

Programmation langage objet

Création d'une page html

Site de référence :

<http://fr.openclassrooms.com/informatique/cours/apprenez-a-creer-votre-site-web-avec-html5-et-css3>

1. Structure d'une page html

1.1. Les balises

Télécharger Notepad++ et l'installer

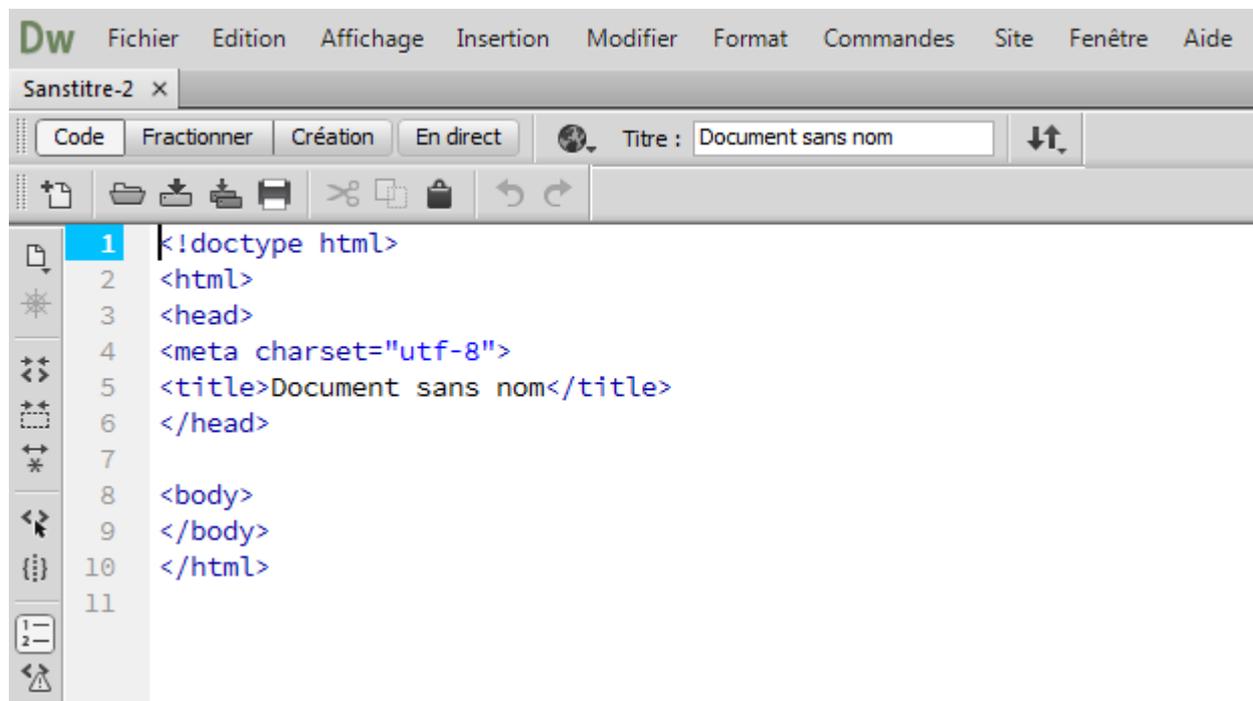
http://www.01net.com/telecharger/windows/Internet/editeur_de_site/fiches/29119.html

Copier le texte suivant dans notepad

```
<!DOCTYPE html>
2<html>
3  <head>
4  <link rel="icon" type="image/png" href="icone.png" />
5    <meta charset="utf-8" />
6    <title>Titre</title>
7  </head>
8
9  <body>
10
11 </body>
12</html>
```

Capturer l'image ci-dessous au format 72x72 et enregistrer la sous le même répertoire que le fichier html que vous avez créé sous le nom icone.png





- **Le doctype**

```
1<!DOCTYPE html>
```

La toute première ligne s'appelle le **doctype**. Elle est indispensable car c'est elle qui indique qu'il s'agit bien d'une page web HTML.

- **La balise </html>**

```
1<html>
2
3</html>
```

C'est la balise principale du code. Elle englobe tout le contenu de votre page. Comme vous pouvez le voir, la balise fermante </html> se trouve tout à la fin du code !

- **L'en-tête <head> et le corps <body>**

Une page web est constituée de deux parties :

- L'en-tête <head> : cette section donne quelques informations générales sur la page comme son titre, l'encodage (pour la gestion des caractères spéciaux), etc. Cette section est généralement assez courte. Les informations que contient l'en-tête ne sont pas affichées sur la page, ce sont simplement des informations générales à destination de l'ordinateur. Elles sont cependant très importantes !
- Le corps <body> : c'est là que se trouve la partie principale de la page. Tout ce que nous écrivons ici sera affiché à l'écran. C'est à l'intérieur du corps que nous écrivons la majeure partie de notre code.
- **L'encodage (charset)**

```
1<meta charset="utf-8" />
```

Cette balise indique l'encodage utilisé dans votre fichier .html.

Sous Notepad++, allez dans le menu Encodage > Encoder en UTF-8 (sans BOM) pour que votre fichier soit enregistré en UTF-8 dès le début. Cela ne s'applique qu'au fichier actuellement ouvert. Pour ne pas avoir à le faire pour chaque nouveau fichier, je vous conseille d'aller dans le menu Paramétrage > Préférences, onglet Nouveau document/Dossier. Sélectionnez UTF-8 sans BOM dans la liste.

- Le titre principal de la page

1 <title>

C'est le titre de votre page, probablement l'élément le plus important ! Toute page doit avoir un titre qui décrit ce qu'elle contient.

Il est conseillé de garder le titre assez court (moins de 100 caractères en général).



```
<!DOCTYPE html>
<html>
<head>
<link rel="icon" type="image/png" href="icone.png" />
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=Edge"/>
  <title>cours et tp sti2d</title>
```

- L'icône (Link)

4 <link rel="icon" type="image/png" href="icone.png" />

Cette balise indique le lien vers l'icône s'affichant sur la page html. Il doit se trouver sur le même dossier que la page html.

- Travail demandé

Créer une page html avec comme titre « exercice1 » avec un icône de votre choix au format png 70x70.

1.2. Les commentaires

Un commentaire est une balise HTML avec une forme bien spéciale :

2. 1 <!-- Ceci est un commentaire -->

Vous pouvez le mettre où vous voulez au sein de votre code source : il n'a aucun impact sur votre page.

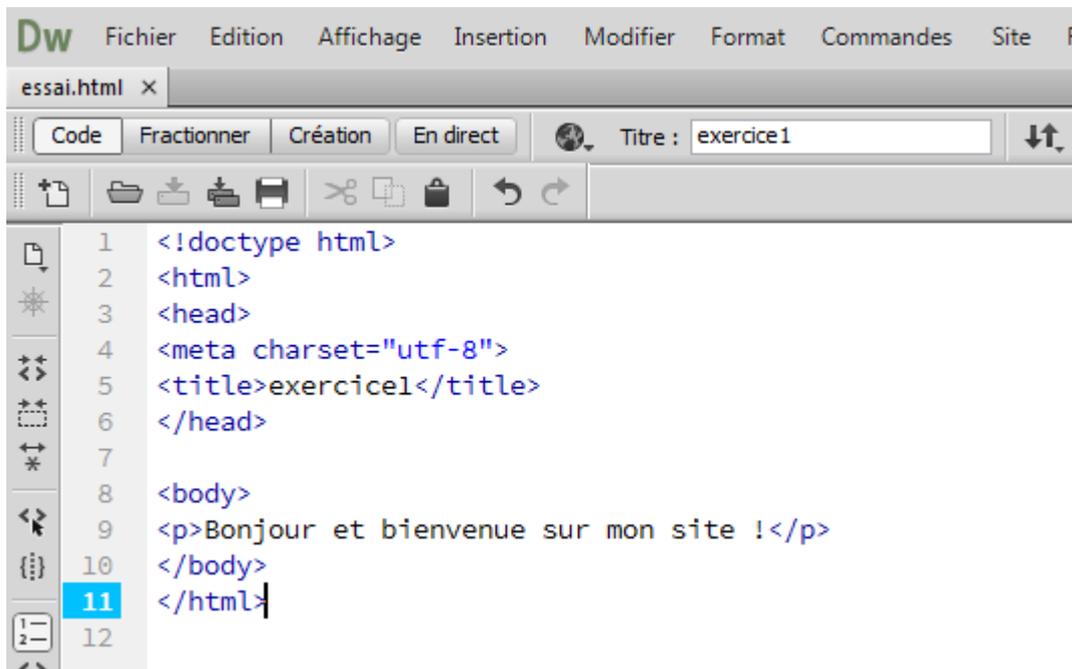
2.1. Les paragraphes

La plupart du temps, lorsqu'on écrit du texte dans une page web, on le fait à l'intérieur de paragraphes. Le langage HTML propose justement la balise <p> pour délimiter les paragraphes.

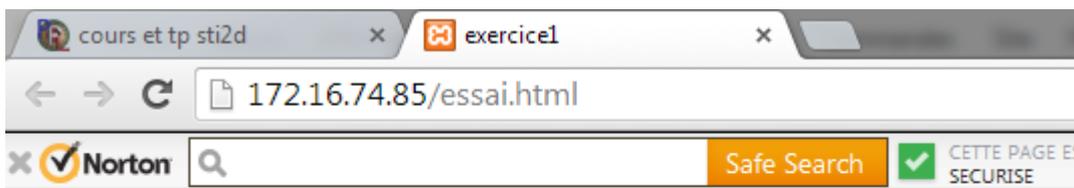
```
1<p>Bonjour et bienvenue sur mon site !</p>
```

- <p> signifie « Début du paragraphe » ;
- </p> signifie « Fin du paragraphe ».

Comme je vous l'ai dit au chapitre précédent, on écrit le contenu du site web entre les balises <body></body>. Il nous suffit donc de mettre notre paragraphe entre ces deux balises et nous aurons enfin notre première vraie page web avec du texte !



```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>exercice1</title>
6 </head>
7
8 <body>
9 <p>Bonjour et bienvenue sur mon site !</p>
10 </body>
11 </html>
12
```



Bonjour et bienvenue sur mon site !

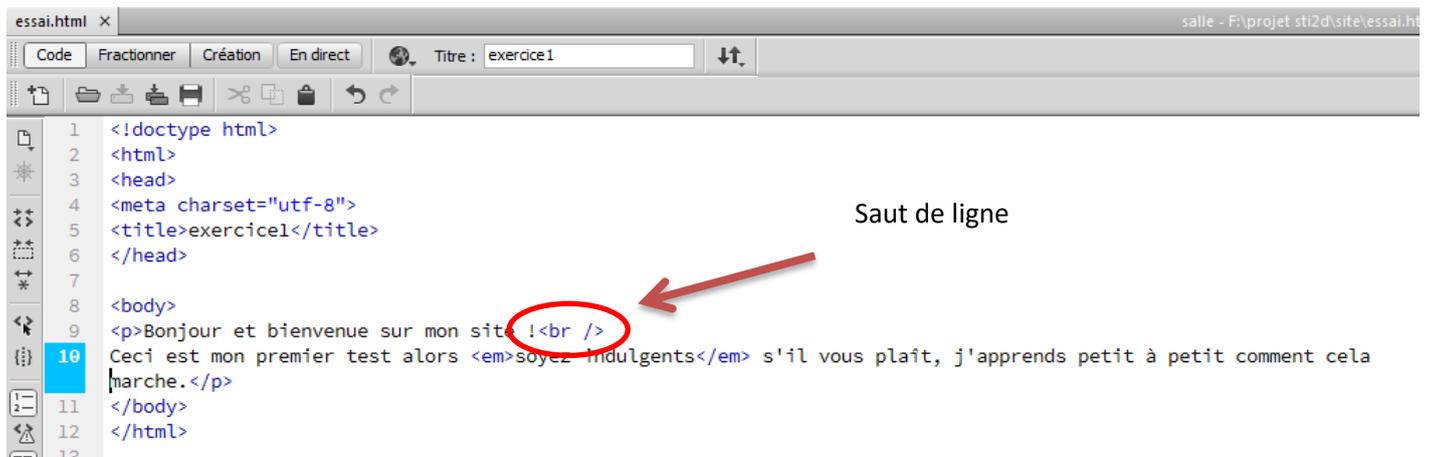
Lorsque le contenu de votre page va s'étoffer avec de nombreux paragraphes, il va devenir difficile pour vos visiteurs de se repérer. C'est là que les titres deviennent utiles.

En HTML, on est verni, on a le droit d'utiliser six niveaux de titres différents. Je veux dire par là qu'on peut dire « Ceci est un titre très important », « Ceci est un titre un peu moins important », « Ceci est un titre encore moins important », etc. On a donc six balises de titres différentes :

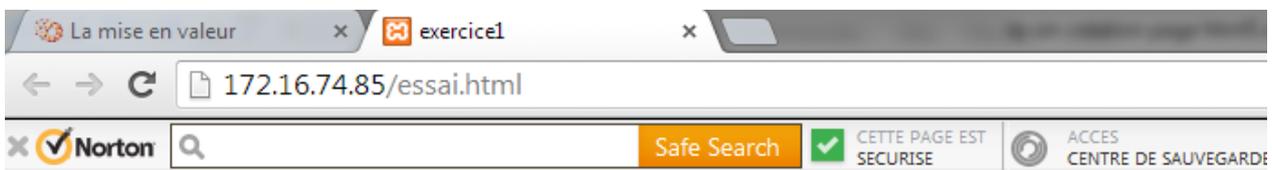
`<h1> </h1>` : signifie « titre important ». En général, on s'en sert pour afficher le titre de la page au début de celle-ci.

`<h2> </h2>` : signifie « titre moins important ».

Pour mettre *un peu* en valeur votre texte, vous pouvez utiliser la balise ` `.



```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>exercice1</title>
6 </head>
7
8 <body>
9 <p>Bonjour et bienvenue sur mon site. <br /> Ceci est mon premier test alors <em>soyez indulgents</em> s'il vous plaît, j'apprends petit à petit comment cela
10 marche.</p>
11 </body>
12 </html>
13
```



Bonjour et bienvenue sur mon site !

Ceci est mon premier test alors *soyez indulgents* s'il vous plaît, j'apprends petit à petit comment cela marche.

On peut aussi utiliser la balise `` qui signifie « fort », ou la balise `<mark>` qui permet de faire ressortir visuellement une portion de texte.

2.2. Les listes

Les listes nous permettent souvent de mieux structurer notre texte et d'ordonner nos informations. Nous allons découvrir ici deux types de listes :

- les listes non ordonnées ou listes à puces ;
- les listes ordonnées ou listes numérotées ou encore énumérations.

2.2.1. Liste non ordonnée

Une liste non ordonnée ressemble à ceci :

- vélo
- voiture
- avion

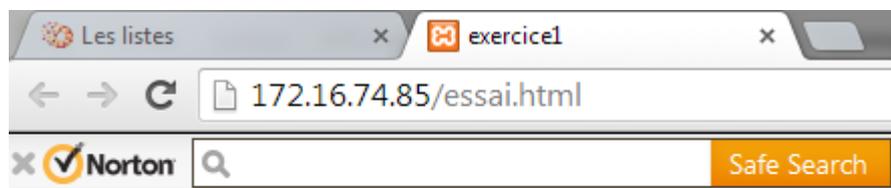
C'est un système qui nous permet de créer une liste d'éléments sans notion d'ordre (il n'y a pas de « premier » ni de « dernier »). Créer une liste non ordonnée est très simple. Il suffit d'utiliser la balise `` que l'on referme un peu plus loin avec ``.

Commencez donc à taper ceci :

```
1<ul></ul>
```

Et maintenant, voilà ce qu'on va faire : on va écrire chacun des éléments de la liste entre deux balises ``. Chacune de ces balises doit se trouver entre `` et ``. Vous allez comprendre de suite avec cet exemple :

```
1<ul>
2  <li>vélo</li>
3  <li>voiture</li>
4  <li>avion</li>
5</ul>
```



- vélo
- voiture
- avion

- `` délimite toute la liste ;
- `` délimite un élément de la liste (une puce).

Pour ceux qui ont besoin de faire des listes complexes, sachez que vous pouvez *imbriquer* des listes à puces (créer une liste à puces **dans** une liste à puces). Si vous voulez faire ça, ouvrez une seconde balise `` à l'intérieur d'un élément ``.

Si vous fermez les balises dans le bon ordre, vous n'aurez pas de problème. Attention néanmoins, cette technique est un peu compliquée à maîtriser.

Liste ordonnée

Une liste ordonnée fonctionne de la même façon, seule une balise change : il faut remplacer `` par ``.

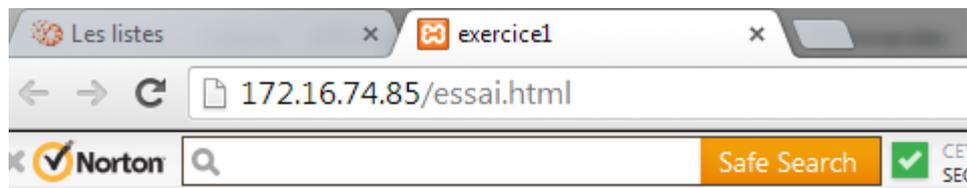
À l'intérieur de la liste, on ne change rien : on utilise toujours des balises `` pour délimiter les éléments.

L'ordre dans lequel vous placez les éléments de la liste est important. Le premier `` sera l'élément n° 1, le second sera le n°2 etc...

Comme c'est particulièrement intuitif, je vous laisse admirer la simplicité de cet exemple (résultat à la figure suivante) :

```
1<h1>Ma journée</h1>
2
3<ol>
```

```
4 <li>Je me lève.</li>
5 <li>Je mange et je bois.</li>
6 <li>Je retourne me coucher.</li>
7</ol>
```

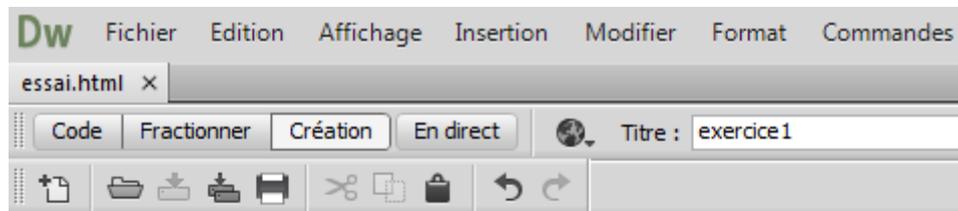


Ma journée

1. Je me lève.
2. Je mange et je bois.
3. Je retourne me coucher.

Pour information, il existe un troisième type de liste, beaucoup plus rare : la liste de définitions. Elle fait intervenir les balises <dl> (pour délimiter la liste), <dt> (pour délimiter un terme) et <dd> (pour délimiter la définition de ce terme).

Travail demandé :



Ecrire le document suivant :

Mon programme de la journée :

1. déjeuner
2. laver
3. partir

Puis j'ai acheté les courses suivantes :

- de la salade
- de la viande
 - poulet
 - veau
- de l'eau

Remarque : surligné balise <u> </u>.

2.3. Création de lien

Comme vous le savez, un site web est composé de plusieurs pages. Pour aller d'une page à une autre, il faut créer un lien.

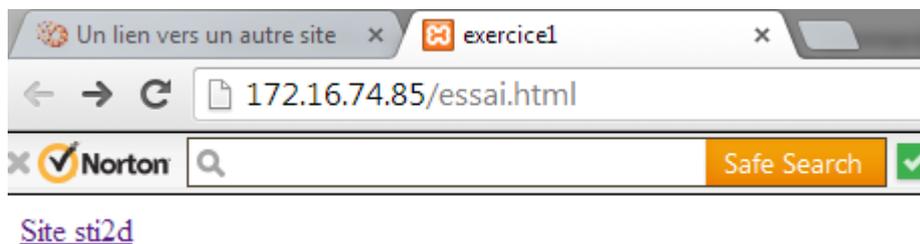
On peut faire un lien d'une page a.html vers une page b.html, mais on peut aussi faire un lien vers un autre site (par exemple, <http://www.siteduzero.com>). Dans les deux cas le fonctionnement est le même.

2.3.1. Un lien vers un autre site

Pour faire un lien, on utilise la balise : `<a>`. Il faut cependant lui ajouter un attribut, href, pour indiquer vers quelle page le lien doit conduire.

Par exemple, le code ci-dessous est un lien qui amène vers le Site coursstimartinique, situé à l'adresse <http://www.coursstimartinique.fr> :

```
1<a href="http://www.coursstimartinique.fr">Site sti2d</a>
```



2.3.2. Un lien vers une ancre

D'abord il faut créer une ancre. Pour cela, il suffit de rajouter l'attribut id à une balise qui va alors servir de repère. Ce peut être n'importe quelle balise.

```
1<h2 id="mon_ancre">Titre</h2>
```

Ensuite, il suffit de créer un lien comme d'habitude, mais cette fois l'attribut href contiendra un dièse (#) suivi du nom de l'ancre. Exemple :

```
1<a href="#mon_ancre">Aller vers l'ancre</a>
```

Normalement, si vous cliquez sur le lien, cela vous amènera plus bas dans la même page (à condition que la page comporte suffisamment de texte pour que les barres de défilement se déplacent automatiquement). Voici un exemple de page comportant beaucoup de texte et utilisant les ancres:

```
<ul>
  <p><a href="#ancre">Aller vers l'ancre</a>
</p>
  <p>&nbsp;</p>
</ul>
<p> Laisser de l'espace pour que Partie 2 n'apparaisse pas sur la page html</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<h2 id="ancre">Partie 2</h2>
<p>Ancre</p>
<p>&nbsp;</p>
```

5. Créer les liens

1. Créer un lien vers une nouvelle page (lien.html avec du texte lien réussi)
 - [Lien](#)
2. Créer un lien vers un paragraphe (partie 2)

[Aller vers l'ancre](#)

Laisser de l'espace pour que Partie 2 n'apparaisse pas sur la page html

Partie 2

[Ancre](#)

2.3.3. [Lien vers une messagerie](#)

```
<h1><a href="mailto:toto@orange.fr">Insertion d'une image</a></h1>
```

On
rajoute
mailto
de
l'adresse

5. Créer les liens

1. Créer un lien vers une nouvelle page (lien.html avec du texte lien réussi)
 - [Lien](#)
2. Créer un lien vers un paragraphe (partie 2)

[Aller vers l'ancre](#)

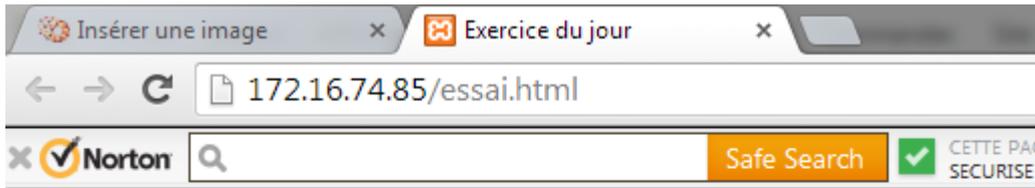
Laisser de l'espace pour que Partie 2 n'apparaisse pas sur la page html

Partie 2

[Ancre](#)

suivi

2.4. Insertion d'une image



Insertion d'une image



```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Exercice du jour</title>
</head>

<body>
<h1>Insertion d'une image</h1>
<p></p>
</body>
</html>
```

- src : il permet d'indiquer où se trouve l'image que l'on veut insérer. Vous pouvez soit mettre un chemin absolu (ex. : `http://www.coursstimartinique.fr/img10.jpg`), soit mettre le chemin en relatif (ce qu'on fait le plus souvent). Ainsi, si votre image est dans un sous-dossier images, vous devrez taper `src="images/img10.jpg "`. dans le cas où l'image est dans un dossier supérieure on mettra `../`
- alt : cela signifie « texte alternatif ». On doit *toujours* indiquer un texte alternatif à l'image, c'est-à-dire un court texte qui décrit ce que contient l'image. Ce texte sera affiché à la place de l'image si celle-ci ne peut pas être téléchargée (cela arrive), ou dans les navigateurs de personnes handicapées (non-voyants) qui ne peuvent malheureusement pas « voir » l'image. Cela aide aussi les robots des moteurs de recherche pour les recherches d'images. Pour le bouton, on mettrait par exemple : `alt="photo d'un bouton"`.
- Width : indique la largeur de l'image
- Height : indique la hauteur.

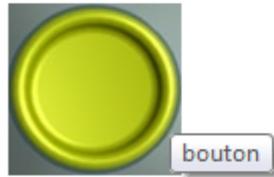
2.4.1. Image avec lien

```
<p><a href="lien.html"></a></p>
```

Pour ajouter un infobulle on rajoute l'attribut title permettant d'afficher une bulle d'aide. Cet attribut est facultatif (contrairement à alt).

```
<p><a href="lien.html"></a></p>
```

Insertion d'une image



3. Le langage CSS

Les **CSS**, *Cascading Style Sheets* (feuilles de styles en cascade), servent à mettre en forme des documents web, type page HTML ou XML, par l'intermédiaire de propriétés d'apparence (couleurs, bordures, polices, etc.) et de placement (largeur, hauteur, côte à côte, dessus-dessous, etc.). Le rendu d'une page web peut être intégralement modifié sans aucun code supplémentaire dans la page web.

On peut écrire du code en langage CSS à trois endroits différents :

- dans un fichier .css (*méthode la plus recommandée*) ;
- dans l'en-tête <head> du fichier HTML ;
- directement dans les balises du fichier HTML *via* un attribut style (*méthode la moins recommandée*). (Ex : <div align="center">

3.1. Dans un fichier .css

On écrit le code CSS dans un fichier spécial ayant l'extension .css (contrairement aux fichiers HTML qui ont l'extension .html). C'est la méthode la plus pratique et la plus souple. Cela nous évite de tout mélanger dans un même fichier.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="style.css" />
    <title>Exercice fichierCSS</title>
  </head>

  <body>
    <h1>Exercice css</h1>
```

C'est le contenu, <link rel="stylesheet" href="style.css" /> : qui indique que le fichier HTML est associé à un fichier appelé style.css qui est chargé de la mise en forme.

Exemple : on veut que le texte du document html soit de couleur rouge

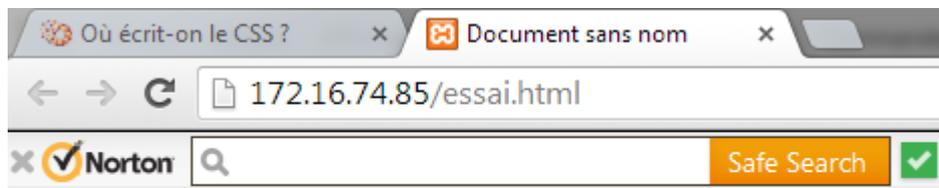
Dans le fichier html on mettra :

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<link rel="stylesheet" href="fichiercss.css" />
<title>Document sans nom</title>
</head>
<body >
  <p>couleur rouge</p>
</body>
</html>
```

Dans le fichier css on mettra :

```
@charset "utf-8";
/* CSS Document */
p
{
color: red;
}
```

On obtient :



couleur rouge

3.2. Dans les balises

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Document sans nom</title>
</head>
<p style="color: red;">couleur rouge</p>
</body>
</html>
```

Vous pouvez ajouter un attribut style à la balise. Vous insérerez votre code CSS directement dans l'attribut.

4. Description du langage CSS

Dans un code CSS comme celui-ci, on trouve trois éléments différents :

- **Des noms de balises** : on écrit les noms des balises dont on veut modifier l'apparence. Par exemple, si je veux modifier l'apparence de tous les paragraphes <p>, je dois écrire p.
- **Des propriétés CSS** : les « effets de style » de la page sont rangés dans des propriétés. Il y a par exemple la propriété color qui permet d'indiquer la couleur du texte, font-size qui permet d'indiquer la taille du texte, etc. Il y a beaucoup de propriétés CSS et, comme je vous l'ai dit, je ne vous obligerai pas à les connaître toutes par cœur.
- **Les valeurs** : pour chaque propriété CSS, on doit indiquer une valeur. Par exemple, pour la propriété color, il faut indiquer le nom de la couleur. Pour font-size, il faut indiquer quelle taille on veut, etc.

Exemple : on veut écrire le paragraphe 1 en bleu et le paragraphe 2 en rouge

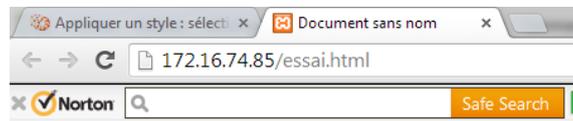
Sur la page html, on écrit :

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<link rel="stylesheet" href="fichiercss.css" />
<title>Document sans nom</title>
</head>
<body>
<h1>Couleur bleu</h1>
<h2>couleur rouge</h2>
</body>
</html>
```

Sur la page CSS on écrit :

```
@charset "utf-8";
/* CSS Document */
h1
{
color: blue;
}
h2
{
color: red;
}
```

On obtient :



Couleur bleu

couleur rouge

Dans le cas où plusieurs balises ont la même propriété. Il suffit de séparer les balises par une virgule.

```
@charset "utf-8";  
/* CSS Document */  
h1,h2  
{  
color: blue;  
}
```

4.1. Sélection d'une balise

Tout ce que l'on a fait jusqu'à maintenant implique que tous les paragraphes possèdent la même présentation.

Pour résoudre le problème, on peut utiliser ces attributs spéciaux *qui fonctionnent sur toutes les balises* :

- l'attribut class ;
- l'attribut id.

4.1.1. L'attribut class :

On doit écrire un nom qui sert à identifier la balise. Ce que vous voulez, du moment que le nom commence par une lettre.

Sur la page html, on écrit :

```
<!doctype html>  
<html>  
<head>  
<meta charset="utf-8">  
<link rel="stylesheet" href="fichiercss.css" />  
<title>Document sans nom</title>  
</head>  
<body>  
<h1 class="bleu"> Couleur bleu</h1>  
<h2 class="rouge">couleur rouge</h2>  
</body>  
</html>
```

Sur la page CSS on écrit :

```
@charset "utf-8";
/* CSS Document */
.bleu
{
color: blue;
}
.rouge
{
color: red;
}
```

4.1.2. L'attribut ID :

Il fonctionne exactement de la même manière que class, à un détail près : il ne peut être utilisé *qu'une seule fois* dans le code.

Si vous utilisez des id, lorsque vous définirez leurs propriétés dans le fichier CSS, il faudra faire précéder le nom de l'id par un dièse (#) :

```
#rouge
{
color: red;
}
```

4.1.3. Les balises universelles

Il arrivera parfois que vous ayez besoin d'appliquer une class (ou un id) à certains mots qui, à l'origine, ne sont pas entourés par des balises.

En effet, le problème de class, c'est qu'il s'agit d'un attribut. Vous ne pouvez donc en mettre que sur une balise. Si, par exemple, je veux modifier uniquement « bienvenue » dans le paragraphe suivant :

```
1<p>Bonjour et bienvenue sur le site acajou2 !</p>
```

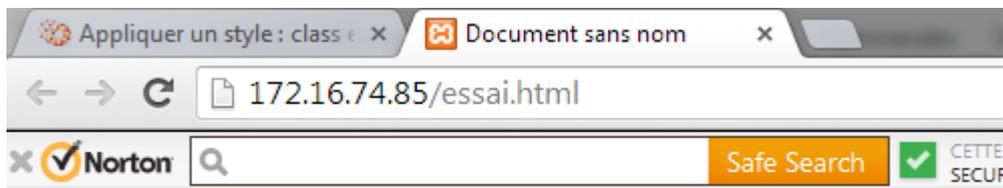
Cela serait facile à faire s'il y avait une balise autour de « bienvenue » mais, malheureusement il n'y en a pas. En fait, on a inventé deux balises dites **universelles**, qui n'ont aucune signification particulière (elles n'indiquent pas que le mot est important, par exemple). Il y a une différence minime (mais significative !) entre ces deux balises :

- ` ` : c'est une balise de type **inline**, c'est-à-dire une balise que l'on place au sein d'un paragraphe de texte, pour sélectionner certains mots uniquement. Les balises `` et `` sont de la même famille. Cette balise s'utilise donc au milieu d'un paragraphe et c'est celle dont nous allons nous servir pour colorer « bienvenue ».
- `<div> </div>` : c'est une balise de type **block**, qui entoure un bloc de texte. Les balises `<p>`, `<h1>`, etc. sont de la même famille. Ces balises ont quelque chose en commun : elles créent un nouveau « bloc » dans la page et provoquent donc obligatoirement un retour à la ligne. `<div>` est une balise fréquemment utilisée dans la construction d'un design (tableau).

Pour le moment donc, nous allons utiliser plutôt la balise . On la met autour de « bienvenue », on lui ajoute une classe (du nom qu'on veut), on crée le CSS et c'est gagné !

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<link rel="stylesheet" href="fichiercss.css" />
<title>Document sans nom</title>
</head>
<body>
<p>Bonjour et <span class="bleu">bienvenue</span> sur le site acajou2</p>
</body>
</html>
```

```
@charset "utf-8";
/* CSS Document */
.bleu
{
color: blue;
}
.rouge
{
color: red;
}
```



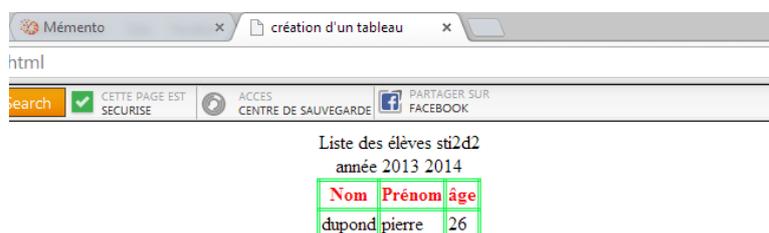
Bonjour et bienvenue sur le site acajou2

4.1.4. Propriétés CSS

<http://fr.openclassrooms.com/informatique/cours/apprenez-a-creer-votre-site-web-avec-html5-et-css3/memento-1-1>

5. Les tableaux

On veut obtenir le tableau suivant :



Page html :

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>création d'un tableau</title>
<link href="feuilleessai.css" rel="stylesheet" type="text/css">
</head>

<body>
<div align="center">
  <table >
    <caption>Liste des élèves sti2d2 année 2013 2014</caption>
    <tr>
      <th >Nom</th>
      <th >Prénom</th>
      <th >âge</th>
    </tr>
    <tr>
      <td>dupond</td>
      <td>pierre</td>
      <td>26</td>
    </tr>
  </table>
</div>
</body>
</html>
```

Titre du tableau

Normalement, tout tableau doit avoir un titre. Le titre permet de renseigner rapidement le visiteur sur le contenu du tableau.

La balise `<caption>` ! se place tout au début du tableau, juste avant l'en-tête. C'est elle qui contient le titre du tableau .

Sachez que vous pouvez changer la position du titre avec la propriété CSS **caption-side** qui peut prendre deux valeurs :

- top : le titre sera placé au-dessus du tableau (par défaut) ;
- bottom : le titre sera placé en dessous du tableau.

Structure du tableau

La première balise à connaître est `<table> </table>`. C'est cette balise qui permet d'indiquer le début et la fin d'un tableau.

Cette balise est de type bloc, il faut donc la placer en dehors d'un paragraphe.

Puis on trouve les balises suivantes :

- `<tr> </tr>` : indique le début et la fin d'une ligne du tableau ;
- `<th> </th>` : indique le début et la fin du contenu du titre des colonnes.
- `<td> </td>` : indique le début et la fin du contenu d'une cellule.

Le nombre de th doit être égal au nombre de td.

En HTML, un tableau se construit ligne par ligne. Dans chaque ligne (`<tr>`), on indique le contenu des différentes cellules (`<td>`).

Schématiquement, un tableau se construit comme à la figure suivante.

**Td
(cellules)**

Nom	Age	Pays
Anne	27 ans	France
Carmen	33 ans	Espagne
Michelle	26 ans	Etats-Unis
Ogasaku Nyagatosoka	18 ans	Japon

**Tr
(ligne)**

Page CSS :

```
@charset "utf-8";
th
{
    font-weight:bold;
    text-align:center;
    color:#F7080C;
    border-style:double;
    border-color:#08F841;
}
td
{
    color:#060000;
    border-style:double;
    border-color:#08F841;
}
table
{
    border-collapse:collapse;
    border-width:medium;
    border-color:#080DF7;
    border-style:double;
    align-self:center;
}
```

La propriété CSS **border-collapse** signifie « coller les bordures entre elles ».

Cette propriété peut prendre deux valeurs :

- collapse : les bordures seront collées entre elles, c'est l'effet qu'on recherche ici ;
- separate : les bordures seront dissociées (valeur par défaut)

6. Les variables CSS4

Variables CSS sont des entités définies par l'auteur du fichier CSS et qui contiennent des valeurs spécifiques pour être réutilisés dans un document. Pour définir la variable, on doit utiliser la règle suivante :

Exemple :

```
--main-color : black ;
```

Elles sont accessibles à l'aide de la fonction var (). Par exemple : background-color : var(--main-color) ;

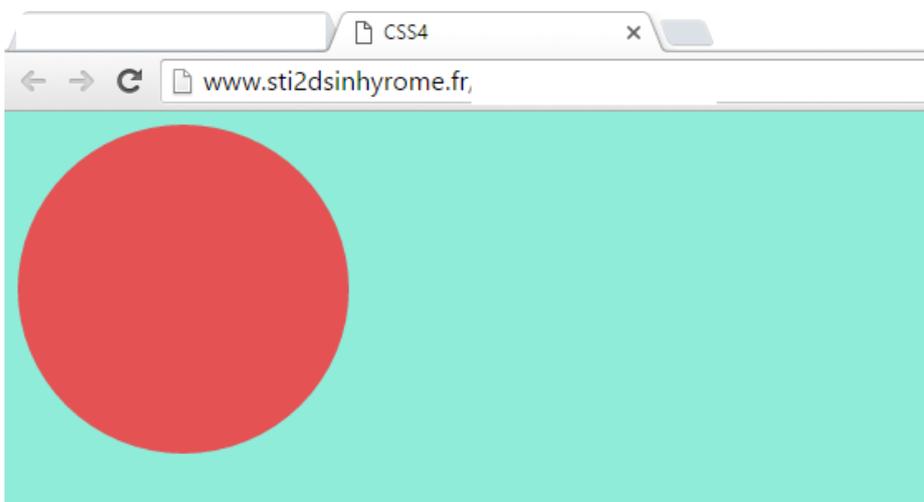
Fichier Html

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>CSS4</title>
6 <link href="ex1.css" rel="stylesheet" type="text/css">
7 </head>
8
9 <body>
10 </body>
11 </html>
12
```

Fichier CSS

```
1 @charset "utf-8";
2 /* CSS Document */
3 body
4 {
5     --body-color:#8FEC99;
6     background-color: var(--body-color);
7 }
8
```

Navigateur



Remarque :

La pseudo-classe : « root » permet de cibler la racine de l'arbre représentant le document. Pour un document HTML, :root ciblera donc l'élément <html> et aura le même comportement que le sélecteur <html> mais sa spécificité sera plus forte.

6.1. Propriété des variables

Héritages des variables CSS

Les propriétés personnalisées héritent. Cela signifie que si aucune valeur est définie pour une propriété personnalisée sur un élément donné, la valeur de son parent est utilisée.

Exemple :

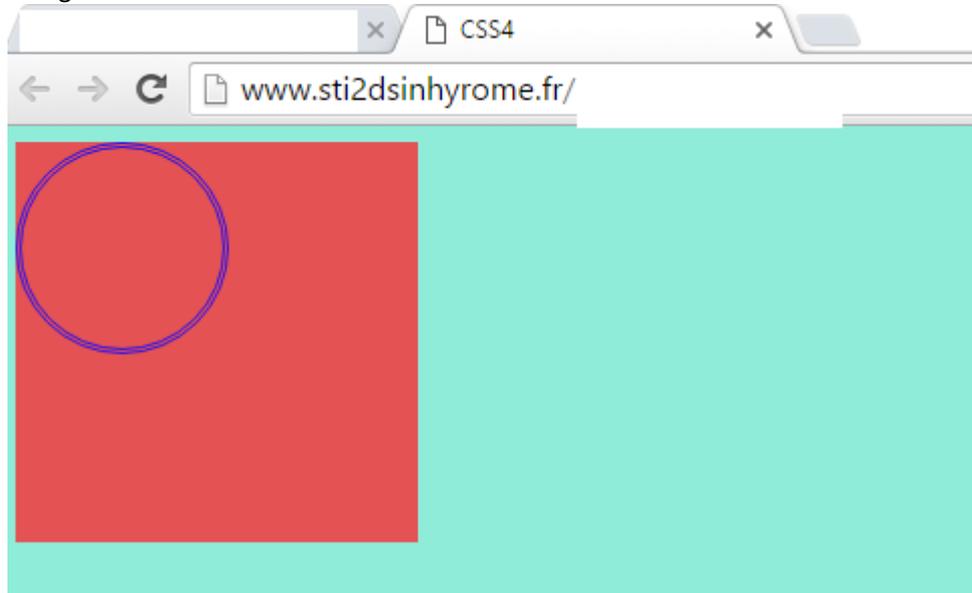
Html

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>CSS4</title>
6 <link href="ex2.css" rel="stylesheet" type="text/css">
7 </head>
8
9 <body>
10 <div id="div1">
11 <div id="div2">
12 </div>
13 </div>
14 </body>
15 </html>
16 |
```

CSS

```
1 @charset "utf-8";
2 /* CSS Document */
3 :root
4 {
5     --body-color:#8FECD9;
6     --div-color:#E45254;
7 }
8 body
9 {
10     --body-color:#8FECD9;
11     background-color: var(--body-color);
12 }
13 #div1
14 {
15     background-color: var(--div-color);
16     width: 200px;
17     height: 200px;
18 }
19 #div2
20 {
21     width: 100px;
22     height: 100px;
23     border-radius: 50%;
24     border: medium double #2710EF;
25 }
26
```

Navigateur



7. Animation CSS

Une animation permet de faire changer une propriété CSS

Vous pouvez changer autant de propriétés CSS que vous voulez, autant de fois que vous voulez.

Pour utiliser l'animation CSS3, vous devez d'abord spécifier des images clés pour l'animation.

Lorsque vous spécifiez des styles CSS dans « @keyframes », l'animation va progressivement changer du style actuel au nouveau style à certains moments.

Pour obtenir l'animation, il faut la relier à un élément

L'exemple suivant lie l'animation exemple à l'élément <div id="div1">. L'animation dure 4 secondes, et il va progressivement changer de dimension.

Html

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>CSS4</title>
6 <link href="ex3.css" rel="stylesheet" type="text/css">
7
8 </style>
9 </head>
10
11 <body>
12 <div id="div1">
13 <div id="div2">
14 </div>
15 </div>
16 </body>
17 </html>
```

CSS

```
1 @charset "utf-8";
2 /* CSS Document */
3 :root
4 {
5     --body-color:#8FECDD;
6     --div1-color:#E45254;
7     --div2-color:#0EEF5F;
8 }
9 body
10 {
11     --body-color:#8FECDD;
12     background-color: var(--body-color);
13 }
14 #div1
15 {
16     background-color: var(--div1-color);
17     width: 200px;
18     height: 200px;
19     animation-name:exemple;
20     animation-duration:4s;
21 }
22 #div2
23 {
24     background-color: var(--div2-color);
25     width: 100px;
26     height: 100px;
27     border-radius: 50%;
28     border: medium double #2710EF;
29 }
30 @keyframes exemple
31 {
32     0%{transform:scaleY(0);}
33     50%{transform:scaleY(0.5)}
34     100%{transform:scaleY(1);}
35 }
```

Voir animation :

<http://www.sti2dsinhyrome.fr/css4/ex3/ex3.html>

7.1. Différentes propriétés de règle d'animation

animation-delay

Cette propriété définit le délai entre le moment où l'élément est chargé et le moment où l'animation commence.

animation-direction

Cette propriété indique si l'animation doit alterner entre deux directions de progressions (faire des allers-retours) ou recommencer au début à chaque cycle de répétition.

animation-duration

Cette propriété définit la durée d'un cycle de l'animation.

animation-iteration-count

Cette propriété détermine le nombre de fois que l'animation est répétée. On peut utiliser le mot-clé `infinite` afin de répéter une animation infiniment.

animation-name

Cette propriété permet de déclarer un nom qui pourra être utilisé comme référence à l'animation pour la règle `@keyframes`.

animation-play-state

Cette propriété permet d'interrompre (« pause ») ou de reprendre l'exécution d'une animation.

animation-timing-function

Cette propriété configure la fonction de minutage d'une animation, autrement dit comment celle-ci accélère entre l'état initial et l'état final notamment grâce à des fonctions décrivant des courbes d'accélération.

animation-fill-mode

Cette propriété indique les valeurs qui doivent être appliquées aux propriétés avant et après l'exécution de l'animation.

http://www.w3schools.com/css/css3_animations.asp