



Création de pages Web

Introduction aux langages HTML et CSS



Les pages affichées par un navigateur internet sur le **World Wide Web** (WWW) sont écrites dans les langages HTML et CSS. Le langage HTML décrit la structure et les contenus à afficher et le langage CSS réalise la mise en forme (présentation et style) de la page web.

Partie A : Le langage HTML

A1. Histoire du HTML

- En 1990, Tim Berners-Lee et Robert Cailliau du Centre Européen de Recherche Nucléaire (CERN) invente le langage HTML (**H**yper **T**ext **M**arkup **L**anguage).
- En 1993, NCSA (National Center for Supercomputing Applications) développe le premier navigateur WEB : MOSAIC.
- En 1996, le W3C (World Wide Web Consortium) définit les marqueurs qui représenteront la norme pour tous les navigateurs et les auteurs de pages Web.
- En 2007, le Web Hypertext Application Technology Working Group (WHATWG), définit les spécifications de HTML5.
- 28 octobre 2014 la standardisation de HTML5 validée par le W3C est publiée.

A2. Introduction


HTML (**H**yper**T**ext **M**arkup **L**anguage) est le langage de description conçu pour représenter les pages Internet. C'est un langage de balises permettant d'écrire de l'hypertexte. HTML permet de structurer le contenu de la page, d'inclure des ressources multimédias (sons, images, vidéos), d'inclure des formulaires de saisie et des programmes informatiques.

A2.1. Prise en main des outils : navigateur Internet et éditeur de texte

Le navigateur Internet est un logiciel qui permet de consulter et d'afficher les pages web. Lorsqu'un utilisateur visite une page web, son navigateur n'affiche pas les balises telles quelles car il analyse le document et l'interprète afin d'afficher la page web correctement. Par exemple, si le document contient une balise image ``, le navigateur chargera l'image associée et affichera l'image à la place de la balise HTML. On parle donc d'un langage interprété (pas de compilation, pas de fichier exécutable).

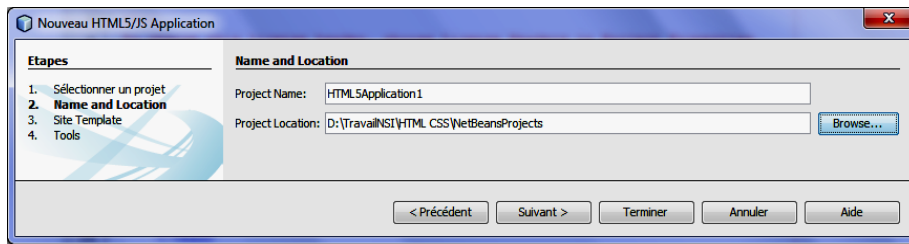
Il existe plusieurs navigateurs comme par exemple Mozilla Firefox, Internet Explorer/Edge, Chrome, Safari, etc...

Les éditeurs de texte sont des logiciels qui permettent d'écrire le code HTML et CSS comme par exemple Notepad++, NetBeans, Sublimtext, etc...

Logiciels utilisés en NSI	
Mozilla Firefox	Navigateur Internet. Permet de visualiser les pages web sur l'écran de l'ordinateur.
 NetBeans	Editeur de texte : Environnement de Développement Intégré (EDI) ou Integrated Development Environment (IDE). Permet de concevoir les pages web.

A2.2. Structure d'un document HTML

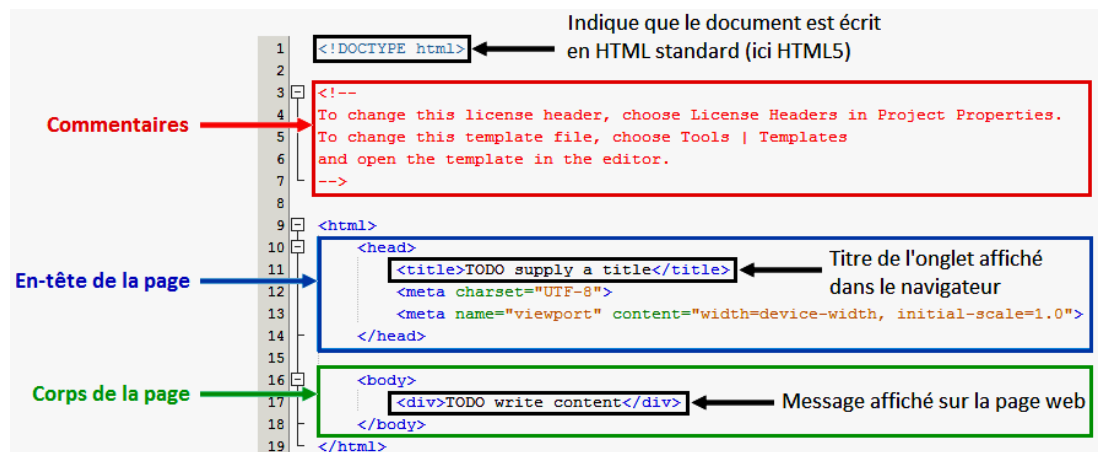
Pour créer une page web avec **NetBeans**, vous devez tout d'abord réaliser un nouveau projet type **HTML5/JS Application**.



Après avoir renseigné le nom du projet, appuyer 2 fois sur suivant pour obtenir le menu **Tools**. Décocher alors l'ensemble des cases de ce menu et cliquer sur Terminer.

Le logiciel NetBeans crée automatiquement le fichier de base **index.html** dans lequel vous allez apporter les modifications pour obtenir votre page web.

En observant le fichier **index.html**, on obtient la structure d'une page web.



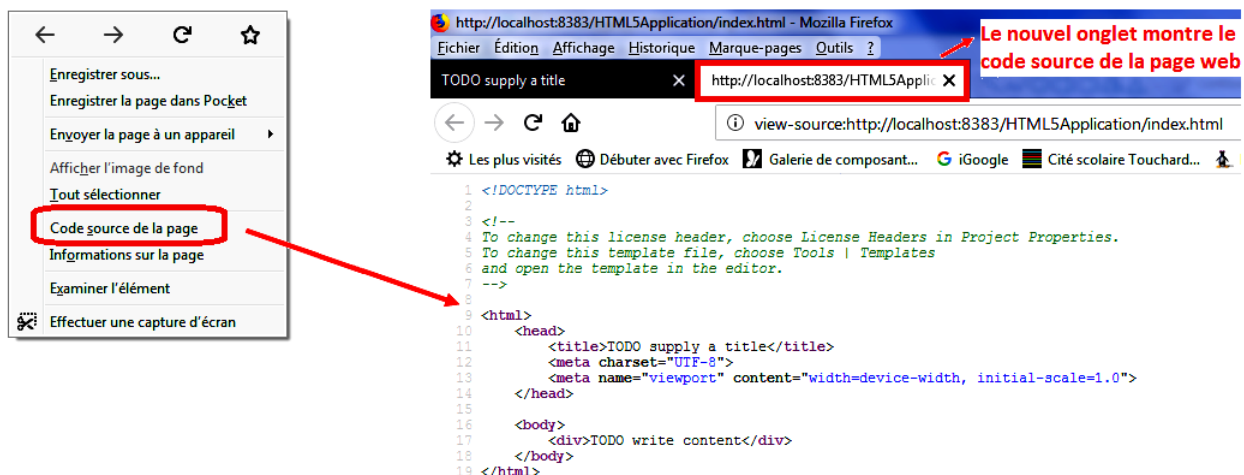
Après avoir choisi le navigateur, on peut afficher directement la page web.



Il est possible de voir le code source de la page et de vérifier la présence des différentes balises HTML.

Pour cela, il faut cliquer sur la page du navigateur avec le bouton droit de la

souris et choisir « Code source de la page » comme le montre l'image ci-dessous.



A2.3. Les premières balises

Le langage HTML est basé sur des balises ou **TAGS** qui permettent de structurer du texte ou d'insérer des images. Une balise est composée d'un chevron ouvrant (<), d'un nom de balise, éventuellement d'un ou plusieurs attributs optionnels pour indiquer comment l'action doit être effectuée, et d'un chevron fermant (>). Le plus souvent, les balises sont ouvrantes et fermantes pour délimiter un bloc. Les balises se présentent donc de la façon suivante :

<marqueur> texte </ marqueur >

ou encore **<marqueur attribut = "argument"> texte </ marqueur >**

ou même **<marqueur />**

Exemples de quelques balises HTML	
<!-- ... -->	Délimite des commentaires pour annoter votre code HTML. Le navigateur n'affiche pas ces commentaires dans la page web. Cela permet de rendre le code source plus lisible.
<html> ... </html>	Délimite le texte HTML dans un fichier.
<head> ... </head>	Délimite l'entête du fichier.
<title> ... </title>	Délimite le titre de la page. S'affiche dans l'onglet de la page.
<body> ... </body>	Délimite le corps du fichier. Contient le contenu de la page.
<div> ... </div>	Définit un conteneur ou un bloc auquel on pourra appliquer un certain nombre de propriétés.
<h1> ... </h1> <h2> ... </h2>	Délimite un titre de paragraphe. Il peut y avoir jusqu'à 6 niveaux de titre h1, h2, h3 ... , h6.
<p> ... </p>	Délimite un paragraphe. Les sauts de ligne sont automatiques à la fin du paragraphe.
 ... 	Délimite un texte en gras.

	Impose un saut de ligne volontaire.
&nbsp;	Force l'affichage d'un espace.

Pour une meilleure lisibilité et éviter les erreurs de syntaxe, le langage HTML doit être indenté, c'est à dire, chaque nouvelle balise imbriquée fait l'objet d'un décalage.

Exemple 1

```

7  <html>
8    <head>
9      <title>Module 1</title>
10     <meta charset="UTF-8">
11     <meta name="viewport" content="width=device-width">
12   </head>
13   <body>
14     <div>
15       <h1>
16         Création de sites Internet
17       </h1>
18       <p>
19         Voici un premier paragraphe
20       </p>
21     </div>
22   </body>
23 </html>

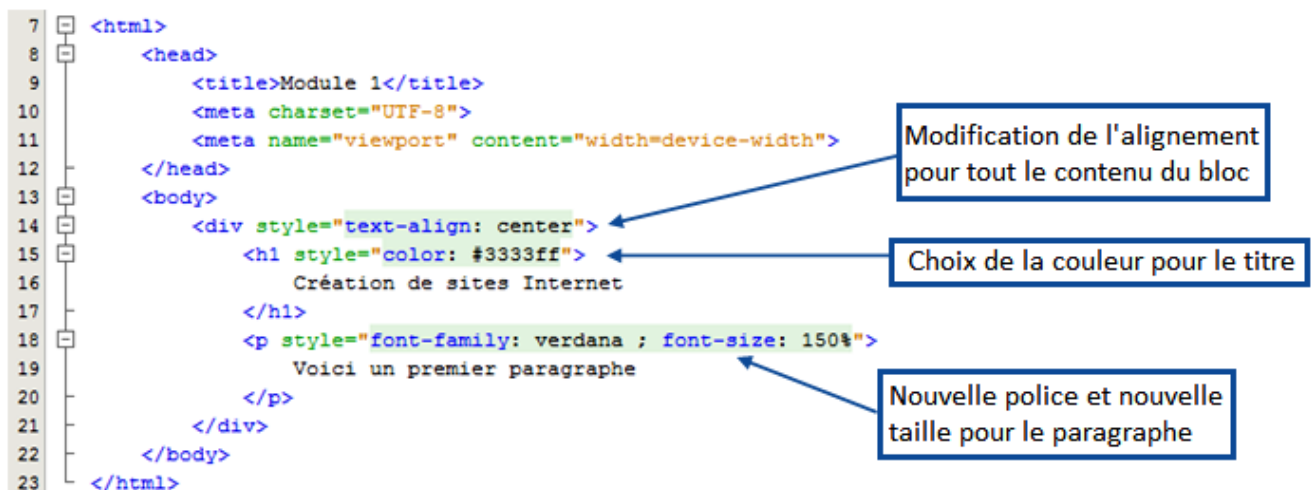
```



On constate que la balise **<h1>** définit un certain style (police, taille, alignement...) qu'il sera possible de personnaliser. Il en est de même pour le paragraphe **<p>**.

A2.4. Modification du style dans le source HTML

Exemple 2



L'attribut **style** permet de modifier la couleur, la manière d'aligner le texte, la taille du texte... Lorsque le style est appliqué au conteneur `<div> ... </div>`, il s'applique alors à toutes les balises qui se trouvent à l'intérieur du conteneur.

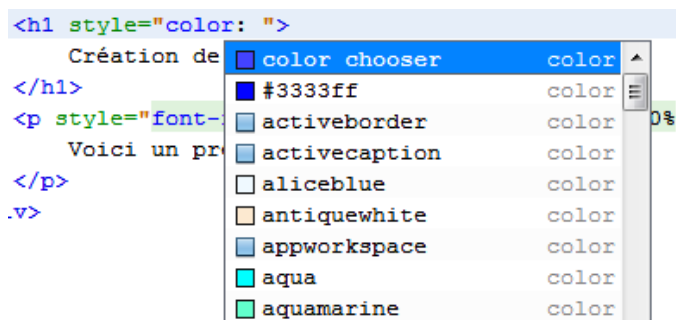
Grâce à l'outil d'auto complétion avec la combinaison des touches **Ctrl <Espace>**, le logiciel **NetBeans** permet de suggérer les possibilités des balises pour terminer une ligne déjà commencée.

L'attribut **color** change la couleur. Il utilise comme valeur un nombre hexadécimal représentant la valeur des couleurs (**rouge**, **vert**, **bleu**) sous la forme `#rrvvbb`. Dans l'exemple, la couleur fixée est bleue. Chaque couleur est codée en hexadécimale. Cette numérotation utilise un système de numérotation en base 16, les chiffres de 0 à 9 et les lettres A, B, C, D, E, F.

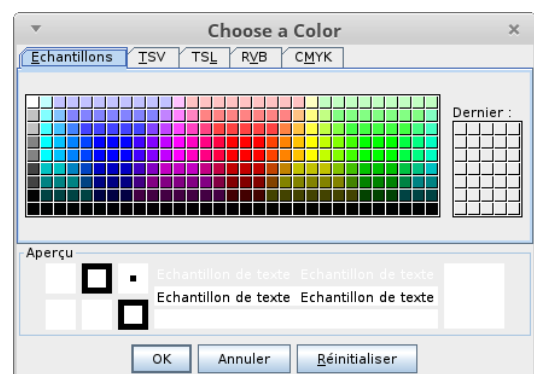
Lorsque la valeur est sur deux caractères hexadécimaux, soit de **00** à **FF**, elle varie entre 0 et 255 en décimal. En agissant sur les trois couleurs, les différentes nuances vont du noir `#000000` au blanc `#FFFFFF`.

Pour le choix de la couleur, NetBeans propose un menu ou un outil pour choisir la couleur. La combinaison des touches **Ctrl <Espace>** après l'attribut **color** permet de faire le choix de la couleur dans le menu suivant ou fait apparaître l'outil de choix de couleur.

Menu



Outil de choix de couleur



A2.5. Incorporation d'une image dans la page web

Le langage HTML permet également de faire apparaître des images.

Balise HTML	
<code></code>	Charge une image dans la page web

Exemple 3

```

```



L'attribut **src** permet de préciser le nom de l'image à charger. Le fichier **isn.jpg** représente donc le nom de cette image. Elle doit être dans le même dossier que le fichier **index.html** ou sinon le chemin doit être spécifié complètement.

L'attribut **alt** permet d'indiquer un message alternatif si le fichier image n'est pas trouvé.

La propriété **width** de l'attribut **style** définit la largeur de l'image en pourcentage de l'image originale.

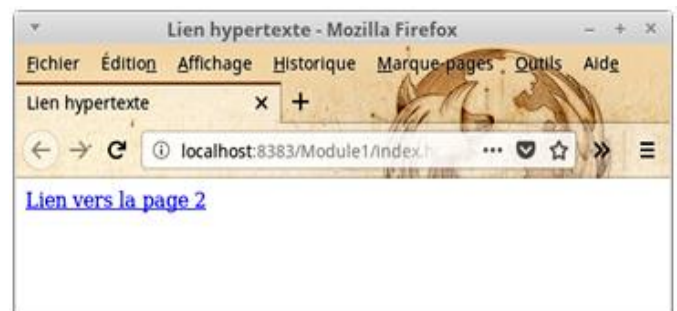
A2.6. Création d'un lien hypertexte

Le langage HTML permet de créer des liens vers d'autres pages web ou vers d'autres sites.

Balise HTML	
<code> </code>	Créer un lien dans la page web vers un site extérieur ou vers une autre page.

Exemple 4

```
<a href="page2.html">Lien vers la page 2</a>
```



Dans cet exemple, la balise **<a>** et l'attribut **href** permettent de créer un lien vers une autre page html de votre site. Le texte entre les balises **<a>** et **** apparaît sur la page web pour matérialiser le lien mis en forme par défaut comme tous les liens hypertextes (souligné, bleu).

La propriété **page2.html** représente l'url de la page à afficher après avoir cliquer sur le lien.

Pour un lien externe, le principe reste identique : l'adresse est précisée comme elle le serait dans la barre d'url du navigateur.

```
<a href="https://touchard-washington.paysdelaloire.e-lyco.fr">Lien vers le lycée</a>
```


A3. Insérer des tableaux dans une page web

Un tableau est composé de lignes et de colonnes ou plus précisément en HTML de rangées (**table row**) contenant des données (**table data**).

A3.1. Les balises de base

Les tableaux sont définis entre les balises `<table> ... </table>`.

Les tableaux sont divisés en rangées ou lignes avec la balise `<tr> ... </tr>`.

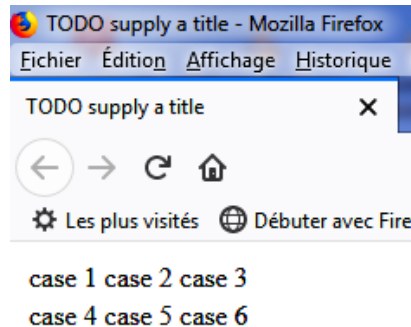
Les rangées sont subdivisées en zone de données avec les balises `<td> ... </td>`. Ces zones de données peuvent contenir du texte, des images, des listes ou encore un autre tableau...

Exemple 5

```

1 <table>
2 <tr>
3   <td>case 1</td>
4   <td>case 2</td>
5   <td>case 3</td>
6 </tr>
7 <tr>
8   <td>case 4</td>
9   <td>case 5</td>
10  <td>case 6</td>
11 </tr>
12 </table>

```



Cette partie de code HTML crée un tableau avec deux lignes et 3 colonnes.

Chaque cellule contient le texte case x où x représente le numéro de la case.

Il est possible d'ajouter autant de lignes et de colonnes que nécessaire avec les balises `<tr>` et `<td>`.

Attention, aucun cadre n'apparaît pour délimiter les cellules. Pour mettre en forme un tableau, il faut modifier le style du tableau.

A3.2. Mise en forme du tableau

Pour effectuer la mise en forme d'un tableau, on ajoute l'attribut **style** dans la balise `<table>`. Avec la propriété **width: 100 %**, le tableau occupe maintenant 100 % de la largeur de la page.

```

<table style="width: 100%">
  ...
</table>

```

Il est possible également d'exprimer cette largeur en pixel avec **width: 200px**. Dans ce cas, le tableau n'est plus proportionnel à largeur de la page mais possède une dimension fixe quelque soit le navigateur et la résolution de l'écran (200 pixels dans notre exemple).

Un pixel représente le plus petit point lumineux sur l'écran. Il permet de mesurer la définition d'une image numérique.

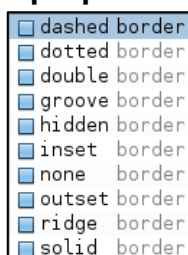
Pour tracer un cadre autour du tableau, on utilise la propriété **border** dans l'attribut **style**. Le chiffre donne le nombre de pixels fixant l'épaisseur du cadre. Son type de trait et sa couleur sont également mentionnés. Dans cet exemple, seul le tableau est encadré.

```

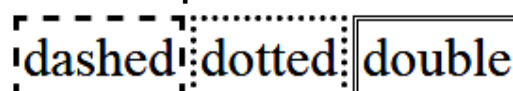
<table style="border: 1px solid black ; width: 100%">
  ...
</table>

```

Bordures proposées par NetBeans



Exemples de bordures



Pour encadrer les cellules, il faut ajouter l'attribut **style** et la propriété **border** à chaque cellule, c'est-à-dire dans chaque balise **<td>**.

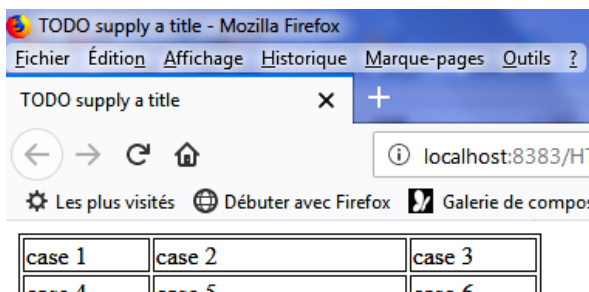
Pour définir la largeur des colonnes, il faut agir sur les balises **<td>** de la première ligne en affectant à l'attribut **style** la valeur de la largeur. Attention, la somme des largeurs de colonnes ne doit pas dépasser les 100 % lorsque l'on travaille de manière proportionnelle.

Exemple 6

```
<table style="border: 1px black solid; width: 25%">
  <tr>
    <td style="border: 1px black solid; width: 25%">case 1</td>
    <td style="border: 1px black solid; width: 50%">case 2</td>
    <td style="border: 1px black solid; width: 25%">case 3</td>
  </tr>
  <tr>
    <td style="border: 1px black solid">case 4</td>
    <td style="border: 1px black solid">case 5</td>
    <td style="border: 1px black solid">case 6</td>
  </tr>
</table>
```

Largeur du tableau dans la page

Largeur des colonnes



Le fait d'ajouter la propriété **border-collapse** détermine si les bordures d'un tableau sont séparées ou fusionnées. La valeur **collapse** permet de les fusionner et la valeur **separate** permet de les séparer (valeur par défaut).

Exemple 7

```
<table style="border: 1px black solid; width: 25%; border-collapse: collapse">
  <!-- Définition du tableau -->
</table>
```

case 1	case 2	case 3
case 4	case 5	case 6

A3.3. Regroupement de cellules

L'attribut **colspan** d'une balise **<td>** permet de regrouper des cellules horizontalement. Il faut préciser le nombre de cellules à fusionner.

Exemple 8

```
<table style="border: 1px black solid; width: 25%; border-collapse: collapse">
  <tr>
    <td style="border: 1px black solid; width: 25%">case 1</td>
    <td style="border: 1px black solid; width: 50%">case 2</td>
    <td style="border: 1px black solid; width: 25%">case 3</td>
  </tr>
  <tr>
    <td style="border: 1px black solid" colspan="2">case 4</td>
    <td style="border: 1px black solid">case 5</td>
  </tr>
</table>
```

2 colonnes fusionnées

case 1	case 2	case 3
case 4		case 5

Dans cet exemple, la deuxième ligne ne comporte que deux colonnes, les deux premières étant regroupées. Elle occupe dans le cas présent 75 % de la largeur du tableau.

L'attribut **rowspan** d'une balise `<td>` permet de regrouper des cellules verticalement. Il faut préciser le nombre de cellules à fusionner.

Exemple 9

```
<table style="border: 1px black solid; width: 25%; border-collapse: collapse">
  <tr>
    <td style="border: 1px black solid; width: 25%">case 1</td>
    <td style="border: 1px black solid; width: 50%">case 2</td>
    <td style="border: 1px black solid; width: 25%">case 3</td>
  </tr>
  <tr>
    <td style="border: 1px black solid" rowspan="2">case 4</td>
    <td style="border: 1px black solid">case 5</td>
    <td style="border: 1px black solid">case 6</td>
  </tr>
  <tr>
    <td style="border: 1px black solid">case 7</td>
    <td style="border: 1px black solid">case 8</td>
  </tr>
</table>
```

2 lignes
fusionnées

case 1	case 2	case 3
case 4	case 5	case 6
	case 7	case 8

A3.4. En-tête de tableau

La balise `<th>` peut remplacer une balise `<td>` pour définir un style particulier sur la première ligne ou sur la première colonne et ainsi mettre en évidence un en-tête au tableau. Par défaut, le texte entre les balises `<th> ... </th>` est centré et gras.

Exemple 10

```
<table style="border: 1px black solid; width: 25%; border-collapse: collapse">
  <tr>
    <th style="border: 1px black solid; width: 25%">En-tête 1</th>
    <th style="border: 1px black solid; width: 50%">En-tête 2</th>
    <th style="border: 1px black solid; width: 25%">En-tête 3</th>
  </tr>
  <tr>
    <td style="border: 1px black solid">case 1</td>
    <td style="border: 1px black solid">case 2</td>
    <td style="border: 1px black solid">case 3</td>
  </tr>
  <tr>
    <td style="border: 1px black solid">case 4</td>
    <td style="border: 1px black solid">case 5</td>
    <td style="border: 1px black solid">case 6</td>
  </tr>
</table>
```

En-tête de
la 1ère ligne

En-tête 1	En-tête 2	En-tête 3
case 1	case 2	case 3
case 4	case 5	case 6

A3.5. Légende d'un tableau

La balise `<caption>` permet d'ajouter une légende à un tableau. Elle est placée après la balise `<table>` définissant le tableau.

Exemple 11

```
<table style="border: 1px black solid; width: 25%; border-collapse: collapse">
  <caption> Légende</caption>
```

Légende du tableau

Légende		
En-tête 1	En-tête 2	En-tête 3
case 1	case 2	case 3
case 4	case 5	case 6

A4. Insérer des listes dans une page web

Une liste permet de structurer un texte et d'ordonner des informations. Il existe deux types de liste :

- Les listes à puce ou non-ordonnées.
- Les listes numérotées ou ordonnées.

A4.1. Les listes à puce (listes non-ordonnées)

Elles sont définies entre les balises ` ... `. Chaque item est ensuite encadré par les balises ` ... `.

Exemple 12

```
<ul>
  <li>Premier point</li>
  <li>Deuxième point</li>
  <li>Troisième point</li>
</ul>
```

- Premier point
- Deuxième point
- Troisième point

Il existe plusieurs styles de listes à puces. Pour modifier la valeur du style par défaut (disque noir rempli), il faut ajouter à la balise `` l'attribut `style` et la propriété `list-style-type`.

Toutes les possibilités sont proposées par l'auto complétion de NetBeans avec les touches **Ctrl <Espace>**. Voici quelques exemples possibles :

<code><ul style="list-style-type: circle"></code>	cercle	○
<code><ul style="list-style-type: square"></code>	carré	■
<code><ul style="list-style-type: none"></code>	sans	

A4.2. Les listes numérotées (listes ordonnées)

Elles sont définies entre les balises ` ... `. Chaque item est ensuite encadré par les balises ` ... `.

Exemple 13

```
<ol>
  <li>Premier point</li>
  <li>Deuxième point</li>
  <li>Troisième point</li>
</ol>
```

1. Premier point
2. Deuxième point
3. Troisième point

Le style par défaut d'une liste numérotée est un numéro qui apparaît devant chaque élément constituant la liste. Pour modifier la valeur du style, il faut ajouter à la balise `` l'attribut `style` et la propriété `list-style-type`. Toutes les possibilités sont proposées par l'auto complétion de NetBeans. Voici quelques exemples possibles :

<code><ol style="list-style-type: decimal"></code>	Un numéro
<code><ol style="list-style-type: upper-alpha"></code>	Une lettre en majuscule
<code><ol style="list-style-type: lower-alpha"></code>	Une lettre en minuscule
<code><ol style="list-style-type: upper-roman"></code>	Un chiffre romain en majuscule
<code><ol style="list-style-type: lower-roman"></code>	Un chiffre romain en minuscule

Bien d'autres alphabets encore peuvent être utilisés : grecque, arménien, hébreux...

A4.3. Les listes imbriquées

De la même manière, il est possible d'imbriquer des listes pour créer ainsi des décalages entre les listes. C'est entre les balises ` ... ` que doit être décrite la nouvelle sous liste.

Exemple 14

```
<ul>
  <li>Premier point
    <ul>
      <li>Premier sous point</li>
      <li>Deuxième sous point</li>
      <li>Troisième sous point</li>
    </ul>
  </li>
  <li>Deuxième point</li>
  <li>Troisième point</li>
</ul>
```

- Premier point
 - Premier sous point
 - Deuxième sous point
 - Troisième sous point
- Deuxième point
- Troisième point

Partie B : Les feuilles de styles - CSS

B1. Introduction

Nous avons vu jusqu'à maintenant que le style d'une page web pouvait être inclus dans le code HTML. Pour simplifier la lisibilité de la page HTML, il est préférable de regrouper le style de la page dans un fichier **CSS** spécifique.

CSS (Cascading Style Sheets) pour feuilles de style en cascade est le langage pour décrire la présentation et le style des pages web. Il décrit comment les éléments d'un document HTML vont être mis en forme pour un affichage à l'écran, sur papier ou sur tout autre média.

Ainsi, pour un même contenu HTML, il sera facile d'obtenir différentes versions de mise en page en fonction du navigateur, de la taille de l'écran, du type de périphériques (ordinateur, tablette, smartphone).

La feuille de style est incluse dans l'entête de la page HTML avec la balise `<link>`.

```
<head>
  <title>TODO supply a title</title>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link href="monStyle.css" rel="stylesheet" type="text/css"/>
</head>
```

↑ Fichier style .css

Le fichier **monStyle.css** contient toutes les déclarations pour le style de cette page HTML.

B2. Syntaxe

Le **CSS** correspond à un ensemble de **propriétés** de style appliqué à un **sélecteur**. Les propriétés sont séparées par un point-virgule (;).

```
sélecteur de balise {
  propriété1 de style: valeur;
  propriété2 de style: valeur;
  ...
}
```

Un sélecteur peut être désigné par un nom de balise HTML, un identifiant unique ou un ensemble d'éléments désigné par une classe.

B2.1. Le sélecteur est une balise HTML (nom balise)

Lorsque le sélecteur est une balise HTML, le style s'applique à toutes les balises de même nom pour toutes les pages où le fichier **CSS** est inclus.

Exemple 15

Sélecteur par nom de balise

Propriétés de style

```

body {
  background-color: greenyellow;
}
h1 {
  color: red;
  font-family: serif;
  font-size: 20px;
  margin-left: 20px;
}
  
```

```

<html>
<head>
  <title>TODO supply a title</title>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link href="monStyle.css" rel="stylesheet" type="text/css"/>
</head>
<body>
  <h1>Voici le premier titre</h1>
</body>
</html>
  
```

Voici le premier titre

Dans cet exemple, tous les éléments placés entre la balise **<body>** seront sur un fond de couleur vert. Le texte contenu entre les balises **<h1>** aura comme propriétés une couleur rouge et une taille de police de 20 pixels.

B2.2. Le sélecteur est un identifiant (#id)

L'**id** d'un élément HTML permet d'identifier de manière unique une balise HTML. Il ne peut être utilisé qu'une seule fois dans une même page. Dans le fichier **CSS**, l'identifiant est précédé par un dièse (#).

Exemple 16

Fichier CSS

```

#menu {
  position: absolute;
  top: 30px;
  left: 5px;
  width: auto;
}
  
```

Fichier HTML

```

<div id="menu">
  texte
</div>
  
```

Identifiant menu placé dans le code HTML

Ici, le conteneur définit par la balise **<div>** à un identifiant spécifique nommé **menu**. Il permet d'appliquer des propriétés particulières à cette portion de code et non pas à toutes les balises **<div>** de la page.

Dans cet exemple, le texte placé dans la balise **<div>** est toujours à 30 pixels du haut de la page et 5 pixels à gauche. Sa largeur s'adapte de manière automatique à la largeur de la page.

Il est également possible de désigner une balise particulière dans cette section de code sans affecter les autres balises du même nom dans le reste de la page. Il est possible d'ajouter des règles spécifiques à d'autres balises incluses dans les premières.

Exemple 17

Fichier CSS

```

#menu {
  list-style-type: none;
}
#menu ul li {
  padding: 5px;
  background-color: aqua;
  border-radius: 5px;
  margin: 5px;
  width: 50px;
}
  
```

Fichier HTML

```

<body>
  <h1>Voici le premier titre</h1>
  <div id="menu">
    <ul>
      <li>menu 1</li>
      <li>menu 2</li>
      <li>menu 3</li>
    </ul>
    <ul>
      <li>menu 4</li>
      <li>menu 5</li>
    </ul>
  </div>
</body>
  
```

Voici le premier titre

menu 1
menu 2
menu 3
• menu 4
• menu 5

Ici, seules les balises **** incluses dans la balise **** de l'identifiant **menu** ont un aspect particulier.

Ainsi, toutes les balises **** du menu peuvent avoir les mêmes propriétés

sans pour autant affecter les autres balises **ul** de la page en dehors du conteneur **menu** (seuls les items du menu n'ont plus de disque et ont une couleur différente).

B2.3. Le sélecteur est une classe (.classe)

Une classe **CSS** désigne un ensemble d'éléments HTML utilisant le même sélecteur. Dans le fichier **CSS**, le nom de la classe est précédé par un point (.).

Exemple 18

```
body {
    background-color: greenyellow;
}

.important {
    margin: 10px;
    padding: 5px;
    background-color: lightsteelblue;
    border: 1px black solid;
    text-align: left;
    width: 100px;
}
```

```
<body>
  <h1>
    Voici le premier titre
  </h1>

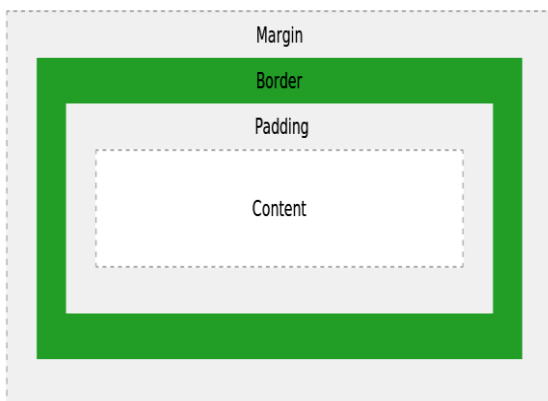
  <div>
    <p class="important">
      Voici un texte important
    </p>
    <p>
      texte moins important
    </p>
    <p class="important">
      Un autre texte important
    </p>
  </div>
</body>
```



Plusieurs paragraphes peuvent utiliser la classe **important** pour mettre en valeur le texte.

B3. Modèle en boîte CSS

Tout élément HTML peut être considéré comme une boîte qui peut être représentée par l'image suivante.



Nom	Traduction	Rôle
Content	<i>Contenu</i>	C'est le contenu de la boîte, cela peut être du texte, une image...
Padding	<i>Rembourrage</i>	C'est l'espace entre le contenu et la bordure. Il est transparent.
Border	<i>Bordure</i>	La bordure entoure le rembourrage et le contenu. Elle peut prendre différents aspects en termes de couleur, de style, de trait et d'épaisseur.
Margin	<i>Marge</i>	C'est l'espace entre la bordure et les autres éléments HTML adjacents à la boîte. Elle est transparente.

Par défaut, les propriétés **margin** et **padding** agissent dans les quatre directions (haut, bas, gauche, droite).

La propriété **width** précise la largeur de la boîte et la propriété **height** détermine la hauteur (exprimées en pixels ou en pourcentage).

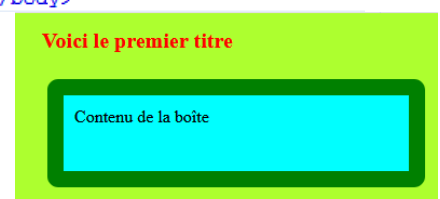
La propriété **border-radius** permet d'arrondir l'angle de la bordure (exprimée en pixels).

La propriété **background-color** agit sur la couleur de fond de la boîte.

Exemple 19

```
div {
    border: 15px solid green;
    border-radius: 10px;
    padding: 10px;
    margin: 25px;
    width: 300px;
    height: 50px;
    background-color: aqua;
}
```

```
<body>
  <h1>
    Voici le premier titre
  </h1>
  <div>
    Contenu de la boîte
  </div>
</body>
```



Il est tout à fait possible de modifier la marge uniquement sur la gauche par exemple en agissant sur la propriété **margin-left**. Les quatre directions sont désignées par les termes **margin-top** (haut), **margin-left** (gauche), **margin-right** (droite) et **margin-bottom** (bas).

Ici on désigne explicitement la marge de gauche.

Ici on précise la valeur de chacune des marges (haut, gauche, droite, bas).

```
#contenu {
  margin-left: 110px;
}
```

```
p {
  margin: 25px 50px 75px 100px;
}
```

B4. Positionnement

La propriété **position** permet de spécifier la manière dont un élément HTML est positionné sur l'écran. Elle peut être soit statique, relative, fixe, absolue, collante ou superposée.

B4.1. Position statique (static)

Les éléments HTML sont positionnés statiquement par défaut. Ils ne sont pas affectés par les propriétés **top**, **bottom**, **left** et **right**. Dans ce cas, ils ne sont pas positionnés de manière particulière et sont toujours positionnés en fonction du flux normal de la page.

Exemple 20

```
div.statique {
  position: static;
  padding: 20px;
  border: 3px solid coral;
  width: 310px;
}
```

```
<div class="statique">
  Voici un élément positionné de manière statique
</div>
```

Voici le premier titre

Voici un élément positionné de manière statique

B4.2. Position relative (relative)

Un élément dont la propriété **position** possède la valeur **relative** est positionné de manière relative à la position qu'il aurait normalement occupée. Il est possible d'agir dans les quatre directions avec les propriétés **top**, **bottom**, **left** ou **right**.

Dans l'exemple suivant, un décalage de 30 pixels sur la gauche est ajouté par rapport à la position de la première phrase positionnée de manière statique.

Exemple 21

```
div.relative {
  position: relative;
  border: 3px solid coral;
  left: 30px;
  padding: 20px;
  width: 310px;
}
```

```
<div class="relative">
  Voici un élément positionné de manière relative
</div>
```

Voici le premier titre

Voici un élément positionné de manière statique

Voici un élément positionné de manière relative

B4.3. Position fixe (fixed)

Un élément positionné de manière fixe l'est par rapport à la fenêtre d'affichage. Ainsi il reste au même endroit même si la fenêtre défile. Les propriétés **top**, **bottom**, **left** ou **right** affecte la position de l'élément.


```
div.fixe {
  position: fixed;
  padding: 20px;
  border: 3px solid coral;
  right: 10px;
  bottom: 10px;
  width: 310px;
}
```

C'est la solution pour positionner un menu, des boutons ou des liens toujours au même emplacement sur l'écran.

B4.4. Position absolue (absolute)

Un élément positionné de manière absolue l'est par rapport à l'élément qui le contient, ou, ancêtre le plus proche contrairement au positionnement fixe qui positionne par rapport à la fenêtre d'affichage. Si un élément positionné en absolu n'a pas d'ancêtre, il est positionné par rapport à la page qui est affichée et est affecté par le défilement.

Exemple 22

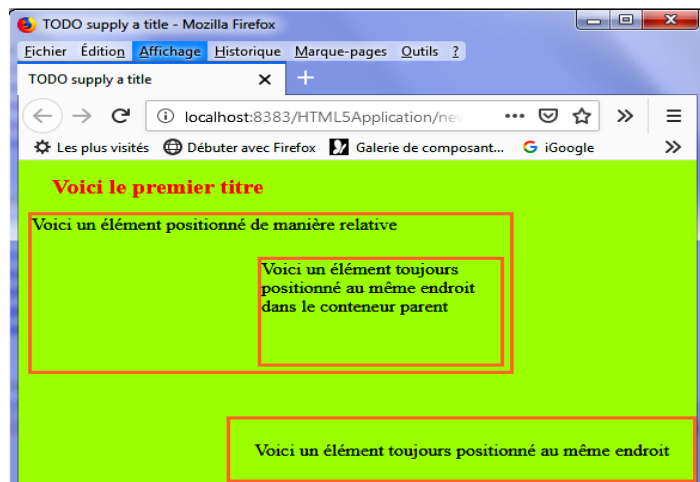
```
div.relative {
  position: relative;
  border: 3px solid coral;
  height: 200px;
  width: 400px;
}

div.fixe {
  position: fixed;
  padding: 20px;
  border: 3px solid coral;
  right: 0px;
  bottom: 0px;
}

div.absolue {
  position: absolute;
  top: 80px;
  right: 5px;
  bottom: 5px;
  width: 200px;
  height: 100px;
  border: 3px solid coral;
}
```

```
<body>
<h1>
  Voici le premier titre
</h1>

<div class="relative">
  Voici un élément positionné de manière relative
  <div class="absolue">
    Voici un élément toujours positionné au même endroit dans le
    conteneur parent
  </div>
</div>
<div class="fixe">
  Voici un élément toujours positionné au même endroit
</div>
</body>
```



B4.5. Position collante (sticky)

Un élément positionné de manière collante est placé en fonction de la position de défilement de l'utilisateur. Il bascule entre un positionnement relatif et fixe, en fonction de la position de défilement. Il est positionné de manière relative jusqu'à ce qu'une position de décalage donnée soit satisfaite dans la fenêtre d'affichage, puis il reste en place, comme pour la position fixe.

```
div.collante {
  position: sticky;
  position: -webkit-sticky;
  border: 2px solid cardetblue;
  padding: 5px;
  top: 30px;
  width: 200px;
  background-color: beige;
}
```

La valeur **sticky** n'est pas toujours reconnu, comme le montre l'éditeur de NetBeans. Il est parfois nécessaire de mettre **-webkit-sticky** à la place ou les deux pour que le code s'adapte à différents navigateurs.

B4.6. Positionnement superposé

Lorsque plusieurs éléments se superposent, par exemple du texte sur une image, il est important de pouvoir choisir celui qui doit se trouver au premier plan. La propriété **z-index** répond à ce besoin.

La propriété **z-index** peut prendre des valeurs positive ou négative. La plus grande valeur sera au premier plan et la plus petite sera plus derrière.

```
img {
  position: absolute;
  top: 0px;
  left: 0px;
  z-index: -1;
}
```

Exemple 23

```
<body>
  <h1>
    Voici le premier titre
  </h1>
  
</body>
```

Comme l'image a un **z-index** à -1, elle est placée derrière le texte.



B5. Alignement

B5.1. Alignement horizontal d'un texte

La propriété **text-align** permet d'aligner un texte horizontalement, soit sur la page, soit dans le conteneur qui le reçoit. Les valeurs courantes sont :

text-align: center;	Pour centrer le texte
text-align: left;	Pour l'aligner à gauche (valeur par défaut)
text-align: right;	Pour l'aligner à droite
text-align: justify;	Pour justifier le texte

B5.2. Alignement vertical

La propriété **vertical-align** permet d'aligner un élément verticalement, par exemple dans une cellule d'un tableau. Les valeurs courantes sont :

vertical-align: middle;	Pour centrer sur la hauteur
vertical-align: top;	Pour l'aligner en haut (valeur par défaut)
vertical-align: bottom;	Pour l'aligner en bas

B5.3. Centrer une image ou un conteneur

Voici deux exemples permettant de centrer en utilisant le réglage des marges de manière automatique. Pour cela, il faut fixer la largeur de l'élément, la valeur **auto** ajuste ensuite en fonction de la place disponible.

```
.center {
  margin: auto;
  width: 50%;
  border: 2px solid green;
  padding: 10px;
}
```

```
img {
  margin-left: auto;
  margin-right: auto;
  width: 40%;
}
```