

Pages HTML - Partie 1

1 - Introduction

L'objectif est de donner au lecteur un petit coup de pouce pour le rendre autonome dans l'écriture de pages HTML et qu'il puisse s'appuyer sur la webographie et la bibliographie abondantes sur le sujet.

Les pages seront écrites au moyen d'un éditeur de texte adapté, ici notepad++ (téléchargement gratuit). Elles pourront être lues par un navigateur tel que Google Chrome, Internet Explorer, Firefox, etc, également gratuits. Des pages complexes pourront n'être que partiellement compatibles avec tous les navigateurs. Le navigateur Firefox a servi pour tester les pages présentées.

Trois langages sont en fait introduits ici : HTML 5, JavaScript et CSS 3, complémentaires et indissociables. HTML donne l'ossature d'une page, JavaScript permet de traiter des actions pour cette page (clic de souris, évolution temporelle, ...) et CSS contrôle la présentation.

HTML est utilisé pour diffuser des informations sous forme de sites internet. La page HTML doit alors être stockée sur un serveur et accessible au moyen d'une adresse (comme <http://www.tuclic.fr>). Le fichier index.htm situé à cette adresse est alors lu par le navigateur, d'où l'importance de nommer index.htm le fichier qui contient la page d'accueil du site.

On peut aussi utiliser ce langage pour des applications locales, telles que la réalisation d'un diaporama image et son (taper sous Google les mots clefs : "tuclic diaporama" ou suivre le lien http://www.tuclic.fr/tuto_pdf_09/_03_diaporama_html5_01.pdf), d'un instrument de musique piloté par un clavier MIDI (voir par exemple sous Google avec les mots clefs "tuclic tempéraments", ou suivre le lien <http://www.tuclic.fr/exec/index.htm>), ou d'une interface graphique interactive pour du calcul intensif (voir par exemple sous Google "tuclic fusées", ou suivre le lien http://www.tuclic.fr/tuto_pdf_06/fusee/index.htm). le langage HTML, associé à Javascript et CSS a de moins en moins de limites pour utiliser les possibilités d'un ordinateur.

2 - Page HTML : le minimum

Créer un dossier page_01 qui contiendra ce premier exemple. Lancer notepad++. Ecrire le début du texte en langage HTML ci-dessous. L'enregistrer dans le dossier page_01 sous le nom index.htm. Les mots clefs du langage HTML apparaissent alors en couleur dans notepad++ ce qui permet d'éliminer un nombre important de fautes de frappe lors de la saisie des pages. On notera que ce langage utilise des balises ouvrantes telles que <body> et des balises fermantes telles que </body>. La page la plus simple débute par les deux lignes :

<!doctype html> et <html> qui indiquent au navigateur qu'il s'agit d'une page HTML.

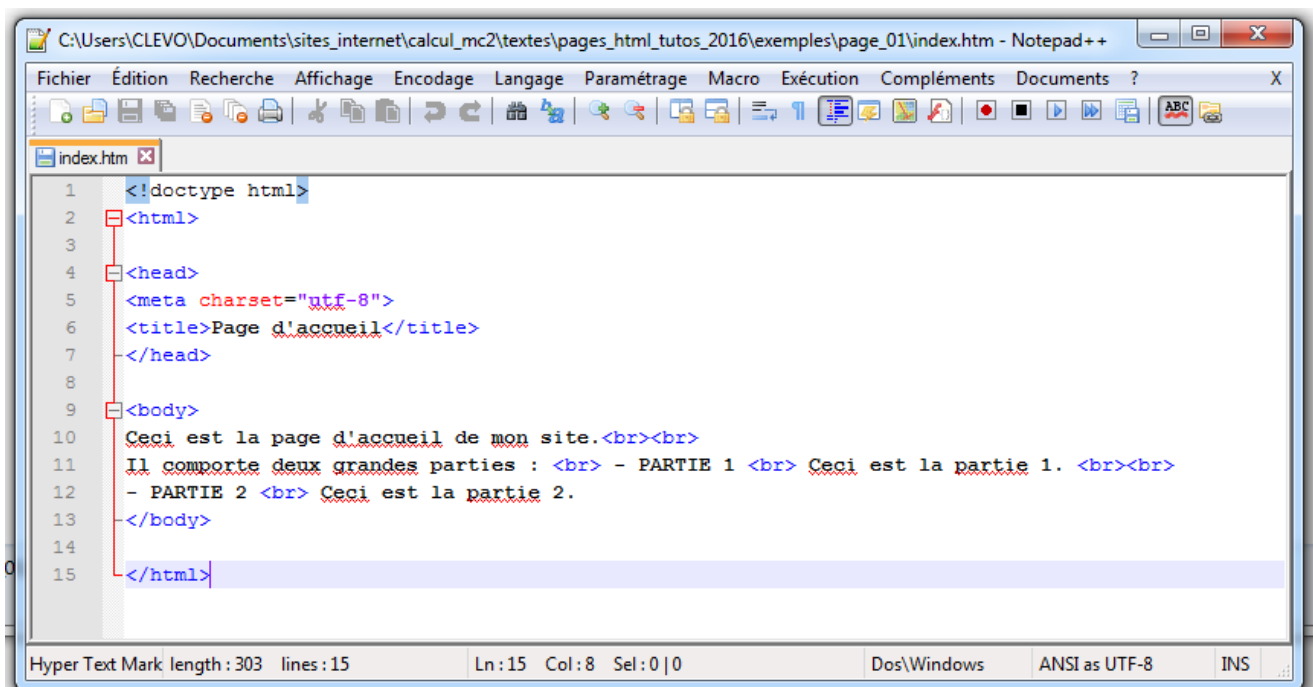
La page contient ensuite un entête entre les deux balises <head> et </head>. Elle contient ensuite un corps entre les deux balises <body> et </body>.

La page se termine par la balise fermante </html>.

Après avoir saisi la totalité de la page ci-dessous, Fichier --> Enregistrer. Un double clic sur le nom du fichier dans l'explorateur de fichiers permet de lancer Firefox et d'afficher "index.htm" si Firefox est le navigateur par défaut. Si ce n'est pas le cas, clic droit sur le nom du fichier index.htm et "ouvrir avec" --> Firefox.

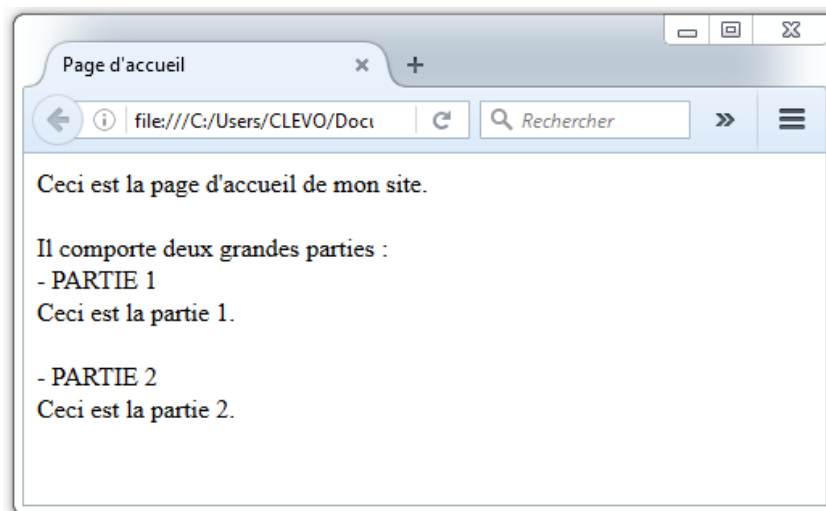
Un clic droit sur le fichier index.htm et "éditer avec notepad++" permet de modifier le fichier.

Contenu du fichier index.htm dans le dossier page_01, édité dans notepad++ :



```
1 <!doctype html>
2 <html>
3
4 <head>
5   <meta charset="utf-8">
6   <title>Page d'accueil</title>
7 </head>
8
9 <body>
10  Ceci est la page d'accueil de mon site.<br><br>
11  Il comporte deux grandes parties : <br> - PARTIE 1 <br> Ceci est la partie 1. <br><br>
12  - PARTIE 2 <br> Ceci est la partie 2.
13 </body>
14
15 </html>
```

Résultat : ouverture de la page dans le navigateur Firefox :



La balise <meta> permet d'indiquer la table de caractères utilisée, qui inclut ici les lettres accentuées. Les balises <title> et </title> encadrent le titre qui figure dans l'onglet du navigateur. Les balises
 provoquent un retour à la ligne.

Tester l'ajout ou la suppression de balises
.

Modifier le texte en noir entre les balises <body> et </body>.

Toujours veiller à la bonne imbrication des balises ouvrantes et fermantes :

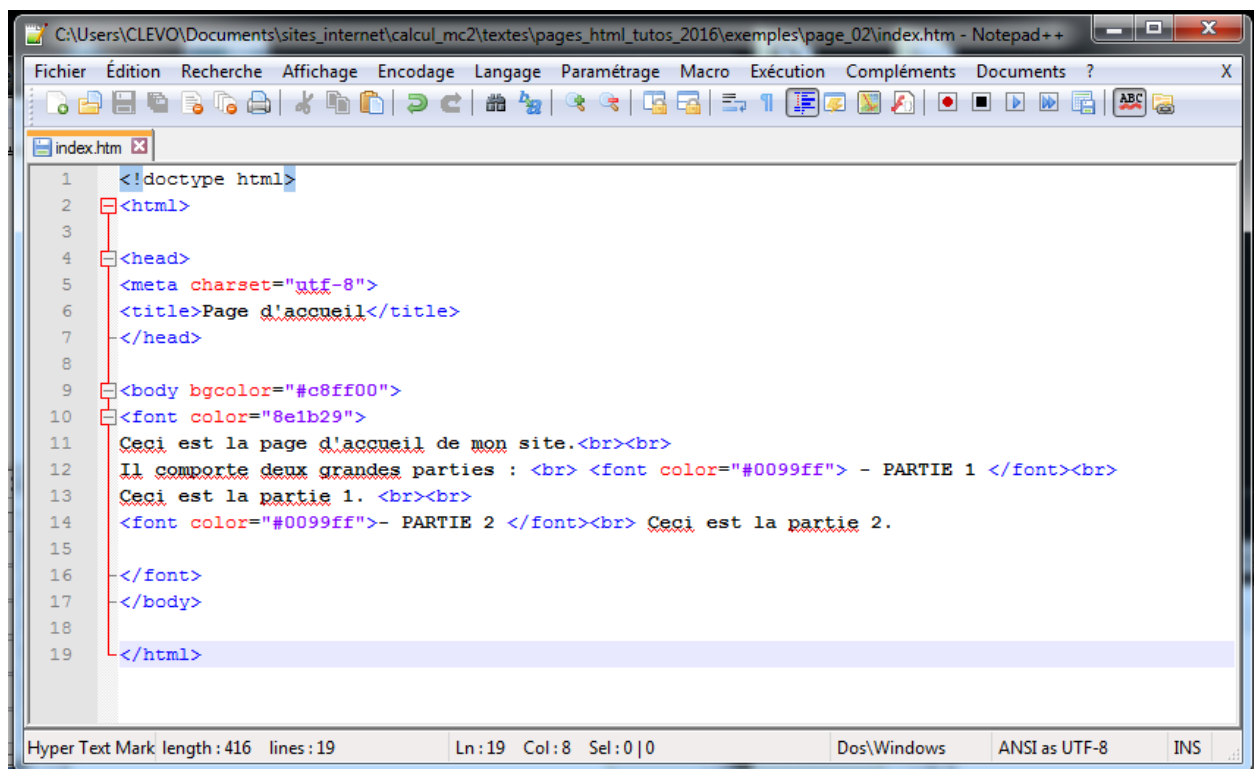
<balise ouvrante 1> <balise ouvrante 2> ... </balise fermante 2> </balise fermante 1>

3 - Un peu de couleur dans la page

Créer un dossier page_02 qui contiendra le deuxième exemple (exemple 1 modifié).

A partir du premier exemple, dans notepad++, Fichier --> Enregistrer sous ... Modifier le dossier (aller dans page_02). Laisser le nom index.htm. Cliquer sur enregistrer.

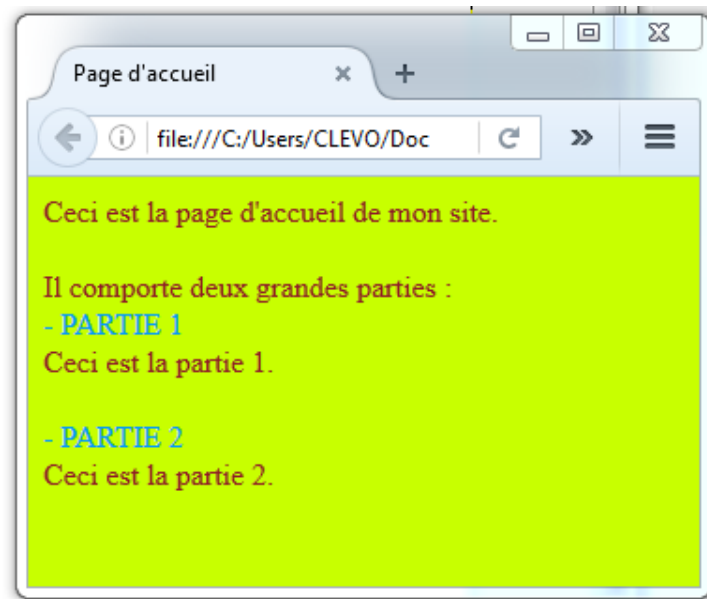
Modifier le fichier index.htm dans notepad++ comme ci dessous :



```
1 <!doctype html>
2 <html>
3
4 <head>
5 <meta charset="utf-8">
6 <title>Page d'accueil</title>
7 </head>
8
9 <body bgcolor="#e8ff00">
10 <font color="8e1b29">
11 Ceci est la page d'accueil de mon site.<br><br>
12 Il comporte deux grandes parties : <br> <font color="#0099ff"> - PARTIE 1 </font><br>
13 Ceci est la partie 1. <br><br>
14 <font color="#0099ff">- PARTIE 2 </font><br> Ceci est la partie 2.
15
16 </font>
17 </body>
18
19 </html>
```

Fichier --> Enregistrer.

On obtient le résultat suivant avec Firefox, en double cliquant sur index.htm dans le dossier page_02 :

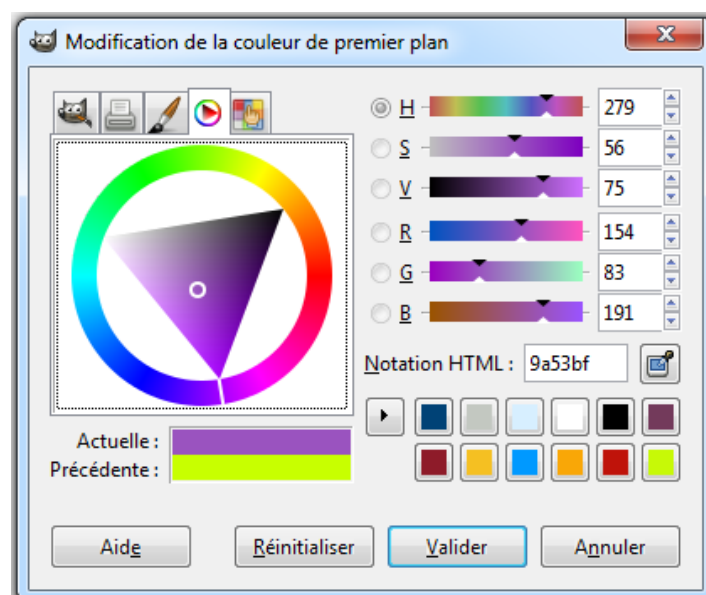


La disposition est la même. La couleur d'arrière plan est modifiée par la balise : `<body bgcolor="#xxxxxxx">` où xxxxxx est la notation hexadécimale de la couleur (ici du jaune fluo avec xxxxxx=c8ff00)

Deux paires de balises imbriquées `` et `` permettent de modifier la couleur du texte (rouge sombre avec xxxxxx=8e1b29 et bleu clair avec xxxxxx=0099ff).

La notation hexadécimale d'une couleur peut être utilisée sous deux points de vue :

- On peut utiliser un logiciel qui établit la correspondance entre une couleur et le code hexadécimal, par exemple, PsPad ou un gros logiciel de dessin comme GIMP, qui peuvent être téléchargés gratuitement. La fenêtre des couleurs de GIMP se présente de la façon suivante :



Le triangle permet de choisir une teinte par rotation et d'y ajouter du noir et du blanc. La notation de ce mauve est donc 9a53bf. On peut se référer pour plus de détails à : http://www.tuclie.fr/tuto_pdf_03/_08_01_couleurs_gimp_01.pdf.

- On utilise les quantités de rouge, de vert et de bleu, entre 0 et 255, indiquées chacune par deux chiffres hexadécimaux entre 0 et 15 : 0 1 2 3 4 5 6 7 8 9 a b c d e f, où a vaut 10 et f vaut 15. On a en particulier :

Noir = 000000

Blanc = ffffff

Rouge = ff0000

Vert = 00ff00

Bleu = 0000ff

Jaune = ffff00 (Rouge + Vert)

Cyan = 00ffff (Vert + Bleu)

Magenta = ff00ff (Rouge + Bleu)

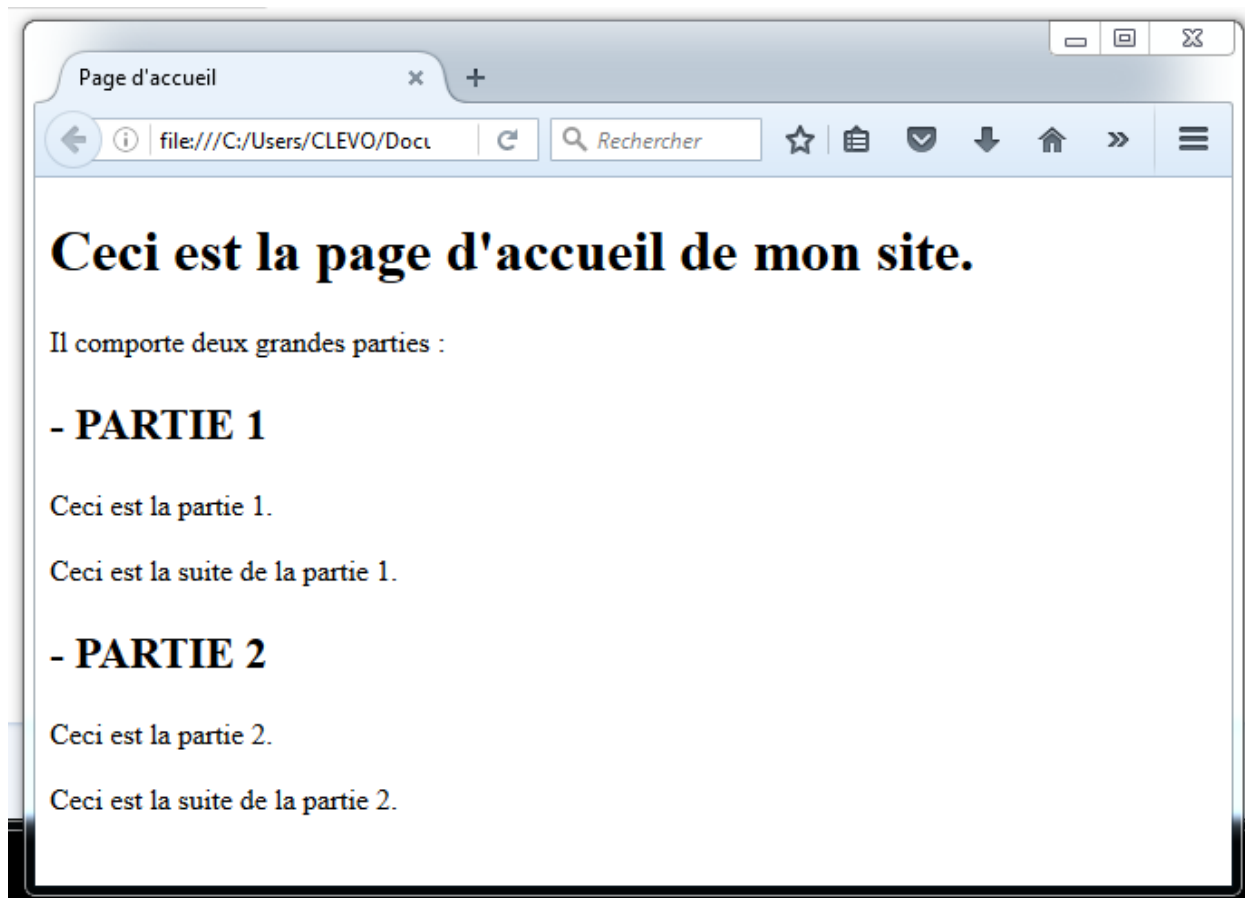
Modifier les codes des couleurs dans cet exemple 2 avec notepad++ et constater l'effet dans Firefox.

4 - Structuration du contenu de la page

Créer un dossier page_03 qui contiendra le troisième exemple (exemple 1 modifié de façon différente).

Editer à nouveau dans notepad++ le premier exemple : index.htm du dossier page_01, et l'enregistrer dans le dossier page_03 sous le même nom index.htm.

On obtient l'affichage suivant dans Firefox :



On a placé un grand titre entre les balises `<h1>` et `</h1>`.

Les sous-titres sont placés entre les balises `<h2>` et `</h2>`.

On peut continuer ainsi avec `h3`, `h4`, `h5`, `h6`, avec des tailles de caractères décroissantes.

Les paragraphes sont placés entre `<p>` et `</p>`, avec retour à la ligne automatique.

On ne peut pas imbriquer des balises `<h>` entre elles. On ne peut pas imbriquer de balises `<p>` entre elles.

Mise à part la taille des caractères, qui peut être modifiée, le résultat est voisin de l'exemple 1 dans lequel on utilisait des balises de retour à la ligne `
`.

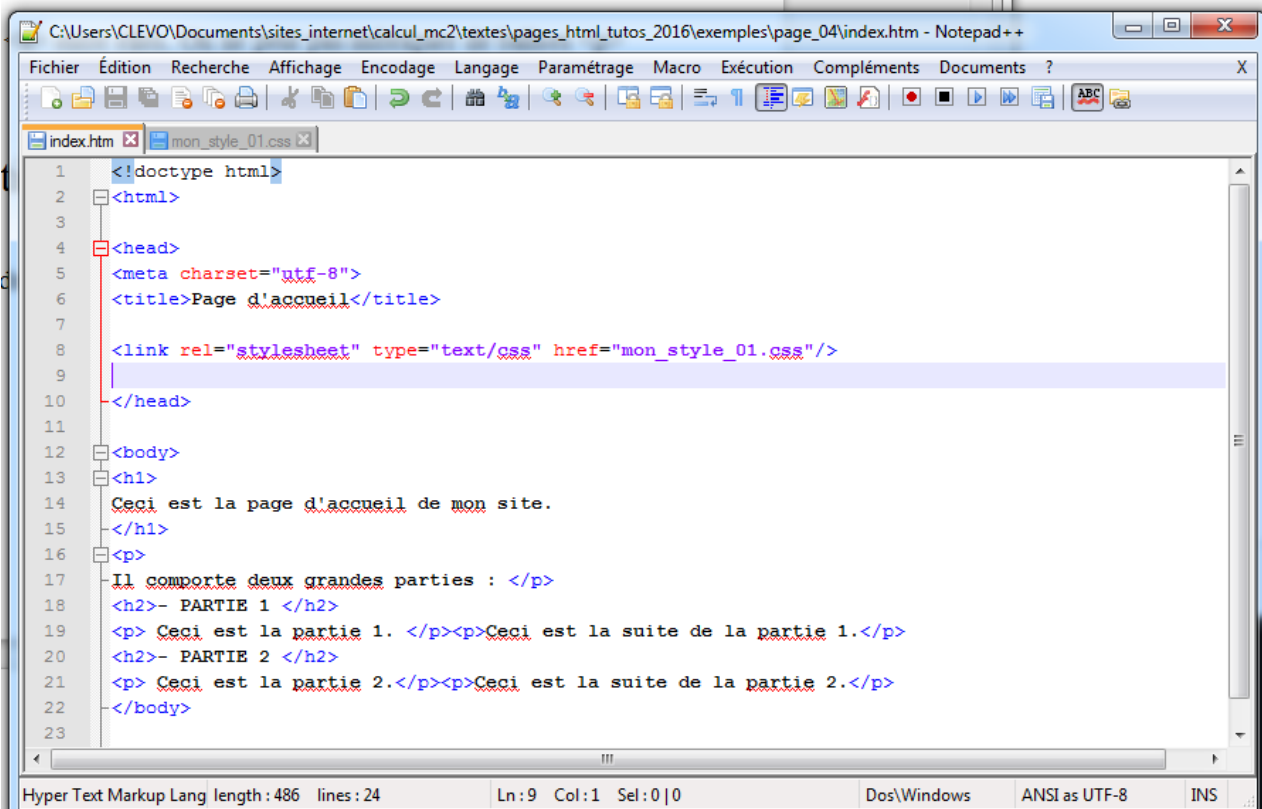
Bien que cela paraisse plus compliqué, il est préférable d'utiliser `<p> ... </p>` pour encadrer les paragraphes et `<h> ... </h>` pour les titres. Il est plus logique de définir la mise en forme par rapport au sens d'une partie de texte, c'est à dire de façon sémantique (titre, paragraphe, complément d'information, ...), qu'en considérant simplement la juxtaposition de caractères de tailles et de couleurs différentes. C'est l'objet du paragraphe suivant. Les moteurs de recherche sont également sensés y attacher de l'importance.

5 - Couleurs, structuration du contenu et CSS

Créer un dossier page_04 qui contiendra le quatrième exemple (exemple 3 modifié).

Dans notepad++ : Fichier --> Enregistrer sous ... --> Dossier page_04 --> index.htm

On ajoute une ligne dans le fichier index.htm qui renvoie à une feuille de style : ce sera le fichier mon_style_01.css (ligne 8).

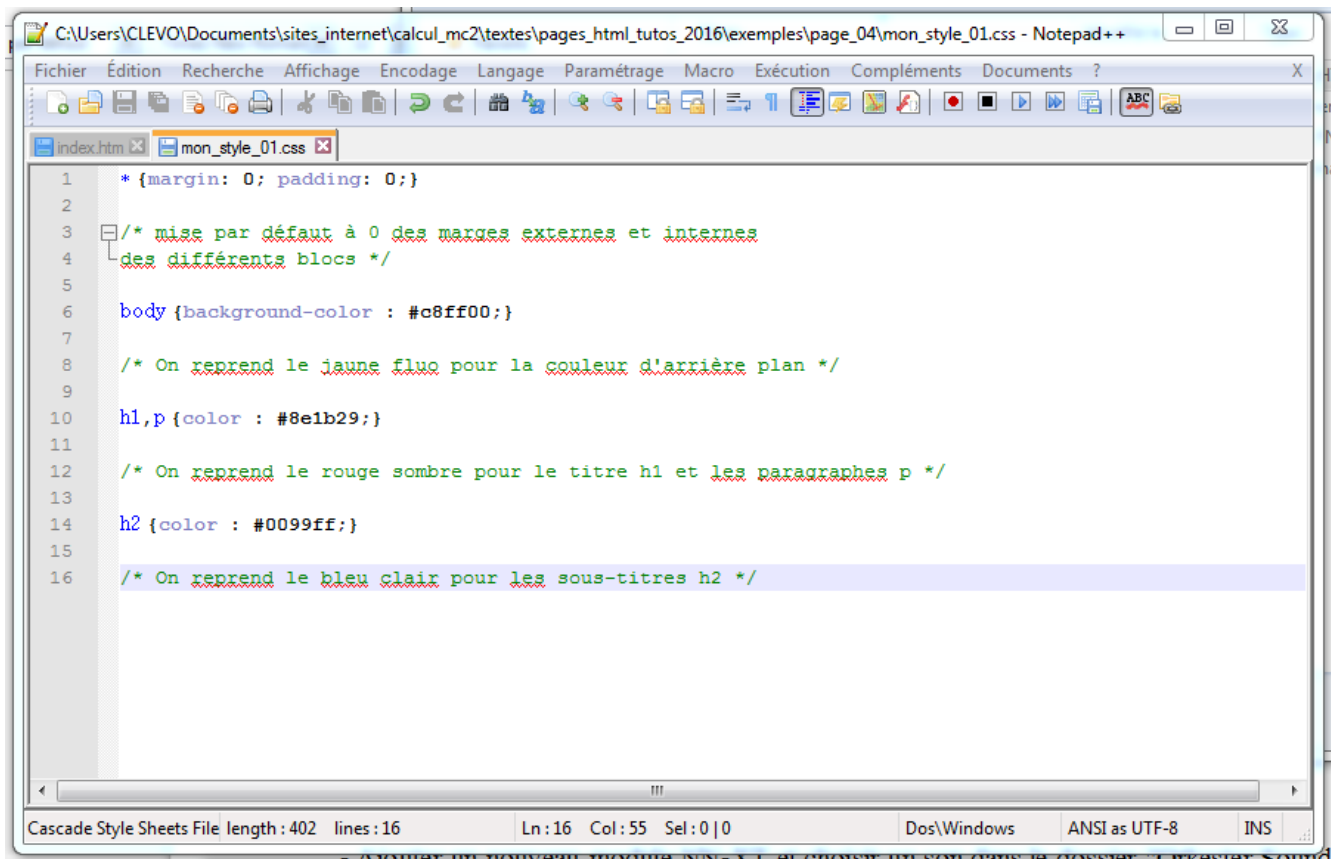


```
1 <!doctype html>
2 <html>
3
4 <head>
5 <meta charset="utf-8">
6 <title>Page d'accueil</title>
7
8 <link rel="stylesheet" type="text/css" href="mon_style_01.css"/>
9
10 </head>
11
12 <body>
13 <h1>
14 Ceci est la page d'accueil de mon site.
15 </h1>
16 <p>
17 Il comporte deux grandes parties : </p>
18 <h2>- PARTIE 1 </h2>
19 <p> Ceci est la partie 1. </p><p>Ceci est la suite de la partie 1.</p>
20 <h2>- PARTIE 2 </h2>
21 <p> Ceci est la partie 2.</p><p>Ceci est la suite de la partie 2.</p>
22 </body>
23
```

Fichier --> Enregistrer.

La feuille de style peut déterminer les couleurs, les positions des textes et des différents éléments de la page HTML qui y fait référence, ainsi que d'autres paramètres de mise en page. On peut parler de langage CSS pour cette feuille de style.

Dans notepad++ : Fichier --> Nouveau. Commencer à saisir la feuille suivante et faire : Fichier --> Enregistrer sous ... mon_style_01.css. Les mots clefs du CSS apparaissent alors en couleurs. Finir de saisir la feuille et l'enregistrer.



```
1  * {margin: 0; padding: 0;}
2
3  /* mise par défaut à 0 des marges externes et internes
4  des différents blocs */
5
6  body {background-color : #c8ff00;}
7
8  /* On reprend le jaune fluo pour la couleur d'arrière plan */
9
10 h1,p {color : #8e1b29;}
11
12 /* On reprend le rouge sombre pour le titre h1 et les paragraphes p */
13
14 h2 {color : #0099ff;}
15
16 /* On reprend le bleu clair pour les sous-titres h2 */
```

On obtient le résultat suivant en ouvrant index.htm dans Firefox :



La feuille de style est constituée d'une suite d'instructions comportant chacune 2 éléments :

- un sélecteur qui définit à quelle(s) balises de la page HTML l'instruction se rapporte,
- les attributs qu'on souhaite modifier pour cette ou ces balises, placés entre accolades { .. }.

Le sélecteur `*` signifie toutes les balises. `margin` concerne la marge extérieure à un bloc, `padding` à la marge intérieure (ligne 1 du fichier `mon_style_01.css`).

Le sélecteur `body` renvoie à la balise `<body>`, c'est à dire pour définir une valeur par défaut pour le corps de la page, et ici pour la couleur d'arrière plan (ligne 6).

Le sélecteur `h1, p` renvoie à la balise `<h1>` (grand titre) et aux balises `<p>` (paragraphes). La virgule est essentielle. Essayer de la remplacer par un espace : le sélecteur `h1 p` signifie "les balises `<p>` situées dans un bloc délimité par `<h1>` et `</h1>`". Cette situation n'apparaît pas dans la page HTML, et il n'y a plus de caractères de couleur rouge sombre `color : #8e1b29`. Remettre la virgule (ligne 10).

Le sélecteur `h2` renvoie à la balise `<h2>` (sous-titre). `color : #0099ff` met les sous-titres en bleu clair (ligne 14).

`/* ... */` encadre un commentaire qui n'est pas interprété.

Déplacer dans la feuille de style `mon_style_01.css` le sélecteur `h1` et la virgule de la ligne 10 à la ligne 14 pour que le grand titre apparaisse en bleu clair, comme les sous-titres `h2`. Enregistrer la modification de la feuille de style et actualiser la page `index.htm`.

Les points-virgules servent à séparer les instructions entre les accolades.

6 - Position des blocs et CSS

Créer un dossier `page_05` qui contiendra le cinquième exemple (exemple 4 modifié).

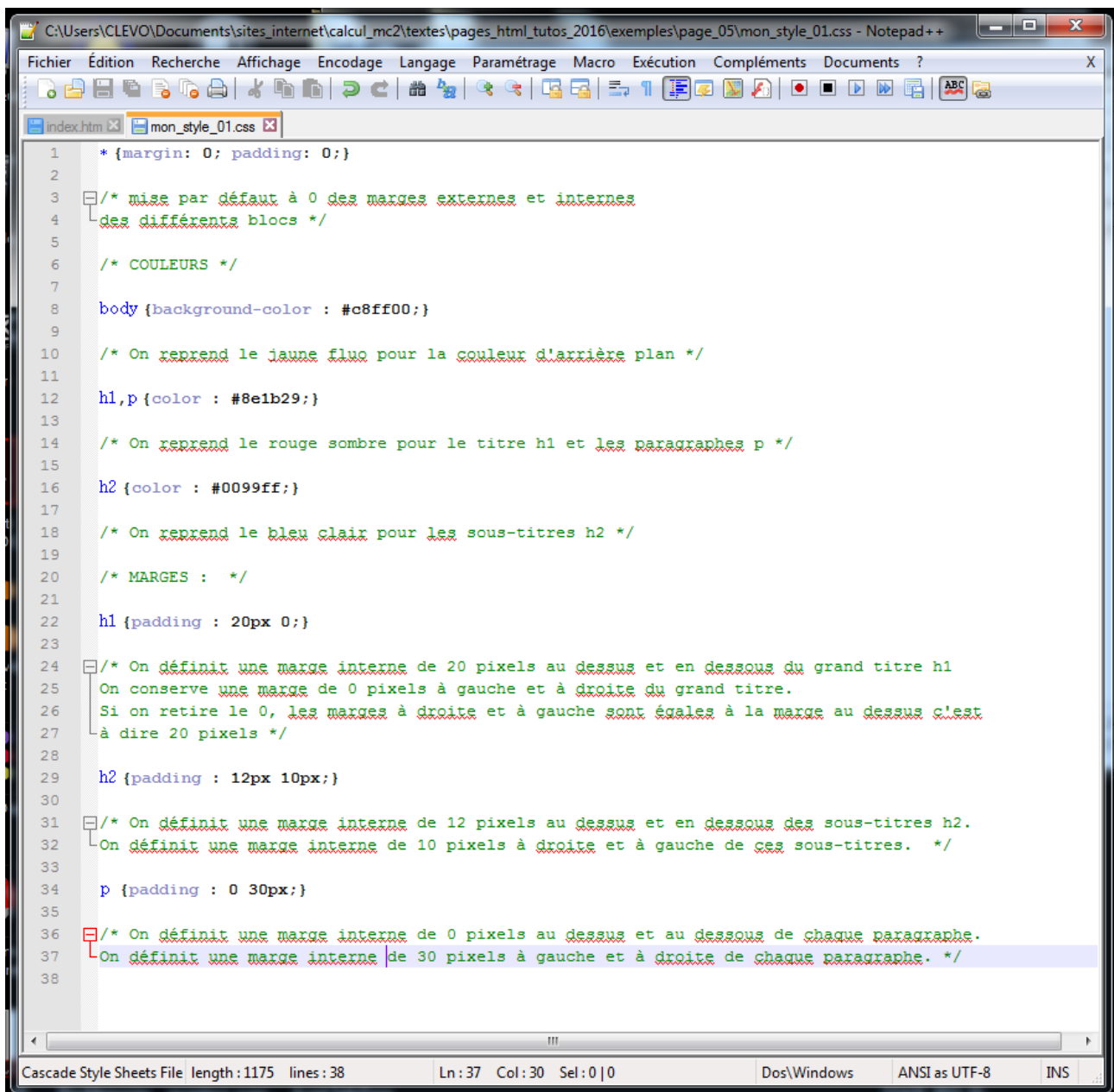
Rendre actif dans notepad++ l'onglet "`index.htm`".

Dans notepad++ : Fichier --> Enregistrer sous ... --> Dossier `page_05` --> `index.htm`

Rendre actif dans notepad++ l'onglet `mon_style_01.css` : Fichier --> Enregistrer sous ... --> Dossier `page_05` --> `mon_style_01.css`.

On ne modifie pas la page HTML par rapport à l'exemple 4, qui fait référence à la feuille de style `mon_style_01.css`.

Le but est d'espacer les titres des paragraphes. On peut y parvenir en augmentant la marge au dessus et en dessous des titres (attribut `margin`). On peut également déplacer ces éléments par rapport à leur position "dans le flux", c'est à dire celle qu'on observe en l'absence de feuille de style (exemple 4 du §5). On essaiera les deux méthodes (la première avec les marges dans ce paragraphe et la seconde avec les déplacements des blocs dans le paragraphe suivant §7).



```
1  *{margin: 0; padding: 0;}
2
3  /* mise par défaut à 0 des marges externes et internes
4  des différents blocs */
5
6  /* COULEURS */
7
8  body {background-color : #c8ff00;}
9
10 /* On reprend le jaune fluo pour la couleur d'arrière plan */
11
12 h1,p {color : #8e1b29;}
13
14 /* On reprend le rouge sombre pour le titre h1 et les paragraphes p */
15
16 h2 {color : #0099ff;}
17
18 /* On reprend le bleu clair pour les sous-titres h2 */
19
20 /* MARGES : */
21
22 h1 {padding : 20px 0;}
23
24 /* On définit une marge interne de 20 pixels au dessus et en dessous du grand titre h1
25 On conserve une marge de 0 pixels à gauche et à droite du grand titre.
26 Si on retire le 0, les marges à droite et à gauche sont égales à la marge au dessus c'est
27 à dire 20 pixels */
28
29 h2 {padding : 12px 10px;}
30
31 /* On définit une marge interne de 12 pixels au dessus et en dessous des sous-titres h2.
32 On définit une marge interne de 10 pixels à droite et à gauche de ces sous-titres. */
33
34 p {padding : 0 30px;}
35
36 /* On définit une marge interne de 0 pixels au dessus et au dessous de chaque paragraphe.
37 On définit une marge interne de 30 pixels à gauche et à droite de chaque paragraphe. */
38
```

La première partie du fichier mon_style_01.css est identique à celle du §5 "COULEURS". La partie suivante "MARGES" affecte des marges internes à chaque balise, en pixels (px) ou la valeur 0. La première valeur concerne les marges internes au dessus et en dessous du contenu de la balise. La deuxième valeur concerne les marges internes à gauche et à droite du contenu de la balise. On peut indiquer des valeurs positives ou négatives.

Modifier la valeur des marges en noir pour voir l'effet sur l'affichage dans le navigateur.



7 - Position des blocs et CSS : balise <div>

Créer un dossier page_06 qui contiendra le sixième exemple (exemple 4 modifié).

Fermer les onglets ouverts dans notepad++. Ouvrir, depuis le dossier page_04, les deux fichiers index.htm et mon_style_01.css.

Les enregistrer tous les deux sous leur ancien nom, mais dans le dossier page_06.

Dans un premier temps, on va ajouter des balises <div> et </div> dans la page HTML index.htm.

Les balises <div> et </div> définissent un conteneur qui ne modifie rien par lui-même. On peut imbriquer des balises <div> et </div> pour avoir des conteneurs imbriqués. On pourra ensuite positionner les conteneurs et les sous-conteneurs au moyen de la feuille de style mon_style_01.css.

Modifier comme ci-dessous la page index.htm du dossier page_06.

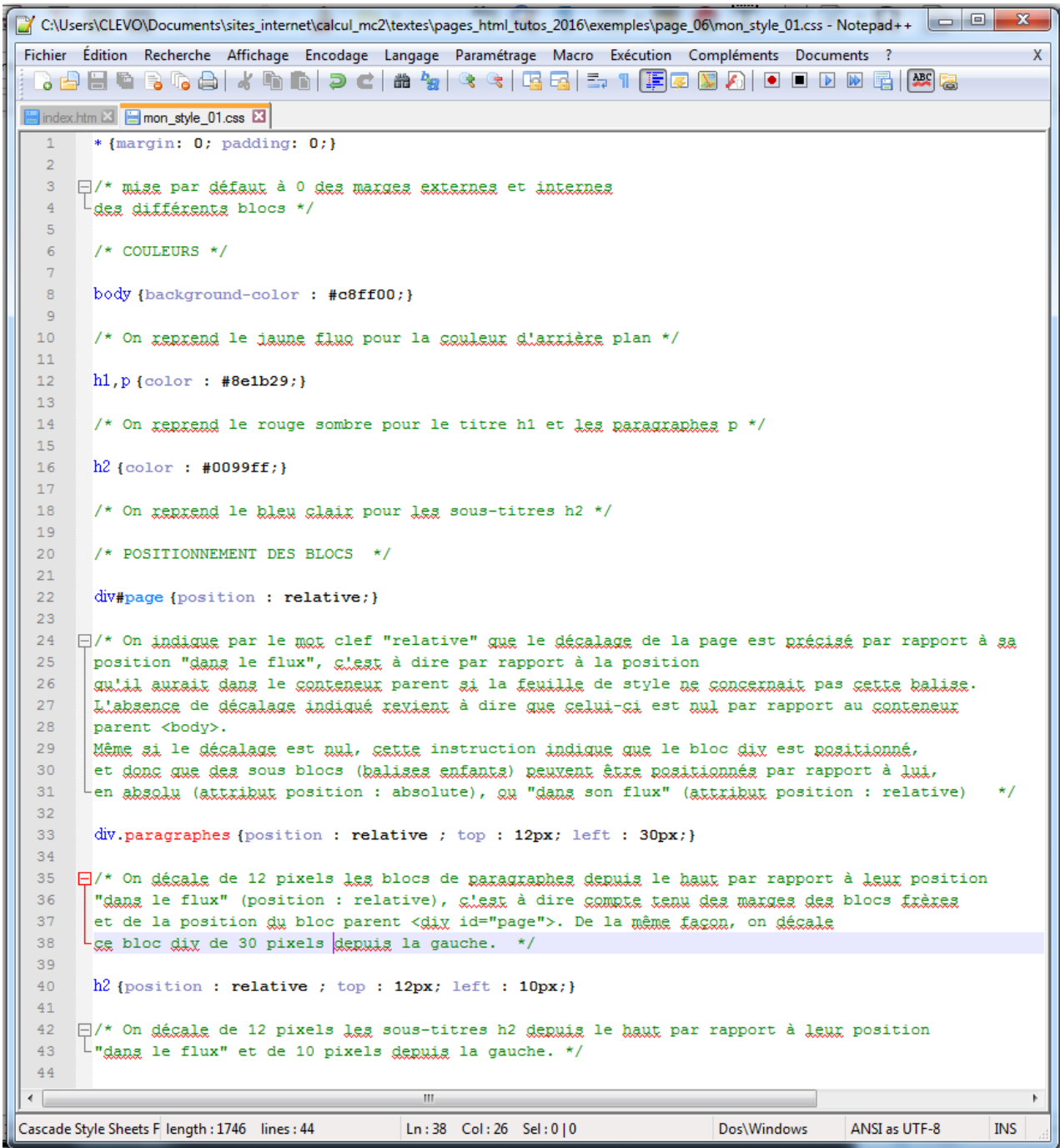
Les conteneurs sont définis par les balises <div> et </div> aux lignes suivantes :

- 14 -- 39 : bloc qui contient toute la page
- 20 -- 25 : bloc qui contient le premier paragraphe
- 29 -- 31 : bloc qui contient les paragraphes de la partie 1
- 35 -- 37 : bloc qui contient les paragraphes de la partie 2

Le rôle des conteneurs n'est pas le même pour ces 4 blocs : le premier encadre l'ensemble de la page. On lui donne un identifiant `id="page"`. Les 3 autres encadrent des paragraphes groupés dans une même partie. On leur associe une classe qui traduit ce point commun : `class="paragraphes"`. Le nom qu'on donne à l'identifiant ou à une classe entre guillemets est totalement libre.

- 12 / 18 -
HTML - Partie 1

Feuille de style mon_style_01.css du dossier page_06 :



```
1  *{margin: 0; padding: 0;}
2
3  /* mise par défaut à 0 des marges externes et internes
4  des différents blocs */
5
6  /* COULEURS */
7
8  body {background-color : #c8ff00;}
9
10 /* On reprend le jaune fluo pour la couleur d'arrière plan */
11
12 h1,p {color : #8e1b29;}
13
14 /* On reprend le rouge sombre pour le titre h1 et les paragraphes p */
15
16 h2 {color : #0099ff;}
17
18 /* On reprend le bleu clair pour les sous-titres h2 */
19
20 /* POSITIONNEMENT DES BLOCS */
21
22 div#page {position : relative;}
23
24 /* On indique par le mot clef "relative" que le décalage de la page est précisé par rapport à sa
25 position "dans le flux", c'est à dire par rapport à la position
26 qu'il aurait dans le conteneur parent si la feuille de style ne concernait pas cette balise.
27 L'absence de décalage indiqué revient à dire que celui-ci est nul par rapport au conteneur
28 parent <body>.
29 Même si le décalage est nul, cette instruction indique que le bloc div est positionné,
30 et donc que des sous blocs (balises enfants) peuvent être positionnés par rapport à lui,
31 en absolu (attribut position : absolute), ou "dans son flux" (attribut position : relative) */
32
33 div.paragraphes {position : relative ; top : 12px; left : 30px;}
34
35 /* On décale de 12 pixels les blocs de paragraphes depuis le haut par rapport à leur position
36 "dans le flux" (position : relative), c'est à dire compte tenu des marges des blocs frères
37 et de la position du bloc parent <div id="page">. De la même façon, on décale
38 ce bloc div de 30 pixels depuis la gauche. */
39
40 h2 {position : relative ; top : 12px; left : 10px;}
41
42 /* On décale de 12 pixels les sous-titres h2 depuis le haut par rapport à leur position
43 "dans le flux" et de 10 pixels depuis la gauche. */
44
```

Cascade Style Sheets F length: 1746 lines: 44 Ln: 38 Col: 26 Sel: 0 | 0 Dos\Windows ANSI as UTF-8 INS

On obtient l'affichage suivant dans Firefox :



Le sélecteur `div#page` renvoie à la balise `<div>` d'identifiant "page".

Le sélecteur `div.paragraphes` renvoie aux balises `<div>` dont la classe est "paragraphes".

Ajouter dans l'accolade de la ligne 22 `top : 50px; left : 60px;`. On voit que toute la page se décale de 50 pixels vers le bas et de 60 pixels vers la gauche. Tous les conteneurs enfants suivent le conteneur parent.

Modifier l'attribut `position : relative;` en `position : absolute;` à la ligne 33. Enregistrer. Rafraîchir la page `index.htm` dans Firefox. Les conteneurs enfants `div.paragraphes` de la page ne font alors plus partie "du flux" et sont placés de façon absolue dans le conteneur parent `div#page`. `top : 0px; left : 0px;` indique l'angle en haut à gauche de ce conteneur.

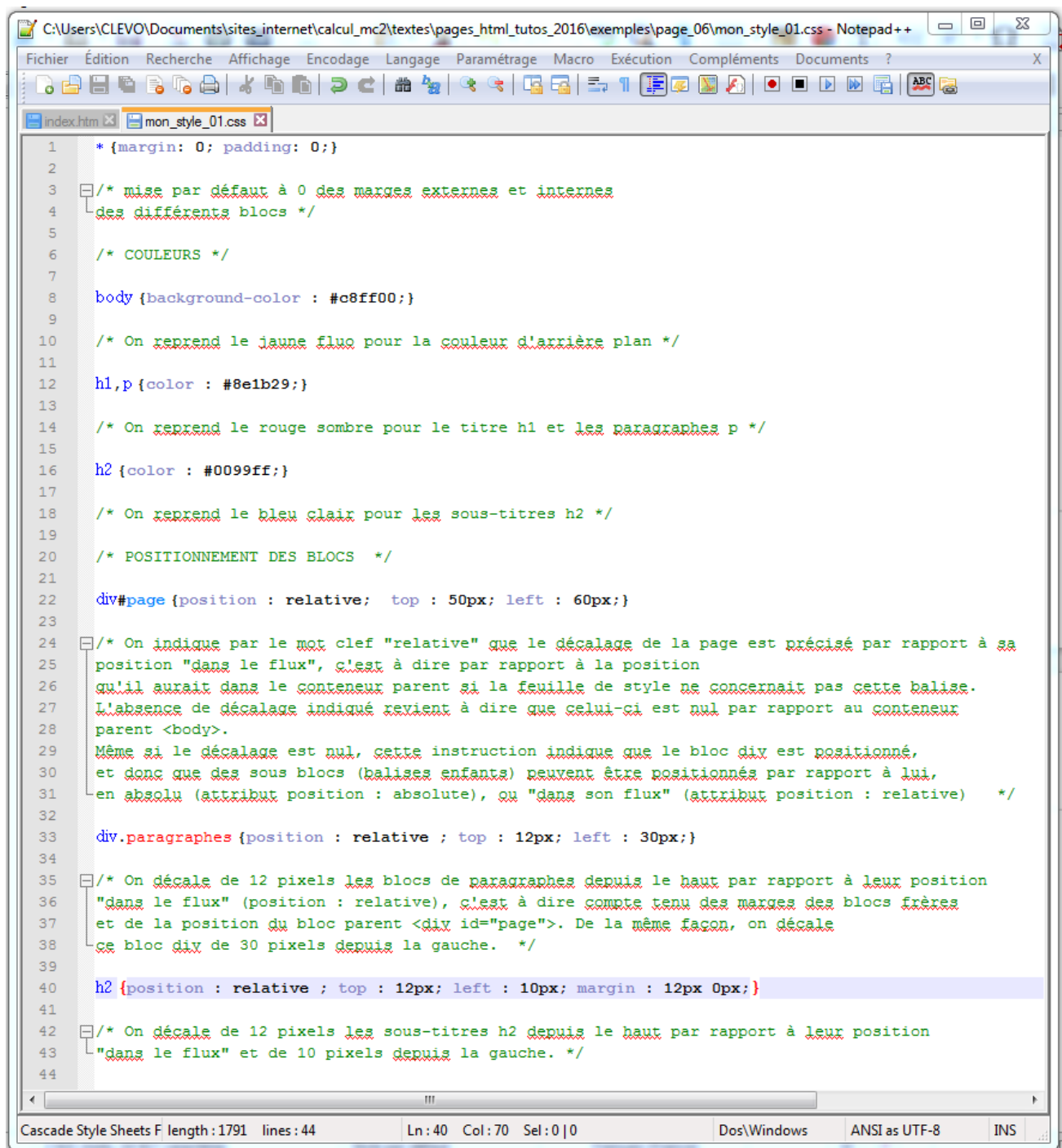
Faire de même à la ligne 40. Enregistrer. Rafraîchir dans Firefox. Bien comprendre ce qui distingue `position : relative` et `position : absolute`. Ces 2 attributs ont chacun leur intérêt.

Rétablir `position : relative;` aux lignes 33 et 40.

Modifier la ligne 40 en ajoutant une marge (interne ou externe) aux titres `h2` :

```
h2 {position : relative; top : 12px; left : 10px; margin : 12px 0px;}
```

On obtient finalement la feuille de style mon_style_01.css suivante :



```
1  * {margin: 0; padding: 0;}
2
3  /* mise par défaut à 0 des marges externes et internes
4  des différents blocs */
5
6  /* COULEURS */
7
8  body {background-color : #c8ff00;}
9
10 /* On reprend le jaune fluo pour la couleur d'arrière plan */
11
12 h1,p {color : #8e1b29;}
13
14 /* On reprend le rouge sombre pour le titre h1 et les paragraphes p */
15
16 h2 {color : #0099ff;}
17
18 /* On reprend le bleu clair pour les sous-titres h2 */
19
20 /* POSITIONNEMENT DES BLOCS */
21
22 div#page {position : relative; top : 50px; left : 60px;}
23
24 /* On indique par le mot clef "relative" que le décalage de la page est précisé par rapport à sa
25 position "dans le flux", c'est à dire par rapport à la position
26 qu'il aurait dans le conteneur parent si la feuille de style ne concernait pas cette balise.
27 L'absence de décalage indiqué revient à dire que celui-ci est nul par rapport au conteneur
28 parent <body>.
29 Même si le décalage est nul, cette instruction indique que le bloc div est positionné,
30 et donc que des sous blocs (balises enfants) peuvent être positionnés par rapport à lui,
31 en absolu (attribut position : absolute), ou "dans son flux" (attribut position : relative) */
32
33 div.paragraphes {position : relative ; top : 12px; left : 30px;}
34
35 /* On décale de 12 pixels les blocs de paragraphes depuis le haut par rapport à leur position
36 "dans le flux" (position : relative), c'est à dire compte tenu des marges des blocs frères
37 et de la position du bloc parent <div id="page">. De la même façon, on décale
38 ce bloc div de 30 pixels depuis la gauche. */
39
40 h2 {position : relative ; top : 12px; left : 10px; margin : 12px 0px;}
41
42 /* On décale de 12 pixels les sous-titres h2 depuis le haut par rapport à leur position
43 "dans le flux" et de 10 pixels depuis la gauche. */
44
```

et le résultat suivant dans Firefox :



8 - Bibliographie

Bien que les navigateurs utilisent actuellement la version 5 du langage HTML, le livre suivant constitue une excellente introduction, pleine d'exemples, pour prolonger ce tutoriel. Le chapitre 13 permet de découvrir le langage JavaScript :

"HTML 4 , Maîtrisez le code source" de Luc VAN LANCKER, éditions ENI, Oct 2008.

- 1 - Présentation
- 2 - Le texte et sa présentation
- 3 - Les tableaux
- 4 - Les liens
- 5 - Les images et arrière-plans
- 6 - Les cadres
- 7 - Les formulaires
- 8 - L'en-tête d'un document Html
- 9 - Images réactives
- 10 - Les feuilles de style
- 11 - Le HTML et le multimédia
- 12 - Le XHTML
- 13 - Les autres langages du WEB

Un autre livre plus récent, excellent pour le langage CSS, est très complet et très clair :

"Premiers pas en CSS3 et HTML5" de Francis DRAILLARD, éditions EYROLLES, 2014.

- 1 - Introduction au HTML et aux feuilles de style CSS
- 2 - L'essentiel du HTML
- 3 - Nouveautés du HTML 5
- 4 - Ecriture des feuilles de style
- 5 - Propriétés de mise en forme
- 6 - Positionnement des blocs
- 7 - Principales nouveautés des CSS3
- 8 - Exemples de sites WEB
- 9 - Un site WEB pour les mobiles
- 10 - Différents types de médias

Deux livres un peu plus anciens, complémentaires, constituent un excellent support pour la programmation en JavaScript :

"JavaScript, des fondamentaux aux concepts avancés", d'Emmanuel GUTIERREZ, Editions ENI, Sept 2008.

- 1 - Bases et introduction à JavaScript
- 2- Insertion du code JavaScript
- 3 - Initiation à la programmation orientée objet et JavaScript
- 4 - Utilisation des constantes, variables et opérateurs
- 5 - Contrôler les scripts avec les structures de contrôle
- 6 - Fonctions et événements
- 7 - Les formulaires
- 8 - Les principaux objets JavaScript en détail
- 9 - Améliorer l'interactivité avec JavaScript et CSS

"Apprendre à développer avec JavaScript", de Christian VIGOUROUX, Editions ENI, Avril 2014.

- 1 - Présentation du langage JavaScript
- 2 - Développement à partir d'algorithmes
- 3 - Bases du langage JavaScript
- 4 - Conditionnement des traitements
- 5 - Traitements itératifs (boucles)
- 6 - Tableaux
- 7 - Procédures et fonctions
- 8 - Approche "objet" en JavaScript
- 9 - Objets de base en JavaScript
- 10 - Saisie de données via des formulaires
- 11 - Modèle DOM
- 12 - Exploration de flux XML via DOM
- 13 - Gestion des cookies en JavaScript
- 14 - Stockage local de données
- 15 - Stockage distant (Ajax - PHP - MySQL -XML)
- 16 - Stockage distant (Ajax - PHP - MySQL - JSON)
- 17 - Géolocalisation
- 18 - Dessin (HTML 5 CANVAS)
- 19 - Graphiques de gestion

20 - Production de documents PDF

21 - Production de codes QR

Enfin, une bible pour le HTML 5 :

"HTML 5, Une référence pour le développeur WEB" de Rodolphe Rimelé, éditions EYROLLES, 2013.

- 1 - Une brève histoire du Web et de ses standards
 - 2 - HTML en seconde langue
 - 3 - Navigateurs et support
 - 4 - Eléments et attributs HTML 5
 - 5 - Les formulaires (Web Forms)
 - 6 - Les microformats (microdata)
 - 7 - Audio et Vidéo
 - 8 - Dessin avec Canvas
 - 9 - Géolocalisation
 - 10 - Interaction avec les fichiers (File API)
 - 11 - Gestion du glisser-déposer (Drag & Drop)
 - 12 - Evénements envoyés par le serveur ("push")
 - 13 - Echange d'informations entre documents (Web Messaging)
 - 14 - Communications en temps réel (Web Sockets)
 - 15 - Stockage des données locales (Web Storage)
 - 16 - Bases de données (Indexed Database & Web SQL Database)
 - 17 - Applications web hors ligne
 - 18 - Historique de navigation
 - 19 - JavaScript en (multi)tâche de fond : les Web Workers
 - 20 - JavaScript, le DOM et l'API Selectors
- Conclusions et perspectives

De nombreux sites internet proposent des applications parfois très spécifiques des langages HTML, CSS et JavaScript.

On peut citer Le site du Zéro, Openclassrooms, qui propose des tutoriels et des livres sur internet, pour différents langages, dont HTML, CSS et JavaScript.