

# HTML/CSS



Formatrice: Anouchka MINKOUE OBAME

# Le langage HTML

- HTML est le langage universel utilisé pour communiquer sur le Web. L'information sera ainsi transportée sur le réseau (éventuellement sur Internet), pour aboutir sur un poste client (Micro-ordinateur) puis interpréter grâce à un programme appelé navigateur ou browser.
- Le logiciel que l'on appelle un browser permet de surfer sur le Net et d'afficher sur l'écran les "pages" qu'il a interceptées. Il existe différents browser, Chrome, Mozilla FireFox et Internet Explorer de Microsoft.... Chaque browser a sa propre façon de travailler.

# HTML: Les langages auxiliaires

**CSS** (Cascading Style Sheets): Langage pour ajouter du style aux éléments HTML

**JavaScript**: Langage de script côté client permet de dynamiser les pages web.

**PHP** et **ASP** s'exécutent sur le serveur. Ils permettent au développeur d'injecter dans une page Web non seulement les balises HTML mais aussi tous les codes Javascript nécessaire à son interactivité.

# Outils pour l'intégration et le développement web

## Framework

- **Framework** signifie « **cadre de travail** » :

un **ensemble cohérent de composants** éprouvés et réutilisables (bibliothèques, classes, helpers...) ;

un **ensemble de préconisations** pour la conception et le développement d'applications.

Exemple: Ruby On Rails, Symfony, Bootstrap

# Outils pour l'intégration et le développement web

## CMS

- **CMS** signifie Content Management System, pour système de gestion de contenu. C'est un type d'application web permettant de séparer le contenu de la mise en forme pour des sites web, et fournissant des moyens simplifiés d'éditer ce contenu. De cette manière, il est possible de maintenir et faire vivre un site web sans la moindre connaissance technique dans les métiers du Web.

Exemple : Wordpress, Drupal, Joomla...

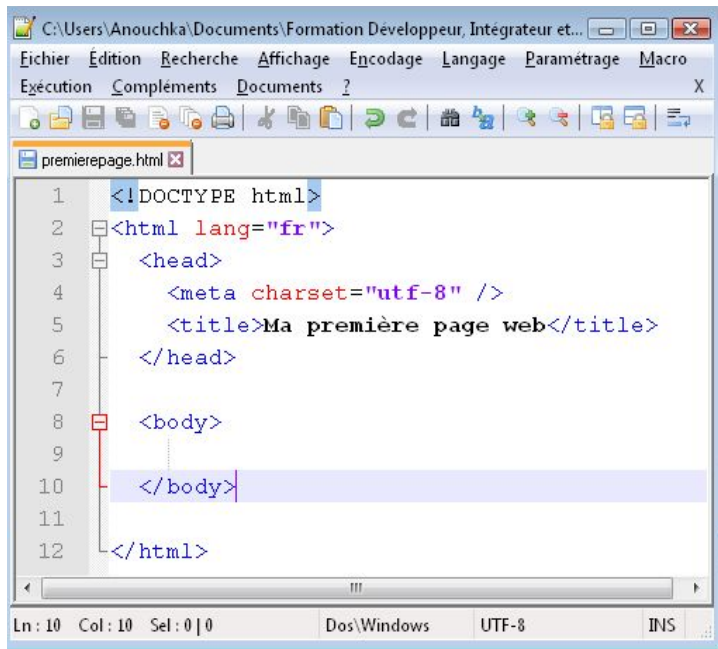
# TRAVAIL EN LOCAL VS EN PRODUCTION

Lorsque vous codez, vous pouvez travailler soit en local, soit en production.

**Travailler en local** signifie travailler avec des fichiers enregistrés sur votre propre ordinateur. Comme les fichiers ne sont que sur votre ordinateur, vous seul pourrez voir ce que vous faites.

**Travailler en production**, au contraire, signifie intervenir sur des fichiers qui sont déjà stockés sur un serveur web, c'est-à-dire travailler sur des fichiers déjà accessibles via Internet.

# Première page HTML



The screenshot shows a text editor window titled "C:\Users\Anouchka\Documents\Formation Développeur, Intégrateur et...". The menu bar includes "Fichier", "Édition", "Recherche", "Affichage", "Encodage", "Langage", "Paramétrage", and "Macro". The toolbar contains various icons for file operations. The active file is "premierepage.html". The code is as follows:

```
1 <!DOCTYPE html>
2 <html lang="fr">
3   <head>
4     <meta charset="utf-8" />
5     <title>Ma première page web</title>
6   </head>
7
8   <body>
9
10  </body>
11
12 </html>
```

The status bar at the bottom indicates "Ln : 10 Col : 10 Sel : 0 | 0", "Dos\Windows", "UTF-8", and "INS".

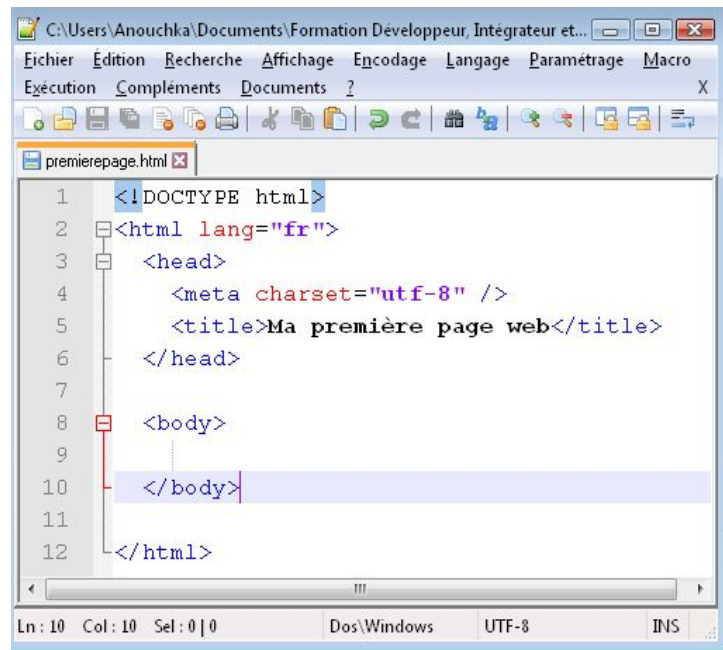
# Structure minimale d'une page en HTML

Pour qu'une **page HTML soit déclarée valide**, elle doit obligatoirement comporter certains éléments et suivre le schéma :

## Le doctype

Nous devons toujours démarrer une page HTML en précisant le doctype de notre document. Comme son nom l'indique, le doctype sert à préciser le type du document.

Dans la balise doctype, on va préciser le langage utilisé, à savoir le HTML dans notre cas.

A screenshot of a text editor window titled 'premierepage.html'. The editor shows the following HTML code:

```
1 <!DOCTYPE html>
2 <html lang="fr">
3   <head>
4     <meta charset="utf-8" />
5     <title>Ma première page web</title>
6   </head>
7
8   <body>
9
10  </body>
11
12 </html>
```

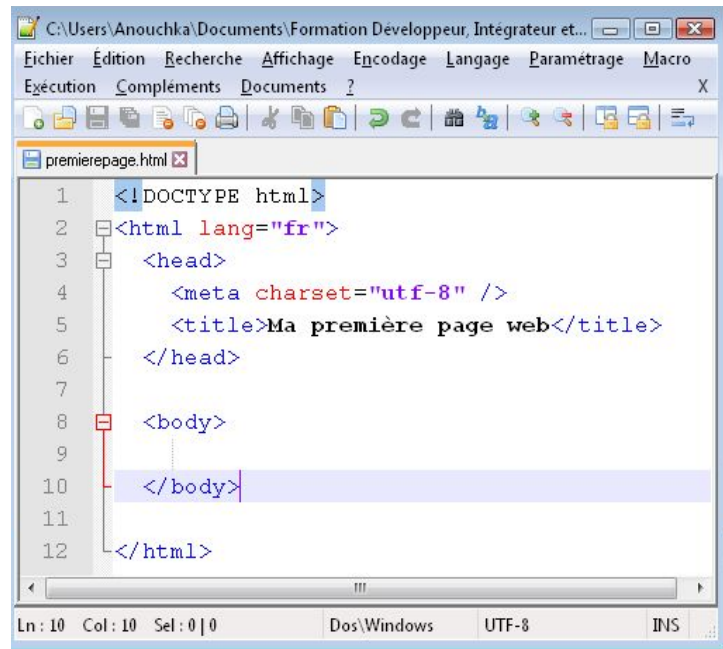
The code is color-coded: blue for tags, red for attributes, and black for text. The status bar at the bottom indicates 'Ln : 10 Col : 10 Sel : 0 | 0', 'Dos/Windows', 'UTF-8', and 'INS'.



# Structure minimale d'une page en HTML

## L'élément html

Après le document, nous devons également mentionner un élément html. Cet élément est composé de deux balises <html> et </html> et va représenter notre page HTML en soi. On va donc insérer tout le contenu de notre page (et donc les autres éléments) à l'intérieur de celui-ci.

A screenshot of a text editor window titled 'premierepage.html'. The editor shows the following HTML code:

```
1 <!DOCTYPE html>
2 <html lang="fr">
3   <head>
4     <meta charset="utf-8" />
5     <title>Ma première page web</title>
6   </head>
7
8   <body>
9
10  </body>
11
12 </html>
```

The code is color-coded: blue for tags, red for attributes, and black for text. The status bar at the bottom indicates 'Ln: 10 Col: 10 Sel: 0 | 0', 'Dos\Windows', 'UTF-8', and 'INS'.

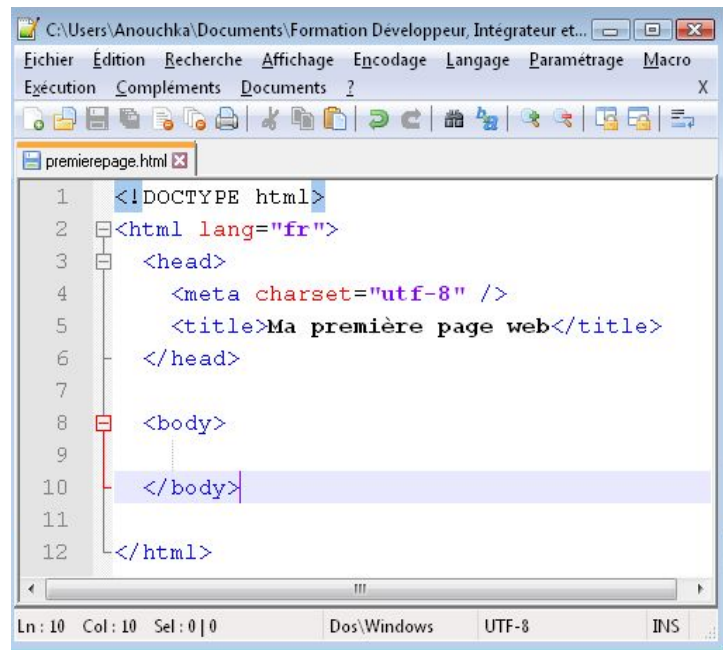
# Structure minimale d'une page en HTML

## Les éléments head et body

A l'intérieur de l'élément html, nous devons à nouveau indiquer deux éléments : head et body.

L'élément head va contenir des meta informations relatives à la page, c'est-à-dire des informations générales dont la page va avoir besoin pour fonctionner, comme le titre de la page ou encore le type d'encodage utilisé.

L'élément body va lui contenir tout le contenu « visible » de la page : les textes, images, liens, vidéos, etc.

A screenshot of a text editor window titled 'premierepage.html'. The editor shows the following HTML code:

```
1 <!DOCTYPE html>
2 <html lang="fr">
3   <head>
4     <meta charset="utf-8" />
5     <title>Ma première page web</title>
6   </head>
7
8   <body>
9
10  </body>
11
12 </html>
```

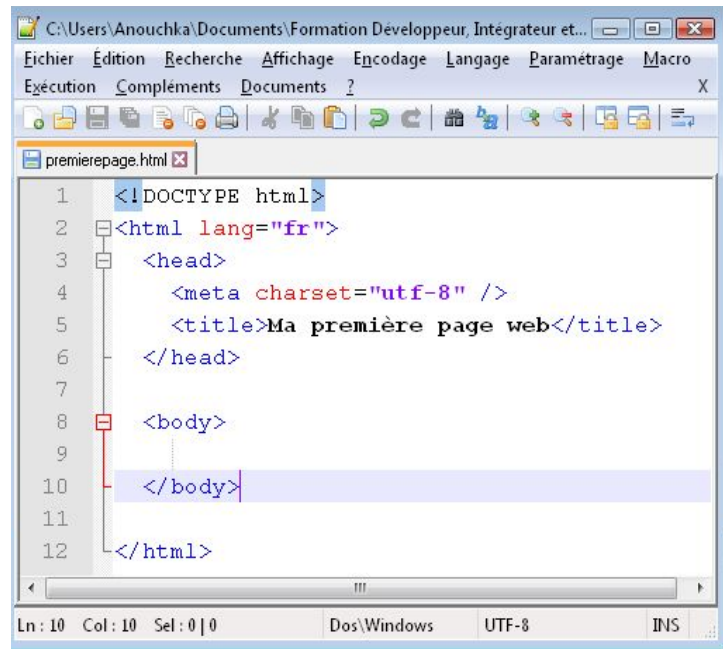
The code is color-coded: DOCTYPE is blue, html is blue, lang is red, fr is red, head is blue, meta is blue, charset is red, utf-8 is red, title is blue, Ma première page web is black, body is blue, and the closing tags are blue. The status bar at the bottom shows 'Ln : 10 Col : 10 Sel : 0 | 0', 'Dos\Windows', 'UTF-8', and 'INS'.

# Structure minimale d'une page en HTML

## Les éléments title et meta

Nous allons devoir indiquer au moins deux autres éléments à l'intérieur de notre élément head : l'élément title, qui va contenir le titre de la page et un élément meta avec son attribut charset qui va nous permettre de définir l'encodage de la page.

L'encodage est très important pour que tous nos caractères, notamment les accents, s'affichent normalement dans notre page. Pour nous autres, français, nous utiliserons la valeur utf-8.



```
1 <!DOCTYPE html>
2 <html lang="fr">
3   <head>
4     <meta charset="utf-8" />
5     <title>Ma première page web</title>
6   </head>
7
8   <body>
9
10  </body>
11
12 </html>
```

The screenshot shows a text editor window titled 'premierepage.html'. The code is as follows:

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="utf-8" />
    <title>Ma première page web</title>
  </head>
  <body>
  </body>
</html>
```

The status bar at the bottom indicates 'Ln : 10 Col : 10 Sel : 0 | 0', 'Dos/Windows', 'UTF-8', and 'INS'.

# LES FONDATIONS DU HTML

# ELEMENTS, BALISES & ATTRIBUTS

- Éléments =
  - Définissent des objets dans notre page web
    - L'élément p définit un paragraphe,
    - Les éléments h1, h2, ... , h6 définissent des titres,
    - L'élément a définit un lien...
  - Généralement constitués d'une paire de balises :
    - Balise ouvrante : <p>
    - Balise fermante : </p>
    - Exception : balises orphelines comme <br/>

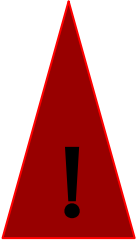
- Doctype : `<!DOCTYPE html>`
- Éléments :
  - html,
  - head,
  - title,
  - meta,
  - body.

# BONNES PRATIQUES, REGLES & COMMENTAIRES

- Vous pouvez imbriquer des balises l'une dans l'autre mais vous devez les refermer dans le bon ordre : `<a> <b> </b> </a>`.
- Indentez votre code et commentez le pour le rendre plus lisible, plus professionnel et plus simple à comprendre.

# BONNES PRATIQUES, REGLES & COMMENTAIRES

- `<!-- Voici un commentaire en HTML -->`

 ! Tout le monde peut voir votre code HTML ! N'écrivez donc pas d'infos sensibles en commentaires comme des mots de passe !



# HEADING, PARAGRAPH, BREAK

- L'élément `p` définit un paragraphe.
- L'élément `br` crée un retour à la ligne.
- Les éléments `h1`, `h2`,... jusqu'à `h6` définissent des titres (par ordre d'importance).
- Vous ne devez jamais utiliser des éléments pour appliquer un style à votre contenu !

# **STRONG, MARK, EMPHASIS**

L'élément strong est utilisé pour définir un contenu comme important.

- L'élément em est utilisé pour définir un contenu comme assez important.
- L'élément mark est utilisé pour faire ressortir du contenu.

- Pour créer une liste non-ordonnée, on utilise les éléments `ul` (pour la liste) et `li` (pour les éléments de la liste).
- Pour créer une liste ordonnée, on utilise `ol` et `li`.

- Vous pouvez très simplement imbriquer des listes en HTML :

```
<ul>  
  <li> Élément 1 </li>  
  <li> Élément 2  
    <ol>  
      <li> Élément 1 ol </li>  
      <li> Élément 2 ol </li>  
    </ol>  
  </li>  
</ul>
```

## LISTES DE DÉFINITIONS & LISTES IMBRIQUÉES

- Les listes de définition sont utilisées pour... définir des termes.
  - On utilise les éléments dl, dt et dd.
  - Le terme à définir doit toujours être placé avant la description.
  - On peut avoir plusieurs descriptions pour le même terme ou plusieurs termes pour une description.

# LISTES DE DÉFINITIONS & LISTES IMBRIQUÉES

## HTML ▼

```

1 < p > Les développeurs Web utilisent ces langages lorsqu'ils créent un site: </ p >
2
3 < dl >
4   < dt > HTML </ dt >
5   < dd > < em > HyperText Markup Language </ em > décrit la structure de la page et son
    contenu. </ dd >
6   < dt > CSS </ dt >
7   < dd > < em > Feuilles de style en cascade </ em > décrit l'apparence d'un site et même,
    dans une certaine mesure, sa réponse à certains événements. </ dd >
8   < dt > JavaScript </ dt >
9   < dd > JavaScript est le langage de programmation utilisé pour définir la logique et la
    fonction d'un site au-delà de la simple apparence. Tout calcul ou "réflexion" est fait en
    utilisant JavaScript. </ dd >
</ dl >

```

## JavaScript + No-Library (pure JS) ▼

```

1

```

## CSS ▼

```

1 p ,
2 dl {
3   police : 1rem 'Fira Sans' , sans-serif ;
4 }
5
6 dl > dt {
7   poids de la police : gras ;
8   décoration de texte : souligner ;
9 }

```

Web developers use these languages when they create a site:

### HTML

*HyperText Markup Language* describes the structure of the page and its contents.

### CSS

*Cascading Style Sheets* describes how a site looks and even to some extent how it responds to certain events.

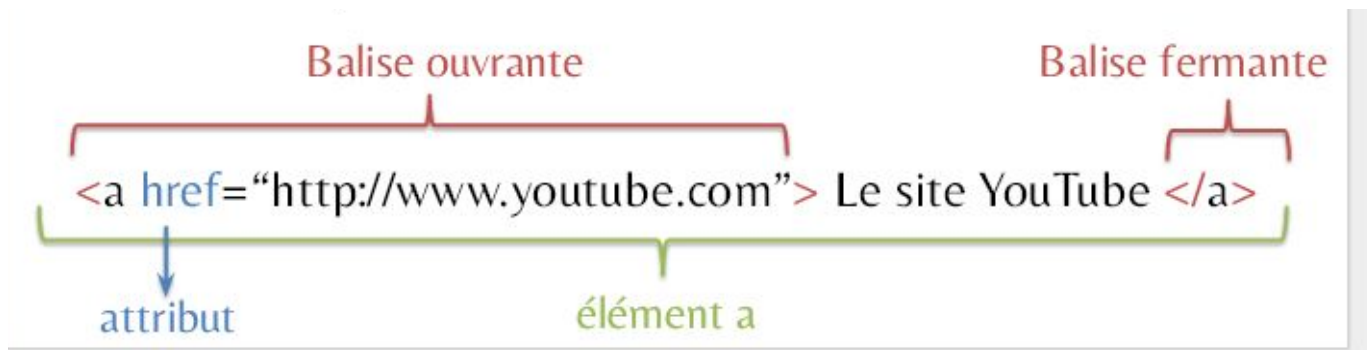
### JavaScript

JavaScript is the programming language used to define the logic and function of a site beyond simple look and feel. Any computation or "thinking" is done using JavaScript.

# LIENS INTERNES ET EXTERNES

- Liens internes = entre 2 pages d'un même site.
- Liens externes = d'un site vers un autre site.
- Pour créer des liens, on utilise l'élément **a** avec son attribut href (Hypertext reference).

- Attributs =
  - Propriétés utilisées pour donner des indications supplémentaires aux éléments.
  - Ex: Indiquer la cible d'un lien.





# LIENS INTERNES ET EXTERNES

- Pour créer des liens internes, on utilise un chemin relatif. 3 cas :
  - Même dossier : href = « page2.html »
  - Sous-dossier : href = « sous\_dossier/page2.html »
  - Dossier parent : href=« ../page2.html »
- Pour des liens externes, on spécifie un chemin absolu. La valeur de l'attribut href est l'adresse du site.
- Pour ouvrir le lien dans une nouvelle fenêtre / onglet, on utilise l'attribut target et sa valeur « \_blank ».

# AUTRES TYPES DE LIENS

- Lien menant à un autre endroit de la même page web : spécifier un id puis #.  
<a href="#dernieres-nouvelles">Les dernières nouvelles</a>  
<div id="dernieres-nouvelles">...</div>

[Voir exemple](#)

- Lien pour envoyer un mail avec mailto:  
<a href="<mailto:minkoueobamea@gmail.com>">minkoueobamea@gmail.com</a>.

Cette syntaxe est aussi possible:

mailto:minkoueobamea@gmail.com  
?subject=Votre  
&cc=partenaire@email.com  
&bcc=collaborateur@email.com  
&body=Bonjour

devis

- Lien pour télécharger un fichier en précisant un chemin relatif.  
`<a href="courshtml.pdf" download>Télécharger le cours HTML</a>`
- Pour créer un lien cliquable vers un numéro de téléphone:  
Book now, call `<a href="tel:01234567890">01234 567 890</a>`

- Créés pour améliorer la sémantique et mieux structurer les pages web.
- Éléments introduits : header, nav, article, section, aside et footer.
- Vont avoir un rôle de plus en plus important dans le futur concernant l'optimisation du référencement.

# VALIDATION & COMPATIBILITÉ

- Différents navigateurs peuvent produire différents résultats à partir d'un même code.
- Pourquoi ? Différents navigateurs et différentes versions de chaque navigateurs coexistent + navigateurs mobiles !
- Nécessité de tester son code sous différents navigateurs !

# VALIDATION & COMPATIBILITE

- Le W3C met deux outils à notre disposition :
  - Valideur HTML : <http://validator.w3.org>
  - Valideur CSS : <http://jigsaw.w3.org/css-validator>
- Vous devez vous efforcer d'avoir toujours un code valide.