

REPUBLIQUE DEMOCRATIQUE DU CONGO
INSTITUT UNIVERSITAIRE D'ETUDES ET DE
FORMATION EN DEVELOPPEMENT

MBANZA-NGUNGU

BALISAGE EN HTML 5 ET CSS 3

Chapitre 1. HyperText MarkupLanguage

HTML sigle anglais qui signifie **HyperText Markup Language** : il a fait son apparition dès 1991 lors du lancement du Web. Il a pour rôle de gérer et d'organiser le contenu d'une page web. Le contenu d'une page web contient des textes, liens, images, tableau...etc

Les versions de HTML

HTML 1 : c'est la toute première version créée par Tim Berners-Lee en 1991.

HTML 2 : la deuxième version du HTML apparaît en 1994 et prend fin en 1996 avec l'apparition du HTML 3.0. C'est cette version qui posera en fait les bases des versions suivantes du HTML. Les règles et le fonctionnement de cette version sont donnés par le W3C (tandis que la première version a été créée par un seul homme).

HTML 3 : apparue en 1996, cette nouvelle version du HTML rajoute de nombreuses possibilités au langage comme les tableaux, les applets, les scripts, le positionnement du texte autour des images, etc.

HTML 4 : il s'agit de la version la plus répandue du HTML (plus précisément, il s'agit de HTML 4.01). Elle apparaît pour la première fois en 1998 et propose l'utilisation de frames (qui découpent une page web en plusieurs parties), des tableaux plus complexes, des améliorations sur les formulaires, etc. Mais surtout, cette version permet pour la première fois d'exploiter des feuilles de style, notre fameux CSS !

HTML 5 : c'est la dernière version. Encore assez peu répandue, elle fait beaucoup parler d'elle car elle apporte de nombreuses améliorations comme la possibilité d'inclure facilement des vidéos, un meilleur agencement du contenu, de nouvelles fonctionnalités pour les formulaires, etc. C'est cette version que nous allons découvrir ensemble.

Quand nous utilisons le HTML, les informations s'affichent brute cela veut dire sans mise en forme. Et pour le mettre en forme, là on fait recours à un autre langage que nous verrons dans le chapitre 2 « le CSS ».

Le HTML est à la base de tout projet de développement web. Il se concentre sur le fond d'une page web.

Pour afficher nos pages web, nous aurons besoin des navigateurs web.



Internet Explorer



Mozilla Firefox



Google Chrome



Safari

XHTML = Extensible HTML, c'est la version la plus stricte codifiée d'HTML. Créé en 2000 pour succéder au HTML puis abandonné en 2009 ou plus exactement intégré dans le HTML 5.

Choix d'un éditeur

Pour coder en HTML, nous avons besoin d'un éditeur de texte.

En réalité, il existe plusieurs éditeurs pour écrire du texte en HTML.

Parmi ces éditeurs, nous pouvons citer :

- Komodo
- Notepad++
- gEdit
- textWrangler

Editeur orienté Système d'exploitation

Windows	: Komodo, Notepad++
Mac	: Komodo, TextWrangler
Linux	: Komodo, gEdit

Nous allons utiliser l'éditeur Notepad++

Il ne suffit pas "simplement" d'écrire du texte dans l'éditeur, il faut aussi donner des instructions à l'ordinateur. En HTML, on passe pour cela par des balises.

Les balises

Les pages HTML sont remplies de ce qu'on appelle des balises, l'élément fait parti intégrante d'une balise encadrée des symboles < et > appelés chevron.

L'élément P définit un paragraphe ;
<p> définit une balise ;

Nous distinguons deux types de balise :

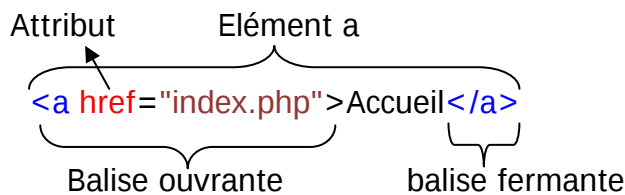
- Balise en paire ;
- Balise orpheline.

Balise en paire est constituée d'une balise ouvrante et d'une balise fermante par exemple : <p></p>

Balise orpheline est constituée que d'un seul élément par exemple

Les attributs

Sont des propriétés utilisées pour donner des indications supplémentaires aux éléments. L'attribut se place toujours dans la balise ouvrante d'un élément et a le plus souvent une valeur.



Structure d'une page en HTML5

Vous noterez que les balises s'ouvrent et se ferment dans un ordre précis.

```
<!DOCTYPE html>
<html>
  <head>
    <title></title>
    <meta charset="utf-8"/>
  </head>
  <body>
  </body>
</html>
```

Explication sur cette structure :

< !DOCTYPE html> : cette balise est indispensable car elle nous permet d'indiquer que nous sommes bel et bien dans une page web html.

<html></html> : balise en paire, c'est la balise principale du code HTML. En elle, elle renferme le contenu de votre page web.

<head></head> (en-tête) : balise en paire, donne quelques informations générales sur la page, comme son titre, l'encodage (pour la gestion des caractères spéciaux), etc. Les informations que l'en-tête contient ne sont pas affichées sur la page, ce sont simplement des informations générales à destination de l'ordinateur.

Elles sont cependant très importantes !

<title></title> : balise en paire, 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 que le titre soit assez court (moins de 100 caractères en général).Le titre ne s'affiche pas dans votre page mais en haut de celle-ci (souvent dans l'onglet du navigateur).

<meta charset="utf-8"/> : balise orpheline, Cette méthode d'encodage « charset » permet d'afficher sans aucun problème pratiquement tous les symboles de toutes les langues de notre planète ! C'est pour cela que j'ai indiqué utf-8 dans cette balise.

Nota : en *Notepad++*, avant de codifier il faut configurer la page en utf-8 sans BOM. C'est-à-dire, menu encodage puis cliquer sur encoder en utf-8 sans BOM.

<body></body> : balise en paire, 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.

Insertion d'un commentaire en HTML

Nous savons qu'un commentaire permet de se situer en programmation, il faut noter qu'un commentaire n'est pas lu par ordinateur.

En HTML, le commentaire est représenté sous : `<!-- ceci est mon commentaire -->`

Les commentaires ne seront pas visible à l'affichage de votre page web sur le navigateur mais uniquement quand vous regarderez les codes de votre page.

Organisation du texte dans une page HTML

L'organisation du texte en HTML demande quelques éléments importants que nous allons voir tout au long de notre cours.

L'élément **p** est une balise en paire qui permet d'écrire les textes en paragraphe.

L'élément **br** est une balise orpheline qui permet d'aller à la ligne suivante. Et il est utilisé à l'intérieur de l'élément **p**.

L'élément **h1**, **h2**, **h3**, **h4**, **h5**, **h6** sont des balises en paire qui permet de dimensionner la taille du texte.

L'élément **strong** est une balise en paire qui indique au navigateur que le contenu est particulièrement important en le mettant en gras.

L'élément **em** est une balise en paire qui indique au navigateur que le contenu est particulièrement important en le mettant en italique.

L'élément **mark** est une balise en paire qui permet de faire ressortir visuellement une portion de texte. Le texte n'est pas forcément considéré comme important mais on veut qu'il se distingue bien du reste du texte.

Listes en HTML

Comme en Microsoft Word nous avons les listes à puces et à numéros, en HTML nous distinguons trois sortes de listes à savoir :

- Liste ordonnée
- Liste non ordonnée
- Liste de définition

Liste ordonnée :

C'est la numérotation automatique de texte. Nous allons utiliser la balise en paire **ol** et à l'intérieur de cette balise nous mettrons une autre balise en paire **li**.

Exemple :

HTML	Résultat
<pre> Amour Commandement Bonnes œuvres </pre>	1. Amour 2. Commandement 3. Bonnes œuvres

Liste non ordonnée :

C'est la mise en puce de texte. Nous allons utiliser la balise en paire **ul** et à l'intérieur de cette balise nous mettrons une autre balise en paire **li**.

Exemple :

HTML	Résultat
<pre> Kongo central Mbanza-ngungu Gombe Matadi </pre>	• Kongo central • Mbanza-ngungu • Gombe Matadi

Liste de définition :

C'est la numérotation automatique de texte. Nous allons utiliser la balise en paire **dl** et à l'intérieur de cette balise nous mettrons deux autres balises en paire **dt** suivi de **dd**.

Exemple :

HTML	Résultat
<pre><dl> <dt>SBZ</dt> <dd>Sungamana Banzolani Zola</dd> </dl></pre>	SBZ Sungamana Banzolani Zola

Nota : nous pouvons aussi imbriquer les listes.

Imbrication des listes

Permet d'imbriquer les listes les unes après les autres.

Par exemple :

HTML	Résultat
<pre> Université Université de Kinshasa Université de Lubumbashi Institut Supérieur ISS de Kinshasa ISC de Kinshasa IUEFD ISP Gombe ISAM </pre>	<pre>1. Université o Université de Kinshasa o Université de Lubumbashi 2. Institut Supérieur o ISS de Kinshasa o ISC de Kinshasa o IUEFD o ISP Gombe o ISAM</pre>

Création des liens

Il sied de savoir qu'il y a différents types des liens, mais nous allons nous concentrer aux liens internes et liens externes.

L'élément utilisé pour créer un lien c'est **a**, étant une balise en paire.

Un lien interne est un lien créé entre deux pages d'un même site web. Pour créer un lien interne, on utilise un chemin relatif.

Nous avons 3 façons pour créer un lien interne :

- Même dossier : href="palmares.html"
- Sous-dossier : href="sousdossier/palmares.html"
- Dossier parent : href="../palmares.html"

Par exemple :

```
<a href="palmares.html">Palmares</a>
```

Un lien externe est un lien créé entre deux pages de site différent. Pour créer un lien externe, on utilise un chemin absolu. La valeur de l'attribut **href** est l'adresse du site.

```
<a href="http://www.google.cd">Google</a>
```

Nota : pour ouvrir le lien dans une nouvelle fenêtre/onglet, on utilise l'attribut **target** et sa valeur `"_blank"`.

```
<a href="palmares.html" target="_blank">Palmares</a>
```

Affichage d'une info bulle

Info bulle permet d'informer le visiteur avant même qu'il n'ait cliqué sur le lien.

```
<a href="palmares.html" title="Affiche le palmares des matchs ">Palmares</a>
```

Lien de type ancre

Si le contenu d'une page web est accès long, nous utilisons les liens du type ancre. En quelque sorte, les ancres sont des points de repère.

Pour créer une ancre, il suffit de mettre l'attribut **id** à une balise qui va alors servir de repère. Cet attribut permet de donner un nom à l'ancre.

Souvent l'ancre est utilisée à de titre et sous titre d'une page web.

```
<h3 id="nom de l'ancre">Titre de référence</h3>
```

Faire appel avec un lien de l'ancre d'une même page web

```
<a href="#nom de l'ancre">Voir le contenu du titre de référence</a>
```

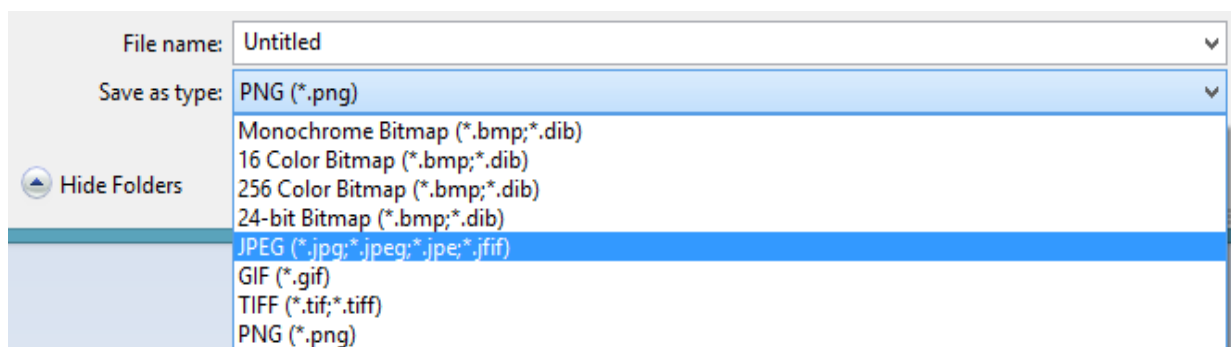
Faire appel avec un lien de l'ancre d'une autre page web

```
<a href="palmares.html#nom de l'ancre">Voir le contenu du titre de référence</a>
```

Insertion d'une image

Nous allons étudier comment placer une image dans une page web. Il existe différents formats d'image que l'on peut utiliser sur des sites web, et on ne doit pas les choisir au hasard.

Voici un exemple de type d'images :



NB : En effet, les images sont parfois volumineuses à télécharger, ce qui ralentit le temps de chargement de la page.

Certains formats sont plus adaptés que d'autres selon l'image (photo, dessin, image animée...). Notre but ici est de faire le tour des différents formats que l'on utilise sur le web, pour que vous les connaissiez et sachiez choisir celui qui convient le mieux à votre image.

Voici les formats que vous devrez utiliser :

Les images au format JPEG (Joint Photographic Expert Group) sont très répandues sur le Web. Ce format est conçu pour réduire la taille des photos, qui peuvent comporter plus de 16 millions de couleurs différentes ayant l'extension .jpg ou .jpeg.

Le format PNG (Portable Network Graphics) est le plus récent de tous. Ce format est adapté à la plupart des graphiques (je serais tenté de dire "à tout ce qui n'est pas une photo"). Le PNG a deux gros avantages : il peut être rendu transparent et il n'altère pas la qualité de l'image.

Le PNG existe en 2 versions, en fonction du nombre de couleurs que doit comporter l'image :

- PNG 8 bits : 256 couleurs
- PNG 24 bits : 16 millions de couleurs (autant qu'une image JPEG)

Le format GIF est un format assez vieux, qui a été néanmoins très utilisé (et qui reste très utilisé par habitude). Aujourd'hui, le PNG est globalement bien meilleur que le GIF : les images sont le plus souvent plus légères et la transparence est de meilleure qualité. Je vous recommande donc d'utiliser le PNG autant que possible.

Le format GIF est limité à 256 couleurs (alors que le PNG peut aller jusqu'à plusieurs millions de couleurs).

Néanmoins, le GIF conserve un certain avantage que le PNG n'a pas : il peut être animé.

Revenons maintenant au code HTML pour découvrir comment placer des images dans nos pages web !

La balise **** est utilisée pour insérer une image. C'est une balise de type orpheline doit être insérer à l'intérieur de l'élément **P**.

La balise doit être accompagnée de 2 attributs obligatoires :

src : il permet d'indiquer où se trouve l'image que l'on veut insérer. Vous pouvez soit mettre un chemin en absolu (ex. :<http://www.sdsoft.org/logo.png>), 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/drapeau.png"**

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 (ça arrive), ou sur 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 la fleur, on mettrait par exemple : **alt="drapeau.png"**.

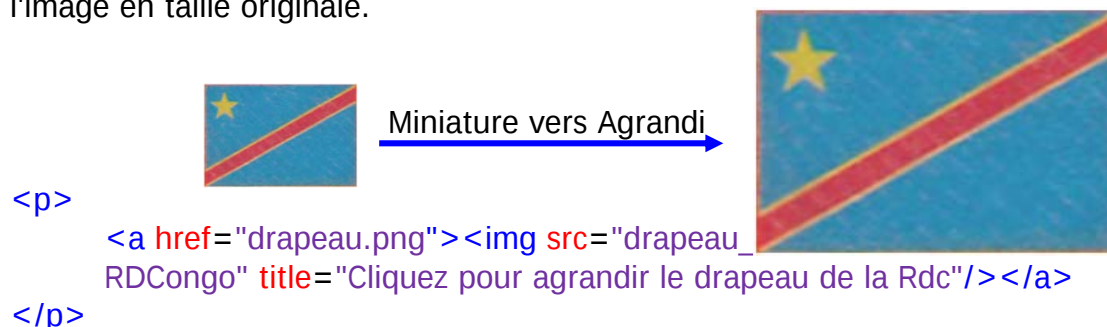
Nous pouvons également ajouter l'attribut **title** pour afficher un texte au survol de la souris.

Voici un exemple :

```
<p>  
    
</p>
```

Image Miniature cliquable

Si votre image est très grosse, il est conseillé d'en afficher la miniature sur votre site. Ajoutez ensuite un lien sur cette miniature pour que vos visiteurs puissent afficher l'image en taille originale.



Un lien pour télécharger un fichier

Beaucoup d'entre vous se demandent comment cela se passe pour le téléchargement d'un fichier. . . En fait, il faut procéder exactement comme si vous faisiez un lien vers une page web, mais en indiquant cette fois le nom du fichier à télécharger.

Par exemple, supposez que vous vouliez faire télécharger monfichier.zip. Placez simplement ce fichier dans le même dossier que votre page web (ou dans un sous-dossier) et faites un lien vers ce fichier :

```
<p><a href="monfichier.zip">Télécharger le fichier </a></p>
```

C'est tout ! Le navigateur, voyant qu'il ne s'agit pas d'une page web à afficher, va lancer la procédure de téléchargement lorsqu'on cliquera sur le lien.

Un lien pour envoyer un e-mail

Si vous voulez que vos visiteurs puissent vous envoyer un e-mail, vous pouvez utiliser des liens de type mailto. Rien ne change au niveau de la balise, vous devez simplement modifier la valeur de l'attribut href comme ceci :

```
<p>  
  <a href="mailto:sungamana@gmail.com">Envoyez - moi un e-mail !</a>  
</p>
```

Il suffit donc de faire commencer le lien par mailto: et d'écrire l'adresse e-mail où on peut vous contacter. Si vous cliquez sur le lien, un nouveau message vide s'ouvre, prêt à être envoyé à votre adresse e-mail.

Création des figures

Ce sont des éléments qui viennent enrichir le texte pour compléter les informations de la page.

Les figures peuvent être de différents types :

- Images
- Codes source
- Citations
- ...

Bref, tout ce qui vient illustrer le texte est une figure. Nous allons ici nous intéresser aux images, mais contrairement à ce qu'on pourrait croire, les figures ne sont pas forcément des images : un code source aussi illustre le texte.

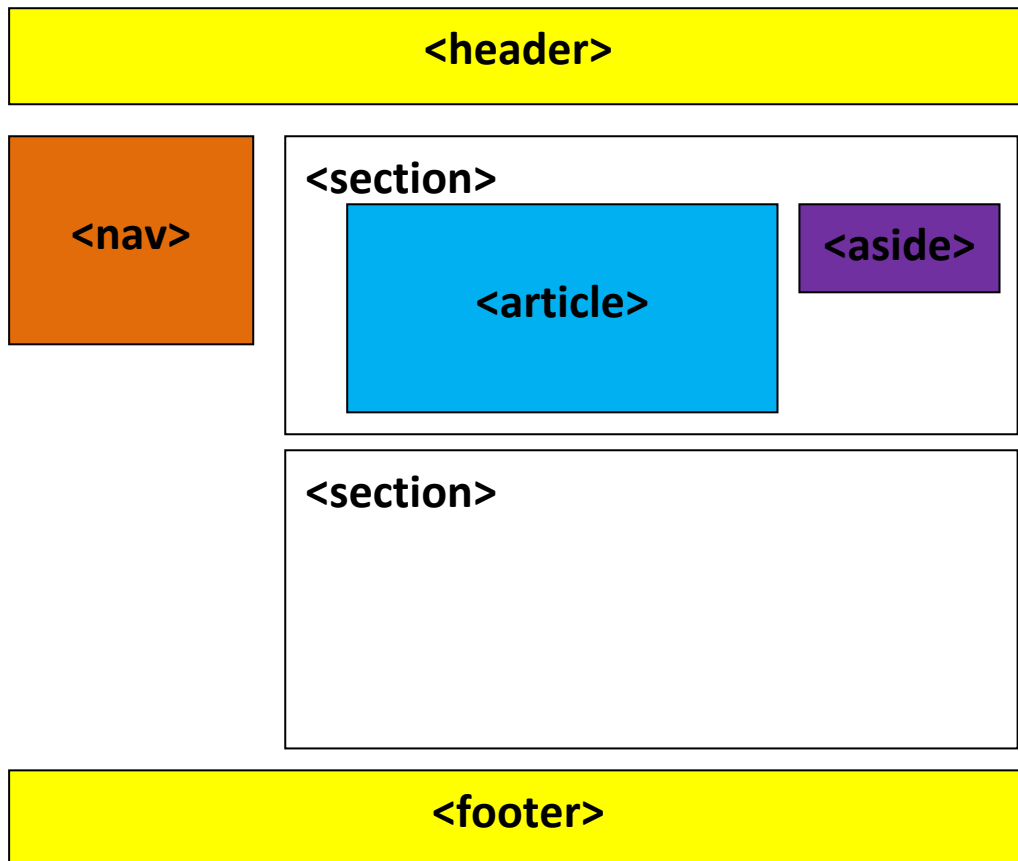
```
<figure>  
    
</figure>
```

Une figure est le plus souvent accompagnée d'une légende. Pour ajouter une légende, utilisez la balise **<figcaption>** à l'intérieur de la balise **<figure>**, comme ceci :

```
<figure>  
    
  <figcaption>Le drapeau de la République Démocratique du  
Congo</figcaption>  
</figure>
```

Balises structurantes en HTML5

Les balises structurantes permettent de structurer les pages web. Voici comment se présente la page web structurée.



Pour l'en tête, nous utilisons la balise **<header>**

```
<header>
    ---Mettre les informations de l'entête ici---
</header>
```

Pour le pied de page, nous utilisons la balise **<footer>**

```
<footer>
    ---Mettre les informations de l'entête ici---
</footer>
```

La balise **<nav>** doit regrouper tous les principaux liens de navigation du site. Vous y placerez par exemple le menu principal de votre site.

Généralement, le menu est réalisé sous forme de liste à puces à l'intérieur de la balise **<nav>** :

```
<nav>
    <ul>
        <li><a href="index.html">Accueil</a></li>
        <li><a href="palmares.html">Palmares</a></li>
        <li><a href="contact.html">Contact</a></li>
    </ul>
</nav>
```

La balise **<section>** sert à regrouper des contenus en fonction de leur thématique. Elle englobe généralement une portion du contenu au centre de la page.

```
<section>
  <h1>Ma section de page</h1>
  <p>Compléter toutes les informations .....</p>
</section>
```

La balise **<aside>** est conçue pour contenir des informations complémentaires au document que l'on visualise. Ces informations sont généralement placées sur le côté (bien que ça ne soit pas une obligation).

```
<aside>
  <!-- Placez ici des informations complémentaires -->
</aside>
```

Il peut y avoir plusieurs blocs **<aside>** dans la page.

La balise **<article>** sert à englober une portion généralement autonome de la page. C'est une portion de la page qui pourrait (par exemple) être reprise ailleurs sur un autre site. C'est le cas par exemple des actualités (articles de journaux ou de blogs).

```
<article>
  <h1>Mon article</h1>
  <p>Compléter toutes les informations .....</p>
</article>
```

Exemple concret de la structure de la page web avec les balises :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>SD Soft–IT Solutions</title>
  </head>
  <body>
    <header>
      <h1>SD Soft</h1>
    </header>
    <nav>
      <ul>
        <li><a href="accueil.html">Accueil</a></li>
        <li><a href="formation.html">Formation</a></li>
        <li><a href="offres.html ">Offres</a></li>
      </ul>
    </nav>
    <section>
      <aside>
        <h1>A propos de SD</h1>
        <p>SD Soft est créé le 11 février 2014 par l'ingénieur
        Concepteur Fabrice Sungamana.</p>
      </aside>
      <article>
        <h1>Développement</h1>
        <p>Nous prenons la charge complète du développement et de la
        maintenance de votre application, vous laissant vous concentrer
        sur les aspects commerciaux et fonctionnels de votre produit.
        </p>
      </article>
    </section>
    <footer>
      <p>Développer par SD Soft --- Copyright réservé à SD Soft 2014
      </p>
    </footer>
  </body>
</html>
```

Balises universelles

Vous les connaissez déjà car je vous les ai présenté il y a quelques chapitres. Ce sont des balises qui n'ont aucun sens particulier (contrairement à **<p>** qui veut dire "paragraphe", **** "important", etc.).

Le principal intérêt de ces balises c'est de pouvoir leur appliquer une class (ou un id) pour le CSS quand aucune autre balise ne convient.

Il existe 2 balises génériques, et comme par hasard la seule différence entre les deux c'est que l'une d'elle est inline, l'autre est block :

`<span></span>` (**inline**)

`<div></div>` (**block**)

Respectez la sémantique !

Les balises universelles sont "pratiques" dans certains cas, certes, mais attention à ne pas en abuser. Je tiens à vous avertir de suite : beaucoup de webmasters mettent des `<div>` et des `<span>` trop souvent, et oublient que d'autres balises plus adaptées existent.

Voici 2 exemples :

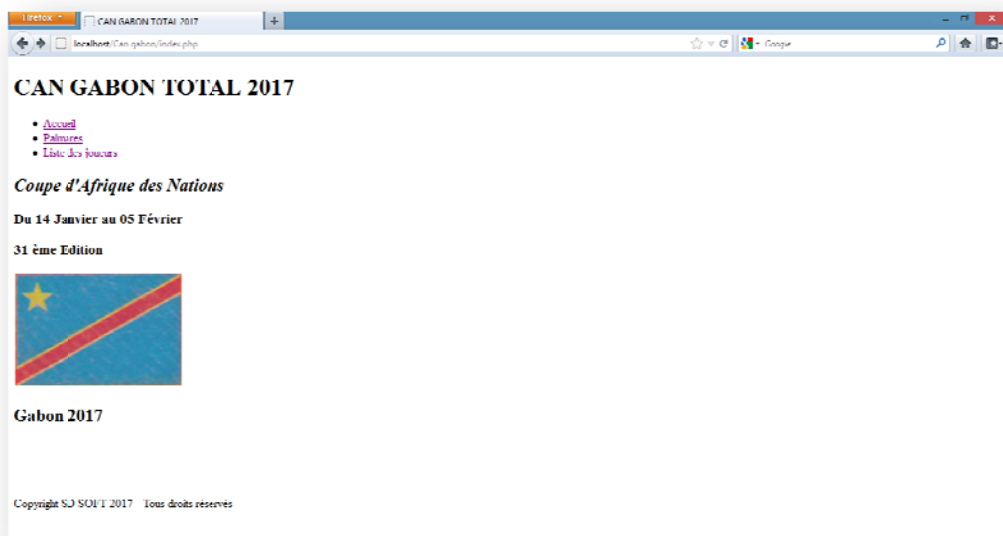
Exemple d'un **span** inutile : `<span class="important">`. Je ne devrais jamais voir ceci dans un de vos codes alors qu'il existe la balise `<strong>` qui sert à indiquer l'importance !

Exemple d'un **div** inutile : `<div class="titre">`. Ceci est complètement absurde puisqu'il existe des balises faites spécialement pour les titres (`<h1>`, `<h2>`...)

Oui, vous allez me dire qu'au final le résultat (visuel) est le même. Je suis tout à fait d'accord. Mais les balises génériques n'apportent aucun sens à la page et ne peuvent pas être comprises par l'ordinateur. Utilisez toujours d'autres balises plus adaptées quand c'est possible. Google lui-même le conseille pour vous aider à améliorer la position de vos pages au sein de ses résultats de recherche !

Travail en pratique

Réaliser ce site web avec les balises html.



Chapitre 2. Cascading Style Sheet

CSS (Cascading Style Sheets, aussi appelées Feuilles de style) : le rôle du CSS est de gérer l'apparence de la page web (agencement, positionnement, décoration, couleurs, taille du texte. . .). Ce langage est venu compléter le HTML en 1996.

Après avoir passé premier chapitre du cours à ne travailler que sur le HTML, nous allons maintenant découvrir le CSS que j'avais volontairement mis à l'écart. Le CSS n'est pas plus compliqué que le HTML. Il vient le compléter pour vous aider à mettre en forme votre page web.

Il faut savoir qu'aux débuts du Web, CSS n'existait pas. En fait, il n'y avait initialement que le langage HTML. Le HTML est né en 1991 et CSS en 1996. Alors, vous vous dites sûrement : comment faisait-on la mise en forme de 1991 à 1996 ?

Eh bien, uniquement en HTML! Il y avait en effet des balises HTML dédiées à la mise en forme. `<font color="#aab1c3">`, par exemple, permettait de définir la couleur du texte.

Cependant, les pages HTML commençaient à devenir assez complexes. Il y avait de plus en plus de balises et c'était un joyeux mélange entre le fond et la forme, qui rendait la mise à jour des pages web de plus en plus complexe. C'est pour cela que l'on a créé le langage CSS.

Cependant, le CSS n'a pas été adopté immédiatement par les webmasters, loin de là. Il fallait se défaire de certaines mauvaises habitudes et cela a pris du temps. Encore aujourd'hui, on peut trouver des sites web avec des balises HTML de mise en forme, anciennes et obsolètes, comme `<font>` !

2.1. Prise en charge des navigateurs

Tout comme le HTML, le CSS a évolué. Je vous avais indiqué qu'il y avait quatre versions importantes de CSS :

- CSS 1 ;
- CSS 2.0 ;
- CSS 2.1 ;
- CSS 3.

En fait, la version CSS 3 n'est pas encore totalement finalisée (ce n'est pas encore une version officielle). Cependant, elle est bien avancée et aujourd'hui déjà bien prise en charge par de nombreux navigateurs, ce qui fait qu'on peut déjà s'en servir. Il serait dommage de passer à côté car CSS 3 apporte de nombreuses fonctionnalités à CSS (leur nombre double par rapport à CSS 2.1 !). Nous nous baserons donc dans ce cours sur CSS 3, qui reprend et complète la plupart des fonctionnalités de CSS 2.1.

Ce sont les navigateurs web qui font le travail le plus complexe : ils doivent lire le code CSS et comprendre comment afficher la page.

Au début des années 2000, Internet Explorer était le navigateur le plus répandu mais sa gestion du CSS est longtemps restée assez médiocre (pour ne pas dire carrément mauvaise). C'était la grande époque de la version 6 (IE6), hélas encore utilisée aujourd'hui par une petite partie des internautes (heureusement, cette proportion tend à diminuer).

Depuis, de nombreux navigateurs sont arrivés et ont chahuté Internet Explorer : Mozilla Firefox bien sûr, mais aussi Google Chrome. Et je ne vous parle pas du succès des Mac et iPhone avec leur navigateur Safari. Cela a incité Microsoft à réagir et publier (après une longue période d'inactivité) IE 7, puis IE 8 et IE 9. On parle déjà de IE 10.

Plus le navigateur est vieux, moins il connaît de fonctionnalités CSS.

2.2. Où écrit-on le CSS ?

Vous avez le choix car 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) ;

Je vais vous présenter ces 3 méthodes, mais sachez d'ores et déjà que la première... est la meilleure.

Dans un code CSS, on trouve 3 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 (sauf s'il me prend une envie sadique de vous faire souffrir).

Les valeurs : à chaque propriété CSS on doit indiquer une valeur. Par exemple, pour la couleur, il faut indiquer le nom de la couleur. Pour la taille, il faut indiquer quelle taille on veut, etc.

Schématiquement, une feuille de style CSS ressemble donc à ça :

Code : CSS

```
balise1
{
  Propriété: valeur;
  Propriété: valeur;
  Propriété: valeur;
}
```

```
balise2
{
  Propriété: valeur;
  Propriété: valeur;
  Propriété: valeur;
  Propriété: valeur;
}
```

Etc.

2.2.1. Dans un fichier .css (recommandé)

Comme je viens de vous le dire, le plus souvent on écrit du 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 flexible. Ça nous évite de tout mélanger dans un même fichier. J'utiliserai cette technique tout au long de la suite de ce cours.

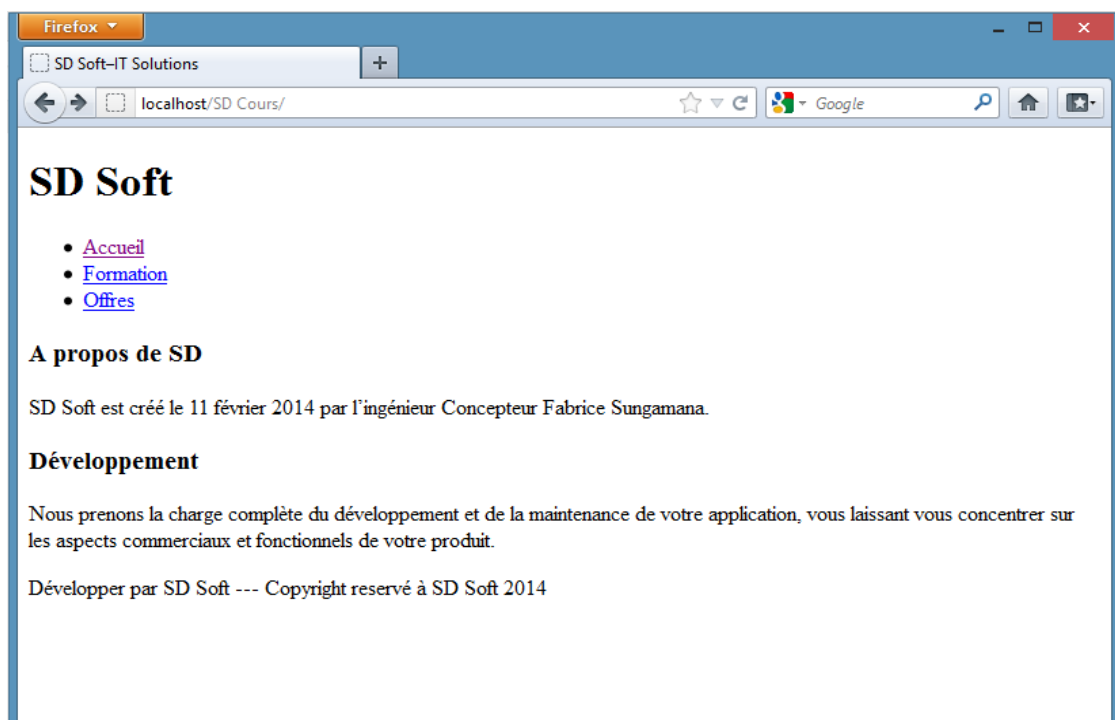
Commençons à pratiquer dès maintenant ! On va partir du fichier HTML suivant :

Code : HTML

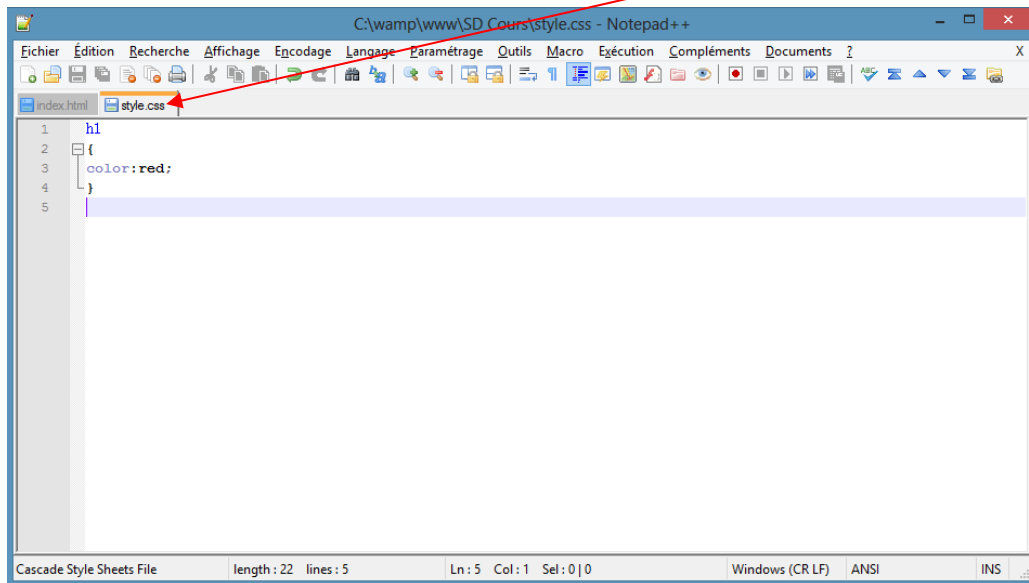
```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="style.css" />
    <title>SD Soft–IT Solutions</title>
  </head>
  <body>
    <header>
      <h1>SD Soft</h1>
    </header>
    <nav>
```

```
<ul>
<li><a href="accueil.html">Accueil</a></li>
<li><a href="formation.html">Formation</a></li>
<li><a href="offres.html ">Offres</a></li>
</ul>
</nav>
<section>
<aside>
<h1>A propos de SD</h1>
<p>SD Soft est créé le 11 février 2014 par l'ingénieur
Concepteur Fabrice Sungamana.</p>
</aside>
<article>
<h1>Développement</h1>
<p>Nous prenons la charge complète du développement et de la
maintenance de votre application, vous laissant vous concentrer
sur les aspects commerciaux et fonctionnels de votre produit.
</p>
</article>
</section>
<footer>
<p>Développer par SD Soft --- Copyright réservé à SD Soft 2014
</p>
</footer>
</body>
</html>
```

Voici le résultat :

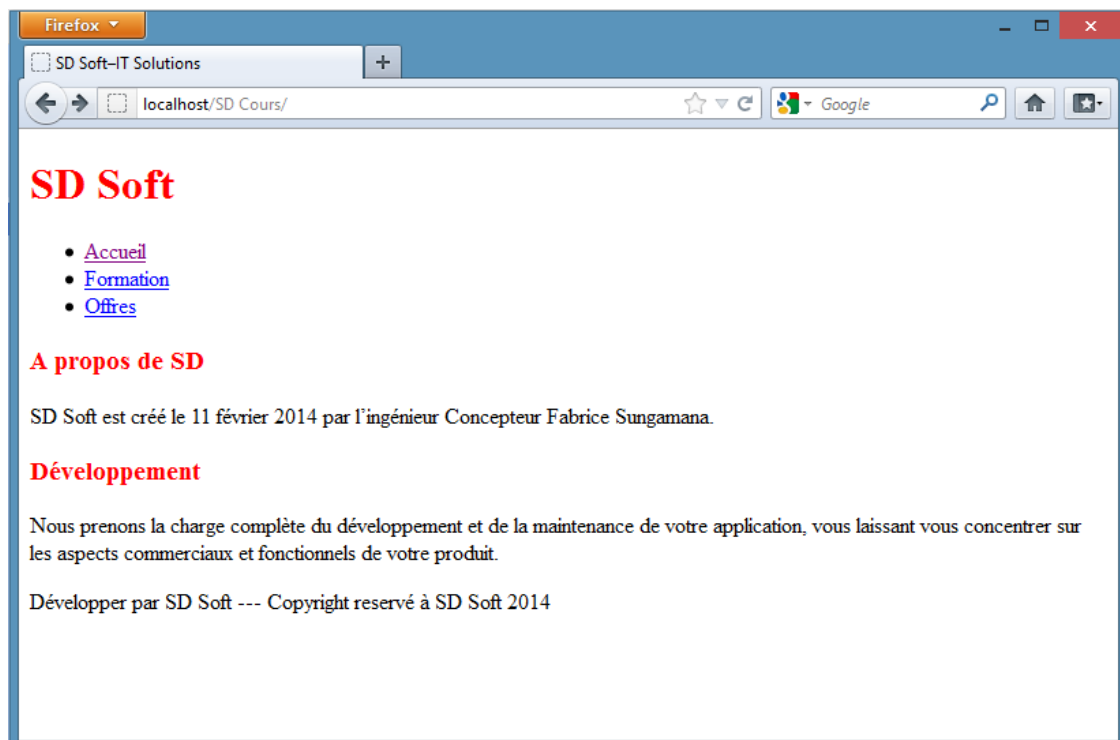


Vous noterez la ligne `<link rel="stylesheet" href="style.css" />` : c'est elle qui indique que ce fichier HTML va être associé à un fichier appelé **style.css** qui va gérer sa mise en forme.



Enregistrez le fichier en lui donnant un nom qui se termine par **.css**, comme **style.css**. Placez ce fichier **.css** dans le même dossier que votre fichier **.html**.

Voici le résultat :



Nota : *Il est inutile d'ouvrir directement le fichier **style.css** dans le navigateur. Il faut ouvrir le fichier **index.html** (il fera automatiquement appel au fichier **style.css**).*

2.2.2. Dans l'en-tête <head> du fichier HTML

Il existe une autre méthode pour utiliser du CSS dans ses fichiers HTML : cela consiste à insérer le code CSS directement dans une balise **<style>** à l'intérieur de l'en-tête **<head>**.

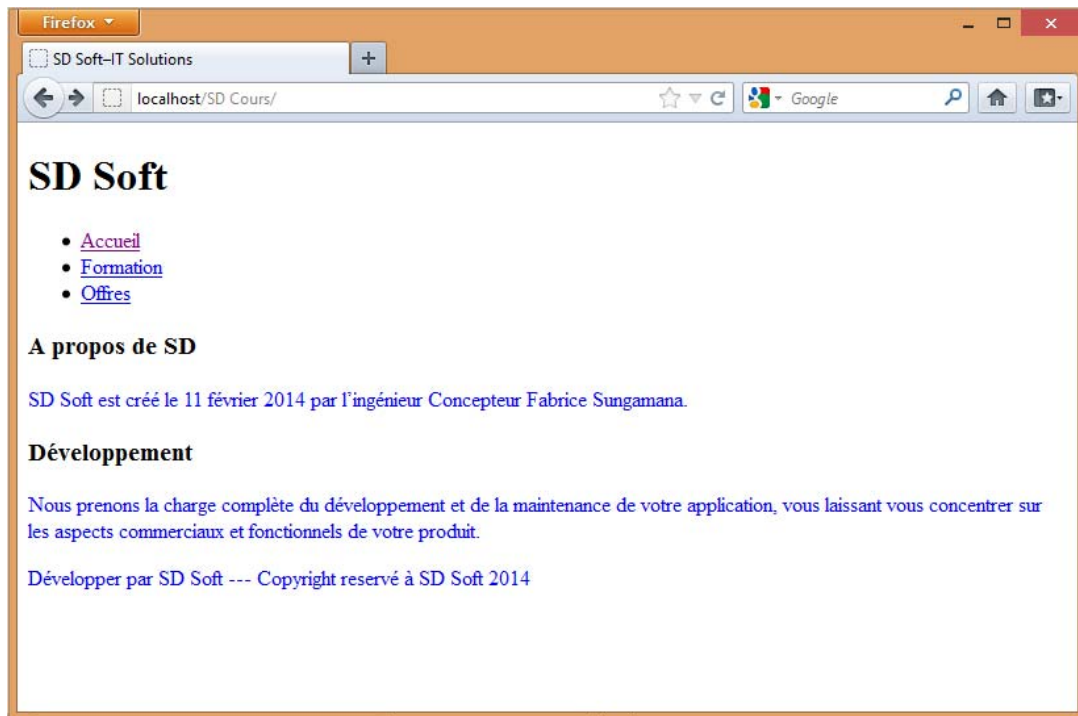
Voici comment on peut obtenir exactement le même résultat avec un seul fichier .html qui contient le code CSS :

Code : HTML

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <style>
      p
      {
        color: blue;
      }
    </style>
    <title>SD Soft–IT Solutions</title>
  </head>
  <body>
    <header>
      <h1>SD Soft</h1>
    </header>
    <nav>
      <ul>
        <li><a href="accueil.html">Accueil</a></li>
        <li><a href="formation.html">Formation</a></li>
        <li><a href="offres.html ">Offres</a></li>
      </ul>
    </nav>
    <section>
      <aside>
        <h1>A propos de SD</h1>
        <p>SD Soft est créé le 11 février 2014 par l'ingénieur
        Concepteur Fabrice Sungamana.</p>
      </aside>
      <article>
        <h1>Développement</h1>
        <p>Nous prenons la charge complète du développement et de la
        maintenance de votre application, vous laissant vous concentrer
        sur les aspects commerciaux et fonctionnels de votre produit.
        </p>
      </article>
    </section>
```

```
<footer>
    <p>Développer par SD Soft --- Copyright réservé à SD Soft 2014
    </p>
</footer>
</body>
</html>
```

Voici le résultat :



2.2.3. Directement dans les balises (non recommandé)

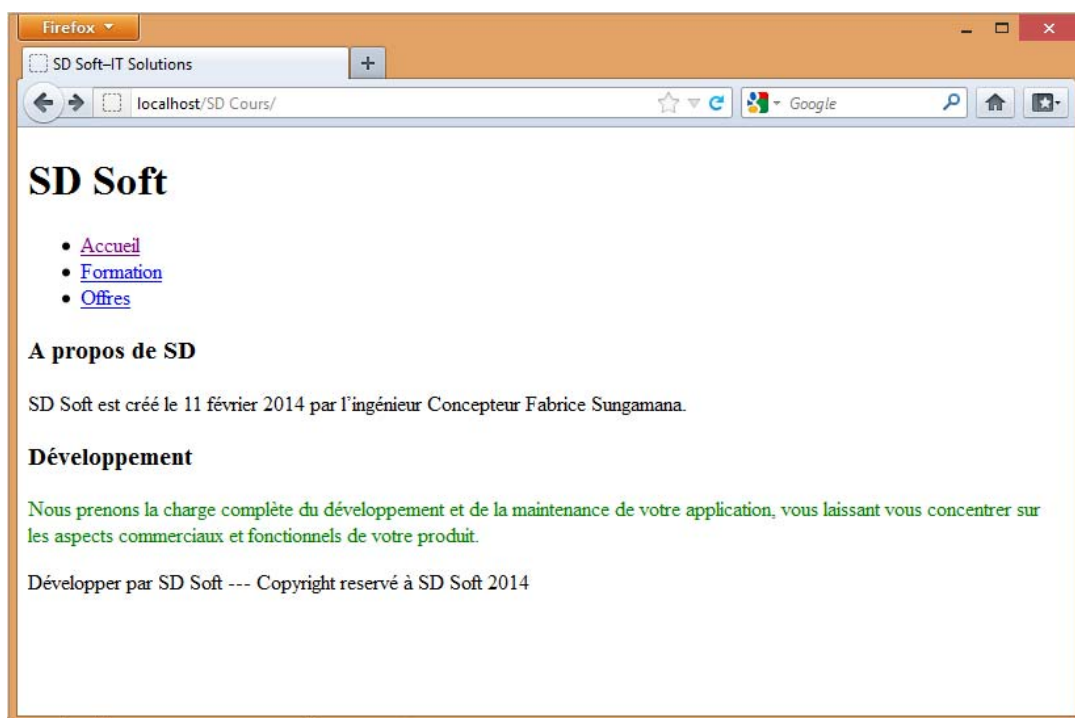
Dernière méthode, à manipuler avec précaution : vous pouvez ajouter un attribut style à n'importe quelle balise. Vous insèrerez votre code CSS directement dans cet attribut :

Code : HTML

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>SD Soft-IT Solutions</title>
  </head>
  <body>
    <header>
      <h1>SD Soft</h1>
    </header>
    <nav>
      <ul>
        <li><a href="accueil.html">Accueil</a> </li>
```

```
<li><a href="formation.html">Formation</a></li>
<li><a href="offres.html ">Offres</a></li>
</ul>
</nav>
<section>
  <aside>
    <h1>A propos de SD</h1>
    <p>SD Soft est créé le 11 février 2014 par l'ingénieur
    Concepteur Fabrice Sungamana.</p>
  </aside>
  <article>
    <h1>Développement</h1>
    <p style="color:green;">Nous prenons la charge complète du
    développement et de la maintenance de votre application, vous
    laissant vous concentrer sur les aspects commerciaux et
    fonctionnels de votre produit. </p>
  </article>
</section>
<footer>
  <p>Développer par SD Soft --- Copyright réservé à SD Soft 2014
  </p>
</footer>
</body>
</html>
```

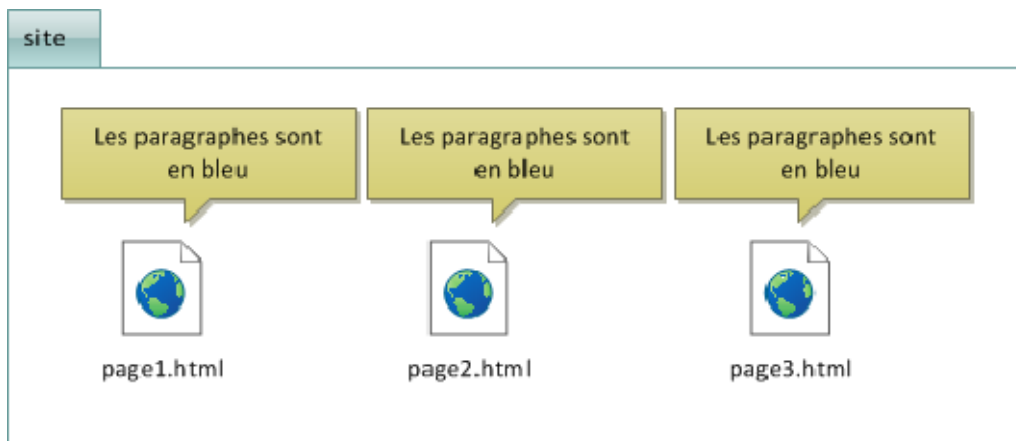
Voici le résultat :



Parmi les trois méthodes qui servent d'écrire le code CSS, je vous recommande fortement de prendre l'habitude de travailler avec la première méthode parce que c'est celle utilisée par la majorité des webmasters.

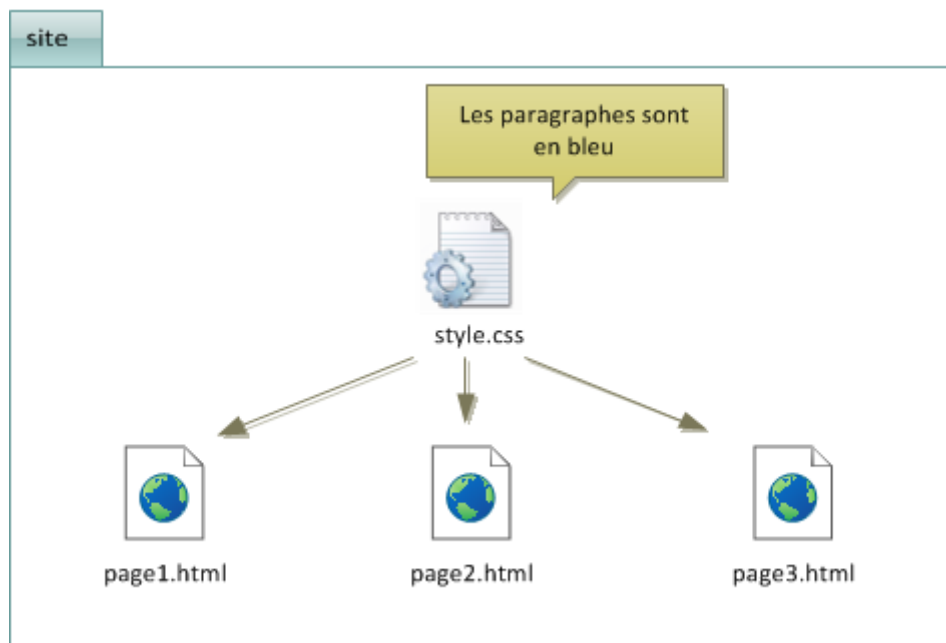
Parce que pour le moment, vous faites vos tests dans un seul fichier HTML. Cependant, votre site sera plus tard constitué de plusieurs pages HTML.

Imaginez : si vous placez le code CSS directement dans le fichier HTML, il faudra copier ce code dans tous les fichiers HTML de votre site ! Et si demain vous changez d'avis, par exemple si vous voulez que vos paragraphes soient écrits en bleu et non en vert, il faudra modifier chaque fichier HTML un à un !



Le code CSS est répété dans chaque fichier HTML : pas pratique !

Si vous travaillez avec un fichier CSS externe, vous n'aurez besoin d'écrire cette instruction qu'une seule fois pour tout votre site !



Le code CSS est indiqué une fois pour toutes dans un fichier CSS : c'est plus simple !

2.3. Appliquer un style

Nous pouvons appliquer le style sur n'importe quel élément, mais il sied de savoir que l'élément **Class** est précédé d'un point tandis que l'élément **Id** est précédé par un dièse (#).

Prenons le code CSS suivant :

Code : CSS

```
h1
{
    color: blue;
}

p
{
    color: green;
}

.important
{
    color: red;
}

#nom de l'ancre
{
    color: green;
}
```

2.4. Les propriétés du CSS

2.4.1. Font-size : cette propriété permet de modifier la taille du texte. Il est à noter que nous distinguons deux types de taille (absolue et relative).

Taille absolue : Pour indiquer une taille absolue, on utilise généralement les pixels.

```
h1
{
    Font-size: 20px;
}
```

Taille relative : C'est la méthode recommandée car le texte s'adapte alors plus facilement aux préférences de tous les visiteurs.

Il y a plusieurs moyens d'indiquer une valeur relative. Vous pouvez par exemple écrire la taille avec des mots en anglais comme ceux-ci :

- xx-small : minuscule ;
- x-small : très petit ;

- small : petit ;
- medium : moyen ;
- large : grand ;
- x-large : très grand ;
- xx-large : gigantesque.

Vous pouvez tester l'utilisation de ces valeurs dans votre code CSS :

```
h1
{
    Font-size: large;
}
```

2.4.2. Font-family : permet de déterminer la police.

```
h1
{
    Font-family: tahoma;
}
```

Nota : *pour mettre plusieurs polices, il faut tout simplement les séparer par des virgules. En général, on indique en tout dernier la police serif, ce qui correspond à une police par défaut c'est-à-dire ne s'applique que si aucune autre police n'a été trouvée.*

```
h1
{
    Font-family: tahoma, arial, "times new roman", serif;
}
```

Conseil : *il est conseillé d'entourer de guillemets une police qui comporte des espaces exemple de times new roman ou arial black...*

2.4.3. Font-style : permet de mettre les textes en italique au niveau de CSS, il regorge trois valeurs : italic, oblique et normal.

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

2.4.4. Font-weight : permet de mettre les textes en gras, il a deux valeurs : bold et normal.

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

2.4.5. Text-decoration : permet de mettre le soulignement sur le texte ou le clignotement de celui-ci.

Voici les différentes valeurs qu'elle peut prendre :

- underline : souligné.
- line-through : barré.
- overline : ligne au-dessus.
- blink : clignotant. Ne marche pas sur tous les navigateurs (Internet Explorer et Google Chrome notamment).
- none : normal (par défaut).

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

2.4.6. Text-align : permet de mettre l'alignement à gauche, à droite, centré et justifié aux textes. Il prend les valeurs : left (gauche), right (droite), center (centré) et justify (justifié).

```
h1
{
  Text-align: left;
}
```

2.4.7. background-color : permet de mettre la couleur de fond dans une page web.

```
body
{
  Background-color: green;
}
```

2.4.8. background-image : permet de mettre l'image de fond dans une page web. Cette propriété s'accompagne avec les options ci-après :

Background-attachment pour fixer le fond, il a deux valeurs : fixed et scroll.

Background-repeat pour la répétition du fond, il a quatre valeurs : no-repeat, repeat-x, repeat-y et repeat.

Background-position pour arranger la position du fond, il est souvent combiné avec background-repeat :no-repeat. Le background-position est exprimé en pixel ou utiliser les valeurs en anglais : top, bottom, left, center et right.

```
body
{
  Background-image: url("SDsoft.png");
  Background-attachment :fixed ;
  Background-repeat :no-repeat ;
  Background-position :30px 50px;
}
```

Nous pouvons aussi combiner les propriétés de background c'est background-image, background-repeat, background-attachment et background-position.

```
body
{
    Background: url("SDsoft.png") fixed no-repeat top right;
}
```

Combiner plusieurs images de fond

Il est possible de donner plusieurs images de fond à un élément. Pour cela, il suffit de séparer les déclarations par une virgule, comme ceci :

```
body
{
    Background: url("logo.png") fixed no-repeat top right, url("fond.png")
fixed;
}
```

2.4.9. Border : permet de mettre la bordure dans une page web. nous en distinguons les valeurs ci-après : none, solid, dotted, dashed, double, groove, ridge, inset, outset.

```
h1
{
    Border : 1px solid white;
    Border-radius:10px
}
```

2.5. Des commentaires dans du CSS

Comme en HTML, il est possible de mettre des commentaires. Les commentaires ne seront pas affichés, ils servent simplement à indiquer des informations pour vous, par exemple pour vous y retrouver dans un long fichier CSS.

D'ailleurs, vous allez vous en rendre compte, en général le fichier HTML est assez court et la feuille CSS assez longue (si elle contient tous les éléments de style de votre site, c'est un peu normal). Notez qu'il est possible de créer plusieurs fichiers CSS pour votre site si vous ressentez le besoin de séparer un peu votre code CSS (en fonction des différentes sections de votre site, par exemple)..

Donc, pour faire un commentaire, c'est facile ! Tapez `/*`, suivi de votre commentaire, puis `*/` pour terminer votre commentaire. Vos commentaires peuvent être écrits sur une ou plusieurs lignes. Par exemple :

```
/*
Mise en forme de votre site est effectuée par SD soft
*/
```

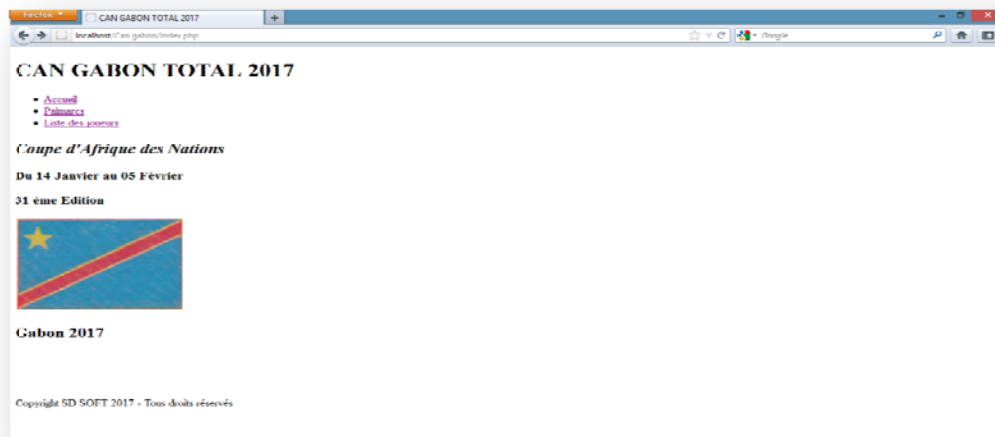
```
h1
{
    color: blue; /* Les paragraphes seront en bleu */
}
```

Travail pratique complet

Balisage :

```
<!DOCTYPE html>
<html>
  <head>
    <metacharset="utf-8" />
    <title>CAN GABON TOTAL 2017</title>
  </head>
  <body>
    <header>
      <h1>CAN GABON TOTAL 2017</h1>
    </header>
    <nav>
      <ul>
        <li><a href="index.php">Accueil</a></li>
        <li><a href="palmares.php">Palmares</a></li>
        <li><a href="joueurs.php">Liste des joueurs</a></li>
      </ul>
    </nav>
    <section>
      <h1 class="jen"><em>Coupe d'Afrique des Nations</em></h1>
      <h3>Du 14 Janvier au 05 Février</h3>
      <h3>31 ème Edition</h3>
      
      <h2>Gabon 2017</h2>
    </section>
    <br/><br/><br/>
    <footer>
      <p>Copyright SD SOFT 2017 - Tous droits réservés</p>
    </footer>
  </body>
</html>
```

Résultat :



Code css :

```
nav
{
border:1px solid green;
width:180px;
border-radius:10px;
display:inline-block;
vertical-align:top;
margin-left:20px;
}
section
{
display:inline-block;
vertical-align:top;
border:1px solid white;
width:900px;
border-radius:10px;
margin-left:7%;
}
li
{
color:green;
}
h1
{
color:white;
}
h2
{
color:white;
text-align:right;
margin-right:10%;
}
h3
{
margin-left:3%;
color:green;
}
img
{
margin-left:35%;
}
.jen
{
font-size:50px;
margin-left:3%;
}
```

```
.sdf
{
font-size:50px;
text-align:center;
border-bottom:1px solid green;
color:green;
}
body
{
font-family: Tahoma, Arial Black, Arial, Verdana, serif;
background-image:url(logo.png);
background-position:20px 5px;
background-repeat:no-repeat;
background-color:orange;
}
footer
{
text-align:center;
border-top:1px solid green;
color:green;
}
```

Résultat :

