les bordures des cellules adjacentes comme le ferait border-collapse : collapse mais simplement coller les bordures des cellules adjacentes.

## Contrôler la largeur des colonnes avec table-layout

La propriété table-layout va nous permettre de gérer la largeur des différentes colonnes de notre tableau ou plus exactement de choisir selon quel algorithme la largeur de chaque colonne doit être calculée.

On va pouvoir choisir parmi deux valeurs :

- auto : Valeur par défaut ; la largeur des différentes colonnes du tableau est calculée automatiquement afin que le tableau prenne le moins d'espace possible en hauteur. De plus, si une largeur a été définie avec width pour le tableau, alors celui-ci essaiera de la respecter tant que son contenu ne dépasse pas du tableau ;
- fixed : La largeur des colonnes va être déterminée à partir de la propriété width des éléments table ou col ou de la première ligne du tableau.



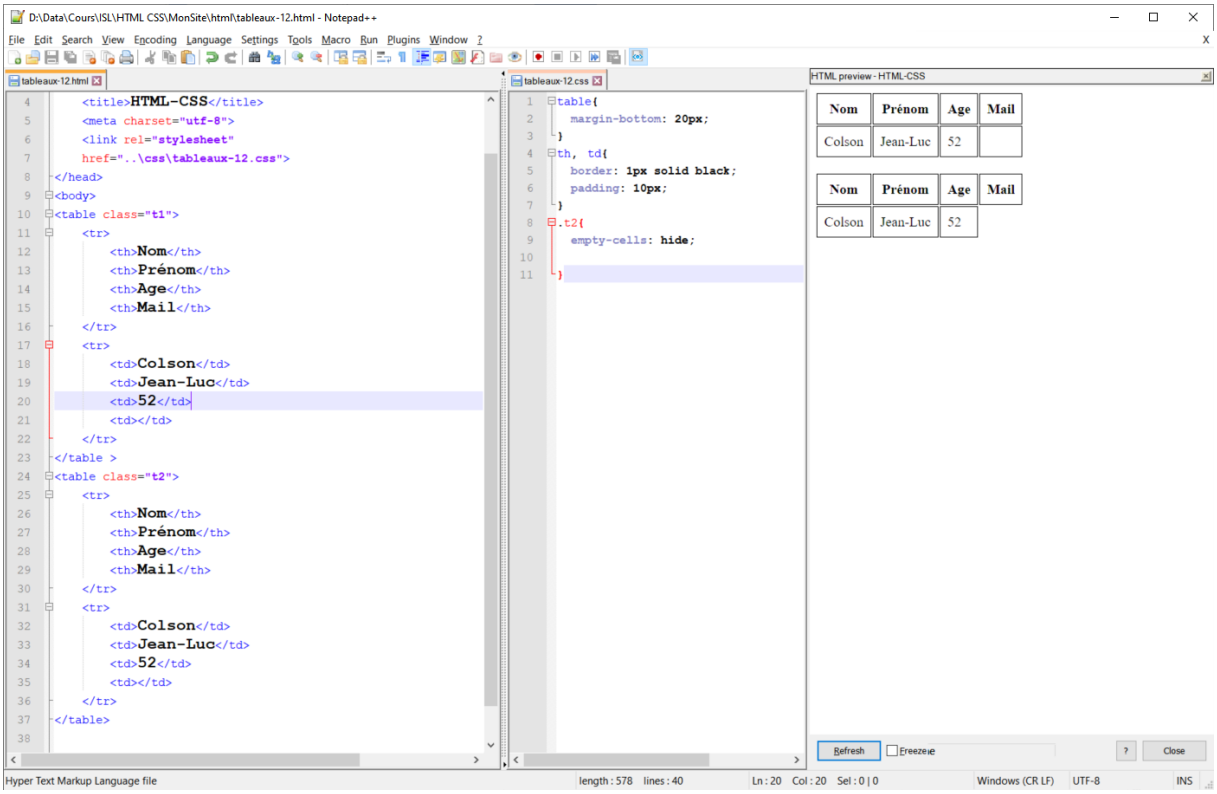

## Définir l'affichage des cellules vides d'un tableau avec empty-cells

La propriété empty-cells va nous permettre de choisir si l'arrière-plan et les bordures d'une cellule sans contenu doivent être affichés ou pas.

Notez que cette propriété ne va pouvoir s'appliquer que si la valeur de la propriété border-collapse pour le tableau est separate.

On va pouvoir choisir parmi deux valeurs :

- show : Valeur par défaut ; les bordures et l'arrière-plan des cellules vides du tableau seront bien visibles ;
- hide : Les bordures et l'arrière-plan des cellules vides du tableau ne sont pas affichés.

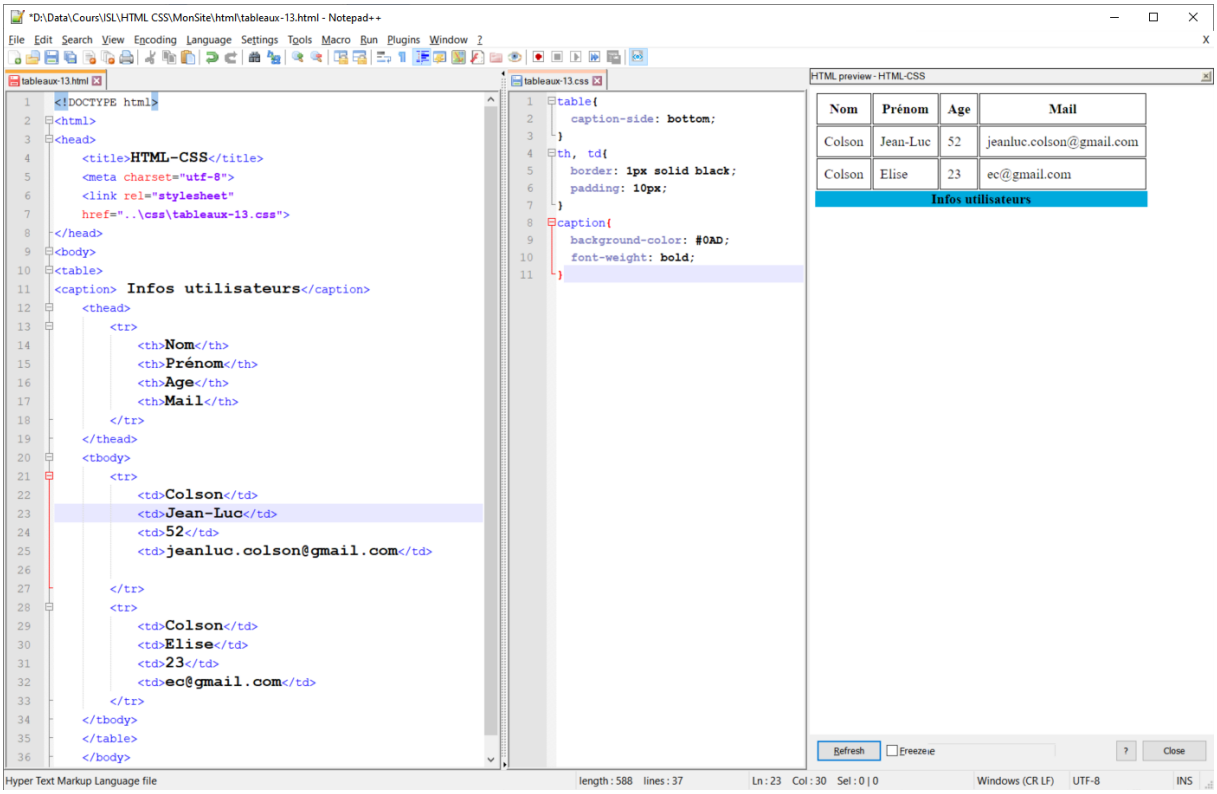



## Choisir où s'affiche la légende d'un tableau avec caption-side

La propriété caption-side va nous permettre de choisir où doit s'afficher la légende d'un tableau, c'est-à-dire où doit s'afficher le texte présent dans l'élément caption du tableau.

Aujourd'hui, nous pouvons choisir entre deux valeurs qui représentent deux emplacements :

- top : Valeur par défaut ; la légende sera positionnée au-dessus du tableau ;
- bottom : La légende sera positionnée en dessous du tableau.



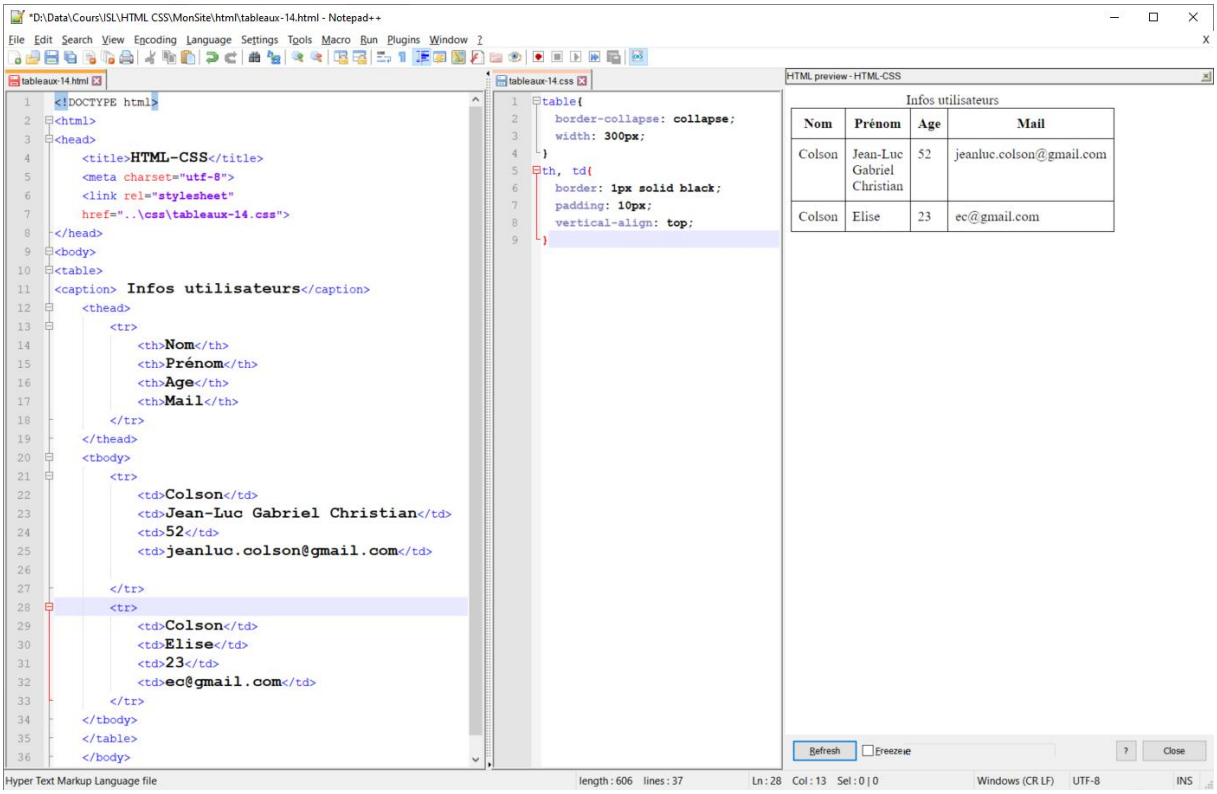

## Aligner le contenu des cellules d'un tableau avec vertical-align

La propriété vertical-align sert à définir l'alignement vertical du contenu d'une cellule d'un tableau.

On va pouvoir lui passer une valeur de type longueur (en px ou en em par exemple), une valeur en pourcentage ou un mot clef à cette propriété.

Dans la grande majorité des cas, nous nous contenterons de choisir parmi l'un de ces 3 mots clefs :

- top : Le contenu de chaque cellule va se placer en haut de la cellule ;
- middle : Le contenu de chaque cellule va se placer au milieu de la cellule ;
- bottom : Le contenu de chaque cellule va se placer se bas de la cellule.




```html
<!DOCTYPE html>
<html>
<head>
    <title>HTML-CSS</title>
    <meta charset="utf-8">
    <link rel="stylesheet"
    href="..\css\tableaux-14.css">
</head>
<body>
<table>
<caption> Infos utilisateurs</caption>
    <thead>
        <tr>
            <th>Nom</th>
            <th>Prénom</th>
            <th>Age</th>
            <th>Mail</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>Colson</td>
            <td>Jean-Luc Gabriel Christian</td>
            <td>52</td>
            <td>jeanluc.colson@gmail.com</td>

        </tr>
        <tr>
            <td>Colson</td>
            <td>Elise</td>
            <td>23</td>
            <td>ec@gmail.com</td>
        </tr>
    </tbody>
    </table>
    </body>
```

```css
table{
    border-collapse: collapse;
    width: 300px;
}
th, td{
    border: 1px solid black;
    padding: 10px;
    vertical-align: top;
}
```