

TALIS BUSINESS SCHOOL 2019-2020

COURS HTML5

HTML



Sommaire

Introduction au HTML

- Les bases du HTML
- Structure d'une page HTML
- Les balises
- Balises <head> ... </head>
 - La balise <title>
 - La balise <link>
 - La balise <meta>
- Le découpage d'un document HTML5
 - La balise <header>
 - La balise <section>
 - La balise <article>
 - La balise <nav>
 - La balise <aside>
- Le corps du document HTML
 - La balise <body>
 - Les attributs
 - Les commentaires
 - Les balises de mise en forme
 - Indenter son code

Plus de balises HTML

- Les listes
 - Les listes non ordonnées
 - Les liste ordonnées
- Les images
 - La balise
 - <figure> & <figcaption>
 - La balise <iframe>
 - Les balises <video> et <audio>
- Les liens
 - La balise <a>
 - Lien hypertexte
 - Lien hypertexte interne (ancree)
- Les tableaux
 - La balise <table>
 - La balise <td> ou <th>
 - La balise <thead>
 - La balise <tbody>
- Les formulaires
- Focus sur l'attribut type=""

Les bases du HTML

HTML ([HyperText Marked Up Language](#)) est le langage de balisage standard permettant la création de pages web.

Celui-ci permet de décrire la structure sémantique et logique d'une page web grâce à un système de balisage, et permet l'intégration de texte, images, tableaux, formulaires, etc...

Le HTML est un langage universel de description (vs langage de programmation) très simple d'accès.

Il est systématiquement associé avec le langage CSS (Cascading Style Sheets) permettant de modifier le style d'un élément HTML. C'est lui qui est responsable des couleurs, tailles, mise en pages etc..

De plus, il est très souvent associé avec le langage de programmation JavaScript permettant de rendre les pages web interactive et dynamique.

Un exemple de page HTML sans css

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean pharetra iaculis nibh, a viverra neque scelerisque a. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Proin suscipit, odio sit amet aliquet vulputate, lectus nulla egestas ante, ut tempor tellus turpis eget ligula.

Un sous-titre

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean pharetra iaculis nibh, a viverra neque scelerisque a. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Proin suscipit, odio sit amet aliquet vulputate, lectus nulla egestas ante, ut tempor tellus turpis eget ligula. Maecenas vehicula viverra elit, iaculis ornare neque volutpat sed. Fusce vulputate neque quis justo vulputate, et maximus dui ullamcorper. Nullam in rutrum neque, eu aliquet leo. Nunc facilisis facilisis nisl, imperdiet pretium sapien laoreet a. Mauris tempor turpis vel eros vehicula facilisis. Pellentesque dictum finibus fringilla. Curabitur quis hendrerit ex, at ultrices nunc. Cras at faucibus quam. Sed eleifend elit eu dignissim volutpat. Suspendisse accumsan rhoncus blandit.

Un autre sous-titre

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean pharetra iaculis nibh, a viverra neque scelerisque a. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Proin suscipit, odio sit amet aliquet vulputate, lectus nulla egestas ante, ut tempor tellus turpis eget ligula.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean pharetra iaculis nibh, a viverra neque scelerisque a. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Proin suscipit, odio sit amet aliquet vulputate, lectus nulla egestas ante, ut tempor tellus turpis eget ligula.

La structure d'une page HTML

Par défaut, une page HTML a toujours le même squelette

`<!DOCTYPE html>` permet d'indiquer à notre navigateur que le document est rédigé en HTML.

`<html>` est la balise principale de notre document HTML, elle va contenir notre site. Un document HTML contient une tête et un corps.

`<head>` contiendra toutes les informations relatives à notre page mais qui ne font pas parti du contenu de la pages. La plus populaire est la balise `<title>`.

`<title>` se trouve à l'intérieur de la balise `<html>` et contiendra le titre de la page HTML, qui sera affiché dans l'onglet du navigateur.

`<body>` contiendra tout le contenu de la page HTML.

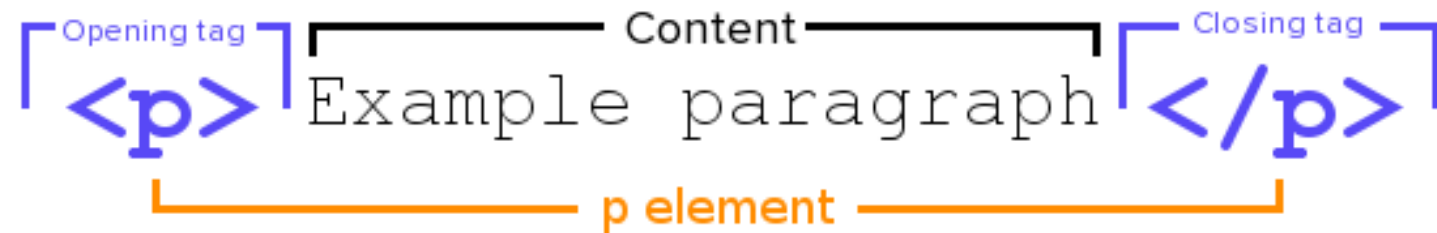
```
<!DOCTYPE html>
<html lang="fr">
<head>
  <title></title>
</head>

<body>

</body>
</html>
```

Les balises

La description d'un document HTML se fait à l'aide de balises (Tags en anglais)
Les balises HTML s'écrivent de cette façon :



Il y a donc 3 éléments obligatoires pour la rédaction des balises HTML :

- 1 : Une balise d'ouverture
- 2 : Le contenu
- 3 : Une balise de fermeture qu'on peut distinguer grâce au "/"

Dans certains cas, il ne sera pas nécessaire d'utiliser une balise fermante, mais nous les verrons plus tard.

La balise <head>

La balise <head> fait partie du squelette de base d'un document HTML. Elle fournit des informations générales appelées métadonnées, comme le titre, ou des liens vers des fichiers CSS et autres scripts.

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <title></title>
</head>

<body>

</body>
</html>
```

La balise <title>

La balise <title> définit le titre du document, ce que vous voyez dans un onglet quand vous naviguez sur le web. Elle ne peut contenir que du texte, et s'avère être indispensable pour une bonne normalisation W3C

Le World Wide Web Consortium, abrégé par le sigle W3C, est un organisme de standardisation à but non lucratif, fondé en octobre 1994 chargé de promouvoir la compatibilité des technologies du Web. #Wikipédia

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <title>Titre du site</title>
</head>

<body>

</body>
</html>
```


La balise <link>

La balise <link> permet de lier la page avec une ressource externe.
Dans ce cas précis, elle permet de relier le document HTML à un fichier CSS.

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <title>Titre du site</title>
  <link rel="stylesheet" type="text/css" href="css/style.css">
</head>

<body>

</body>
</html>
```

Elle possède 3 attributs :

- **rel** : indique le type de relation avec la ressource
- **type** : indique le type de la ressource
- **href** : indique le nom et le chemin de la ressource

La balise <meta>

La balise <meta> permet de fournir une description du site avec des mots clés et des phrases décrivant le contenu du site. Les métadonnées ne s'affichent pas sur la page, mais seront analysées par le navigateur.

Ces balises sont très importantes car elles permettent un meilleur référencement du site auprès des différents moteurs de recherches (Google / Yahoo / etc..)

La balise <meta> dispose de plusieurs attributs :

- **author** : L'auteur du site
- **description** : Description du site
- **keywords** : Liste de mots décrivant le site
- **charset** : Définit l'encodage de la page
- **robots** : Indique au robot la façon dont il doit parcourir le site

```
<head>
  <title>Titre du site</title>
  <link rel="stylesheet" type="text/css" href="css/style.css">

  <meta name="author" content="Christopher Legrand" />
  <meta name="description" content="Ce site parle de développement web" />
  <meta name="keywords" content="html, css, balises" />
  <meta name="charset" content="utf-8" />
  <meta name="robots" content="index, follow" />
</head>
```

La balise <meta>

Comme mentionné précédemment, l'attribut **robots** joue un rôle capital dans le référencement de votre site puisque c'est lui qui va diriger le robot. Pour cela, il dispose de plusieurs valeurs :

- **all** : Valeur par défaut, indique au robot qu'il doit tout indexer
- **index** : indique au robot qu'il peut indexer la page.
- **noindex** : indique au robot qu'il ne faut pas indexer la page.
- **follow** : indique au robot qu'il peut suivre les liens dans la page.
- **nofollow** : indique au robot qu'il ne faut pas suivre les liens dans la page.

Mais il existe un autre moyen pour ne pas indexer le site web : le fichier **robots.txt**

Il s'agit d'une convention visant à empêcher les robots d'explorations d'accéder à une, plusieurs ou toutes les parties d'un site. Le fichier robots.txt sera à placer à la racine du site, et contiendra une liste de ressources qui ne doivent pas être explorées par les moteurs de recherches.

```
user-agent: *  
Disallow: /
```

Bloque totalement l'exploration du site

```
User-agent: *  
Disallow: /dossier1/  
Disallow: /page3.html/
```

Bloque partiellement l'exploration du site

Exercice n°1

Pour vous entrainer aux bases du HTML et bien comprendre l'importance de la balise <head>, je vous propose de créer un document HTML contenant un titre, une description, un auteur et des mots clés.

La balise <header>

Cette balise est l'introduction de votre page ou de votre section. Elle peut contenir des éléments de titre, mais aussi d'autres éléments tels qu'un logo, un formulaire de recherche, etc.

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <title>Cours HTML-CSS</title>
</head>

<body>
  <header>Le titre de ma page HTML</header>
</body>
</html>
```

La balise <article>

Cette balise représente du contenu autonome dans un document, une page, une application, ou un site. Elle permet de séparer le contenu général du site avec un bloc de contenu spécifique.

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <title>Cours HTML-CSS</title>
</head>

<body>
  <header>Le titre de ma page HTML</header>

  <article>
    <h1>Ceci est le titre de mon blog</h1>
    <p>Ceci est le contenu de mon blog</p>
  </article>
</body>
</html>
```

La balise <aside>

Cette balise représente une partie d'un document HTML dont le contenu n'a aucun rapport direct avec le contenu principal. <aside> est fréquemment présent sous la forme d'encadrés ou de boîtes de légende.

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <title>Cours HTML-CSS</title>
</head>

<body>
  <header>Le titre de ma page HTML</header>

  <article>
    <h1>Ceci est le titre de mon blog</h1>
    <p>Ceci est le contenu de mon blog</p>
  </article>

  <aside>
    <p>Ceci est un élément aparté de mon contenu</p>
  </aside>
</body>
</html>
```

La balise <nav>

Cette balise représente une section d'une page ayant des liens vers d'autres pages. Autrement dit, c'est une section destinée à la navigation dans un document (avec des menu par exemple).

```
<aside>
  <nav>
    <h1>Menu Principal</h1>
    <ul>
      <li><a href="index.html">Accueil</a></li>
      <li><a href="a-propos.html">A propos</a></li>
      <li><a href="contact.html">Contact</a></li>
    </ul>
  </nav>
</aside>
```


La balise <section>

Cette balise représente une section générique d'un document, par exemple un groupe de contenu thématique. Il faut abuser de cette balise afin d'améliorer la structure du code.

```
<section>
  <aside>
    <nav>
      <h1>Menu Principal</h1>
      <ul>
        <li><a href="index.html">Accueil</a></li>
        <li><a href="a-propos.html">A propos</a></li>
        <li><a href="contact.html">Contact</a></li>
      </ul>
    </nav>
  </aside>
</section>
```

La balise <footer>

Cette balise représente le pied de page de la section ou de la racine de sectionnement la plus proche. Un pied de page ou de section contient habituellement des informations sur l'auteur de la section, les données relatives au droit d'auteur (copyright) ou les liens vers d'autres documents en relation.

```
<section>
  <footer>
    <p>Cours par Christopher Legrand</p>
    <ul>
      <li><a href="https://facebook.com">Mon facebook</a></li>
      <li><a href="https://twitter.com">Mon Twitter</a></li>
    </ul>
  </footer>
</section>
```

La balise <body>

Vous l'aurez donc compris, l'élément <body> représente le contenu principal du document HTML. Il ne peut y avoir qu'un élément <body> par document.

Sans elle, rien ne s'affiche ! Toutes les balises que nous venons de voir doivent se trouver dans la balise <body>

Si vous avez eu la curiosité d'ouvrir votre fichier HTML dans votre navigateur web, vous avez pu remarquer qu'il ne prends pas en compte la mise en page de votre document HTML (Retour à la ligne, tabulations, etc..).

Cette mise en page se fait à travers les différentes balises HTML et leurs attributs

Les attributs

Chaque élément HTML peut avoir un ou plusieurs attributs. Ces attributs sont des valeurs supplémentaires qui permettent de configurer les éléments et d'adapter leur comportement.

Voici la liste des attributs globaux, les plus souvent utilisés :

ACCESKEY	Cet attribut permet de définir un raccourci clavier pour activer un élément ou lui passer le focus.	STYLE	Cet attribut définit des styles CSS qui auront la priorité sur ceux définis précédemment.
CLASS	Permet de définir un nom de classe afin de le ou de le mettre en forme avec CSS (et JS).	TABINDEX	Cet attribut permet de modifier l'ordre dans la navigation à la tabulation.
CONTENTEDITABLE	Cet attribut indique si le contenu de l'élément peut être édité.	TITLE	Cet attribut définit un texte expliquant le contenu de l'élément
DIR	Cet attribut définit la direction du texte (gauche à droite ou droite à gauche).	ID	Permet de définir un identifiant unique afin de le ou de le mettre en forme avec CSS (et JS).
LANG	Cet attribut définit la langue utilisée dans l'élément.	ROLE	Défini le rôle du widget (menu, case à cocher) pour les mal voyants ou aveugles (spécification ARIA)

Les commentaires

En HTML, comme dans tous les langages informatiques, vous pouvez et devez écrire des commentaires à l'intérieur de votre code. Le but d'un commentaire est d'indiquer que ce que fait telle ou telle balise, section etc..

Les commentaires ne seront pas affichés à l'écran pour votre utilisateur. En revanche, ils pourront les retrouver dans l'outil de développement de votre navigateur web. Il est donc important de n'y inscrire que des informations non sensibles, et relatives au contenu.

```
<!-- Ceci est le menu de mon site -->
<section>
  <aside>
    <nav>
      <h1>Menu Principal</h1>
      <ul>
        <li><a href="index.html">Accueil</a></li>
        <li><a href="a-propos.html">A propos</a></li>
        <li><a href="contact.html">Contact</a></li>
      </ul>
    </nav>
  </aside>
</section>
```

Les balises de mise en forme

L'élément HTML `
` crée un saut de ligne (un retour chariot) dans le texte. Il s'avère utile lorsque les sauts de ligne ont une importance.

```
texte sur <br> 2 lignes
```

L'élément HTML `` indique que le texte a une importance particulière ou un caractère sérieux voire un caractère urgent. Cela se traduit généralement par un affichage en gras. Attention, il ne doit pas être utilisé à des fins stylistique mais plutôt logique.

```
Un élément <strong>important</strong>
```

L'élément HTML `` (pour emphase) est utilisé afin de marquer un texte sur lequel on veut insister. Cependant, il ne doit pas être utilisé pour appliquer un style italique. Pour la mise en forme, on utilisera du CSS.

```
Je veux insister sur <em>cet élément</em>
```

L'élément HTML `<small>` permet de représenter des commentaires ou des textes à écrire en petits caractères (des termes d'un contrat, des mentions relatives au droit d'auteur, etc.) quelle que soit la présentation.

```
<small>Copyright Christopher Legrand</small>
```

Les balises de mise en forme

L'élément HTML `<hr>` représente un changement thématique entre des éléments de paragraphe. Elle trace un trait de séparation horizontal.

```
<hr>
```

L'élément HTML `<p>` représente un paragraphe de texte. Les paragraphes sont généralement représentés comme des blocs et séparés par un saut de ligne. Les paragraphes sont des éléments blocs.

```
<p>Ceci est le contenu de mon blog</p>
```

Les éléments `<h1>` à `<h6>` représentent six niveaux de titres dans un document, `<h1>` est le plus important et `<h6>` est le moins important. Un élément de titre décrit brièvement le sujet de la section qu'il introduit.

```
<h1>Titre H1</h1>  
<h2>Titre H2</h2>  
<h3>Titre H3</h3>  
<h4>Titre H4</h4>  
<h5>Titre H5</h5>  
<h6>Titre H6</h6>
```

Les balises de mise en forme

L'élément HTML `<div>` (qui signifie division du document) est un conteneur générique qui permet d'organiser le contenu sans représenter rien de particulier. Probablement la balise de découpage la plus importante de toutes.

Il peut être utilisé afin de grouper d'autres éléments pour leur appliquer un style (en utilisant les attributs `class` ou `id`) ou parce qu'ils partagent des attributs aux valeurs communes, tel que `lang`.

Il doit uniquement être utilisé lorsqu'aucun autre élément sémantique (par exemple `<article>` ou `<nav>`) n'est approprié.

Il existe énormément d'autres balises de mise en forme. Vous trouverez toutes les informations sur le site :

<https://www.w3schools.com/tags/>

Exercice n°2

Reproduire le document ci-dessous :

Je viens de créer ma première page HTML tout seul !

Lien 1

Lien 2

Lien 3

Mon premier titre

Et son sous-titre

Lorem ipsum *dolor sit amet*, consectetur adipisicing elit. Omnis provident ducimus quasi itaque pariatur autem quos eos nostrum, voluptatibus vero a, earum, odit suscipit nulla aperiam aliquam! Dolorem, hic, mollitia!

Mon deuxième titre

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Omnis provident ducimus quasi itaque pariatur autem quos eos nostrum, voluptatibus vero a, **earum, odit suscipit nulla aperiam aliquam!** Dolorem, hic, mollitia!

Copyright de entrez votre nom

Indenter son code

Vous avez probablement remarqué que j'ai régulièrement laissé une marge à gauche en début de ligne quand j'écrivais mon code. C'est ce qu'on appelle l'indentation

Vous remarquerez que lorsque l'on imbrique des balises dans d'autres balises, nous les indentons chaque fois un peu plus. Cela améliore grandement la lisibilité du code ! C'est un paramètre obligatoire que vous devrez appliquer systématiquement lorsque vous coderez. Un code non-indenté sera considéré comme un code de mauvaise qualité.

Pour indenter son code, il suffit de se positionner au début de la ligne et appuyer sur la touche Tab qui va générer 4 espaces.

```
<!DOCTYPE html>
<html lang="fr">
<head>
<title>Cours HTML-CSS</title>
</head>
<body>
<section>
<h1>Ma première page HTML</h1>
<p>Lorem ipsum dolor sit amet </p>
</section>
</body>
</html>
```

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <title>Cours HTML-CSS</title>
  </head>
  <body>
    <section>
      <h1>Ma première page HTML</h1>
      <p>Lorem ipsum dolor sit amet </p>
    </section>
  </body>
</html>
```

Exercice n°3

Reprenez le fichier HTML de l'exercice n°2 et vérifiez son indentation. Si elle n'est pas bonne, corrigez-la.

Les listes ordonnées

L'élément HTML `...` représente une liste ordonnée. Les éléments d'une telle liste sont généralement affichés avec un indicateur ordinal pouvant prendre la forme de nombres, de lettres, de chiffres romains ou de points.

A l'intérieur de cette balise `...` qui indique que nous allons avoir une liste ordonnée, il faut maintenant créer les éléments, et cela sera possible grâce aux balises `...`. Il est également possible d'imbriquer des listes dans des listes, mais uniquement dans des ``

```
<section>
  <h1>Ma première liste ordonnée</h1>
  <ol>
    <li>Element 1</li>
    <li>Element 2</li>
    <li>Element 3</li>
    <li>Element 4</li>
  </ol>
</section>
```

```
<section>
  <h1>Ma première liste ordonnée</h1>
  <ol>
    <li> Element 1
      <ol>
        <li>Sous-élément 1</li>
        <li>Sous-élément 2</li>
        <li>Sous-élément 3</li>
      </ol>
    </li>
    <li>Element 2</li>
    <li>Element 3</li>
    <li>Element 4</li>
  </ol>
</section>
```

Les listes ordonnées

L'élément HTML `...` possède différents attributs universels :

- Reversed
- Start
- Type (Déprécié... La propriété CSS `list-style-type` doit être utilisé à la place de cet attribut.)

```
<ol reversed="reversed">
  <li> Element 1</li>
  <li>Element 2</li>
  <li>Element 3</li>
  <li>Element 4</li>
</ol>
```

```
<ol start="2">
  <li> Element 1</li>
  <li>Element 2</li>
  <li>Element 3</li>
  <li>Element 4</li>
</ol>
```

```
<ol type="A">
  <li> Element 1</li>
  <li>Element 2</li>
  <li>Element 3</li>
  <li>Element 4</li>
</ol>
```

Les listes non-ordonnées

L'élément HTML `...` fonctionne exactement de la même manière que la balise `...`. La seule différence est que la liste devient non-ordonnées, c'est à dire qu'on remplace les chiffres par des puces.

```
<section>
  <h1>Ma première liste non-ordonnée</h1>
  <ul>
    <li>Element 1</li>
    <li>Element 2</li>
    <li>Element 3</li>
    <li>Element 4</li>
  </ul>
</section>
```

Exercice n°4

Reproduire ce document :

Mon exercice sur les listes

1. Elément 1
2. Elément 2
3. Elément 3
 - sous-élément 1
 - sous-élément 2
 - D. sous-élément D
 - E. sous-élément E
 - sous-élément 3
4. Elément 4

Les images

L'élément HTML permet de représenter une image dans un document. C'est une balise spéciale car elle n'a pas de balise de fermeture. Il a juste un balise d'ouverture.

Elle possède un attribut obligatoire **src** (source) qui indique le chemin relatif ou absolu de l'image.

Je vais vous inviter à télécharger l'image *chat.jpg* qui se trouve dans le drive d'équipe, dans le dossier HTML.

```

```

ALT=""

L'attribut alt d'une image indique à Google ce que contient l'image.

HEIGHT=""

Définit la hauteur de l'image, en pixel ou en pourcentage

WIDTH=""

Définit la largeur de l'image, en pixel ou en pourcentage

ID=""

Donne un nom à l'image que l'on pourra utiliser avec des langages informatiques (JS/PHP)

TITLE=""

Texte qui apparaît pendant le chargement de l'image, puis en info-bulle une fois chargée.

SRC=""

Indique l'adresse de l'image à afficher

Les images

Il faut bien faire la distinction entre un chemin relatif et un chemin absolu. Pour vous illustrer cela, je vais créer un dossier **image** et je vais transférer l'image du chat à l'intérieur. Si je fais ça et que je ne touche pas au code, le navigateur ne trouvera pas mon image.

Notre point de référence, c'est notre index.html. Il va donc falloir indiquer le chemin du dossier image, puis du fichier chat.jpg par rapport au fichier index.html. Ça c'est ce qu'on appelle un chemin relatif. Parce qu'il est relatif au fichier index.html. On s'appuie sur index.html pour lui indiquer le chemin du dossier image.

```

```

On a également le chemin absolu. Pour celui ci, il faut d'abord initialiser l'attribut src par un slash "/" précisant que l'on ne se réfère plus à l'emplacement courant, ici index.html, mais que l'on remette directement à la racine pour ensuite préciser le chemin complet vers notre source.

```

```

Exercice n°5

Reprenez l'exercice 2 et intégrer l'image du chat en modifiant sa largeur, sa hauteur, en lui donnant un titre et un texte de description

<figure> & <figcaption>

L'élément HTML <figure> représente une figure (un schéma), qui peut être accompagné d'une légende grâce à l'élément <figcaption>. Cette balise est utilisée afin d'affiner la découpe du code, mais reste malgré tout très peu utilisée.

L'élément HTML <figcaption> (pour figure caption en anglais) permet d'indiquer un sous-titre, une légende, associé à une figure ou à une illustration (cette dernière étant représentée par l'élément <figure> qui est le parent direct de la légende). L'élément <figcaption> est optionnel ; s'il n'est pas présent, la figure n'aura pas de légende.

```
<figure>  
    
  <figcaption>Un chat</figcaption>  
</figure>
```

La balise <iframe>...</iframe>

L'élément HTML <iframe> représente un contexte de navigation imbriqué qui permet en fait d'obtenir une page HTML intégrée dans la page courante.

En gros, la balise <iframe> est utilisé pour incorporer un autre document dans le notre. On peut incorporer une Maps, une vidéo Youtube, etc.

La balise <iframe> dispose d'un très grand nombre d'attributs. Le plus important est le **allow** qui permet de définir une politique de fonctionnalité pour l'iframe.

```
<iframe width="560" height="315" src="https://www.youtube.com/embed/YNcBfSb3o-s" frameborder="0"
allow="accelerometer; autoplay; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>
```

Les balises <video> & <audio>

L'élément HTML <video> permet d'intégrer un contenu vidéo à votre site. Néanmoins, j'attire tout de suite votre attention sur le fait qu'on préférera uploader notre vidéo sur YouTube ou Vimeo ou n'importe quel site de streaming, puis intégrer l'iframe plutôt que d'ajouter la vidéo directement à notre site, qui va l'alourdir de manière très très conséquente.

L'élément HTML <audio> est utilisé afin d'intégrer un contenu sonore à votre site. Il peut contenir une ou plusieurs sources audio représentées avec l'attribut src ou l'élément <source>. On peut également y ajouter les contrôles de lecture au besoin via l'attribut controls, et définir pleins d'autres fonctionnalités.

La balise <a>...

L'élément <a> (pour ancre ou anchor en anglais) définit un un lien hypertexte (ou hyperlien) vers un autre endroit de la même page (ancre) ou vers une autre page sur le Web. Il est composé de 2 choses :

- 1) Le texte qui va décrire notre lien
- 2) L'adresse sur laquelle va nous redirigé ce lien

```
<a href="https://christopherlegrand.fr">Mon site</a>
```

Il s'agit d'un lien hypertexte simple auquel on peut donner les attributs suivant :

href=""

Indique le chemin du lien à suivre

```
<a href="https://christopherlegrand.fr">Mon site</a>
```

hreflang=""

Indique la langue de la page vers laquelle le lien pointe

```
<a href="https://christopherlegrand.fr" hreflang="fr">Mon site</a>
```

target=""

Indique où le lien va s'ouvrir (nouvel onglet etc..)

```
<a href="https://christopherlegrand.fr" target="_blank">Mon site</a>
```

La balise <a>...

Nous avons pour le moment défini une simple phrase en tant que lien. Il est également possible de remplacer cette phrase par une image.

Avant de vous montrer comment faire, je vous invite à prendre quelques minutes pour chercher par vous même la solution et remplacer "Mon site" par l'image du chat.

```
<a href="https://christopherlegrand.fr" target="_blank">  
    
</a>
```

La balise <a>...

Il est possible de réaliser des liens vers différents type de fichier.

Lien vers une image :

```
<a href="chat.jpg" target="_blank">  
    
</a>
```

Lien vers un son :

```
<a href="son.wav" >Ecouter le son : </a>
```

Lien pour envoyer un email

```
<a href="mailto:contact@christopherlegrand.fr">Contactez-moi</a>
```


Lien hypertexte vers une ancre

Le fonctionnement classique des liens hypertexte est de pointer vers une page différente de celle en cours de consultation, pour naviguer dans le site.

Il est également possible de créer un lien vers un endroit précis de la page courante, ou d'une autre page afin de positionner correctement le navigateur grâce à ce qu'on appelle une **ancre**.

Créer une ancre est très simple, il suffit d'attribuer un ID à l'élément vers lequel on veut pointer et d'y associer un lien débutant par le caractère # suivi du nom de l'ID

```
<header>
  <nav>
    <ul>
      <a href="#accueil"><li>Accueil</li></a>
      <a href="#aPropos"><li>A propos</li></a>
      <a href="#contact"><li>Contact</li></a>
    </ul>
  </nav>
</header>
```

```
<section id="accueil">
  <h1>Accueil</h1>
  <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Aliquam exercitationem, voluptatibus veritatis! Dolor delectus nam debitis obcaecati veritatis impedit dolor consequuntur provident.</p>
</section>

<section id="aPropos">
  <h1>A propos</h1>
  <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Aliquam exercitationem, voluptatibus veritatis! Dolor delectus nam debitis obcaecati veritatis impedit dolor consequuntur provident.</p>
</section>

<section id="contact">
  <h1>Contact</h1>
  <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Aliquam exercitationem, voluptatibus veritatis! Dolor delectus nam debitis obcaecati veritatis impedit dolor consequuntur provident.</p>
</section>
```

Exercice n°6

Reprenez l'exercice 2 et modifier le menu afin de créer une navigation avec ancres entre les 2 articles. Ajouter également des liens hypertextes vers l'extérieur et vers une image.

La balise <table>...</table>

L'élément HTML <table> permet de représenter un tableau, c'est-à-dire des informations exprimées sous formes de lignes et de colonnes, cellules, etc..

Les tableaux HTML ne doivent être utilisés que pour des données tabulaires, c'est pour cela qu'ils sont conçus. Il ne faut surtout pas les utiliser pour organiser des pages Web. La disposition d'une page est prise en charge par le CSS. Petite exception pour l'intégration d'email qui fonctionne uniquement avec des tableaux pour la mise en page.

Le contenu de chaque tableau est encadré par les balises <table>...</table>. Le plus petit conteneur d'un tableau est la cellule, elle est créée avec l'élément <td>...</td> (tableau données). Si nous voulons une colonne de quatre cellules, nous devons créer 4 balises <td>...</td>

```
<table>
  <td>Bonjour, je suis votre première cellule</td>
  <td>je suis votre deuxième cellule</td>
  <td>je suis votre troisième cellule</td>
  <td>je suis votre quatrième cellule</td>
</table>
```

La balise <table>...</table>

Comme vous pouvez le voir, les cellules ne sont pas placées les unes en dessous des autres, mais sur une même ligne.

Pour empêcher cette ligne de croître et commencer à placer les cellules sur une deuxième ligne, nous allons devoir utiliser la balise <tr>...</tr> (table rangée). Nous pouvons créer autant de rangées que nous le souhaitons, en fonction du contenu du tableau.

```
<table>
  <tr>
    <td>A</td>
    <td>B</td>
    <td>C</td>
    <td>D</td>
  </tr>
  <tr>
    <td>E</td>
    <td>F</td>
  </tr>
  <tr>
    <td>G</td>
    <td>H</td>
  </tr>
</table>
```

La balise <table>...</table>

Il est également possible de définir des en-têtes de colonne. Pour cela, il suffit simplement de remplacer le <td> par un <th>

```
<table border="1">
  <tr>
    <th>Colonne 1</th>
    <th>Colonne 2</th>
    <th>Colonne 3</th>
    <th>Colonne 4</th>
  </tr>
  <tr>
    <td>A</td>
    <td>B</td>
    <td>C</td>
    <td>D</td>
  </tr>
  <tr>
    <td>E</td>
    <td>F</td>
    <td colspan="2" rowspan="2">
  </tr>
  <tr>
    <td>G</td>
    <td>H</td>
  </tr>
</table>
```

Colonne 1	Colonne 2	Colonne 3	Colonne 4
A	B	C	D
E	F		
G	H		

La balise <table>...</table>

Il existe plusieurs attributs pour les tableaux, mais je tiens à vous présenter les 2 suivants :

colspan=""

Fusionnes des cellules d'une rangée

rowspan=""

Fusionne des cellules d'une colonne

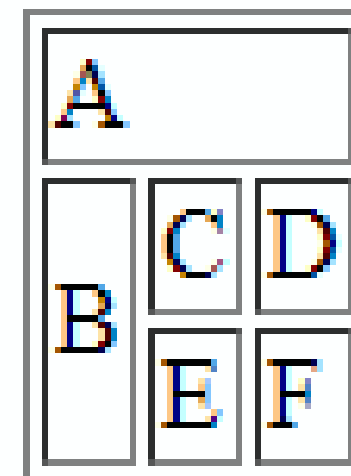
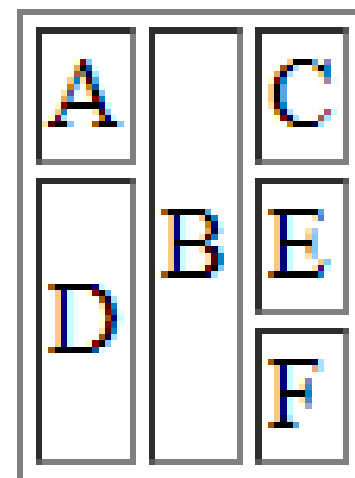
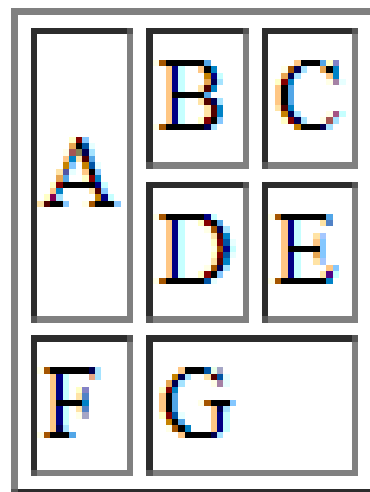
La cellule va occuper la place du nombre de cellules défini dans l'attribut

Colonne 1	Colonne 2	Colonne 3	Colonne 4
A	B	C	D
E		F	
G	H		

```
<table border="1">
  <tr>
    <th>Colonne 1</th>
    <th>Colonne 2</th>
    <th>Colonne 3</th>
    <th>Colonne 4</th>
  </tr>
  <tr>
    <td>A</td>
    <td>B</td>
    <td>C</td>
    <td rowspan="3">D</td>
  </tr>
  <tr>
    <td colspan="2">E</td>
    <td>F</td>
  </tr>
  <tr>
    <td>G</td>
    <td>H</td>
  </tr>
</table>
```

Exercice n°7

Reproduire les figures suivantes :



La balise <thead> ... </thead>

Nous connaissons maintenant le principe des cellules et des rangées, mais allons encore un peu plus loin dans la conception de nos tableaux.

Nous avons besoin d'une zone d'en-tête pour le tableau. Précédemment, nous avons vu que nous pouvions utiliser les balises <th> pour indiquer que la cellule sera un titre. Avant cela, il faut indiquer qu'il s'agit d'une zone d'en-tête grâce à la balise <thead>...</thead>.

A l'intérieur, on indique qu'on veut une ligne de cellules de titre en faisant <tr>...</tr>, puis on donne le nom de la cellule dans les balises <th>...</th>.

```
<thead>
  <tr>
    <th>Colonne n°1</th>
    <th>Colonne n°2</th>
    <th>Colonne n°3</th>
  </tr>
</thead>
```


La balise <tbody> ... </tbody>

Maintenant que nous avons défini les titres de nos colonnes, nous pouvons ajouter nos données en commençant par la balise <tbody> juste en dessous de notre </thead>.

Vous aurez donc compris qu'on utilise les <th> dans le <thead> puis les <td> dans le <tbody>

```
<thead>
  <tr>
    <th>Colonne n°1</th>
    <th>Colonne n°2</th>
    <th>Colonne n°3</th>
  </tr>
</thead>
```

Exercice n°8

Vous connaissez maintenant les bases du HTML, de la mise en forme, des tableaux etc..
Vous allez réaliser un projet plus conséquent que les exercices précédents, puisque vous allez devoir réaliser une page Web.

Voici le sujet :

Vous êtes un grand passionné de cuisine, et vous souhaitez mettre vos recettes à disposition sur un site web.
Votre site devra dans cet ordre contenir :

- Un sommaire des recettes (3 recettes minimum)
- L'image du plat
- Une liste d'informations contenant : "Temps de cuisson" et "Temps de préparation"
- Un tableau contenant les ingrédients et la quantité nécessaire de chaque ingrédient
- Pour finir, une liste des différentes étapes de la recette à respecter

Vous devrez réaliser 3 recettes, n'hésitez pas à vous aider des sites de cuisine comme Marmiton pour récupérer tous les éléments

Les formulaires

Nous allons maintenant voir le dernier élément pour le cours sur HTML5, les formulaires.

Les formulaires sont très importants car ils permettent de capter les informations de nos utilisateurs comme par exemple avec un formulaire de contact.

Avant de commencer, je tiens à vous préciser qu'un formulaire fonctionne avec 2 parties :

- Le front-end
- Le back-end

Nous sommes pour le moment concentré sur la partie Front-End, nous ne rentrerons pas tout de suite dans les détails du Back-End, cela se passera dans un autre cours.

Les formulaires

Pour écrire un formulaire, il faut utiliser la balise `<form>...</form>`. C'est le point de départ de notre formulaire. Tout comme pour les tableaux, nous allons venir imbriquer d'autres balises à l'intérieur.

Le tag le plus utilisé dans un formulaire est le tag `<input>`. Un input est tout simplement un champ, une zone à laquelle on attend une information communiquée par le client. Cet input peut avoir plusieurs formes. (Boutons / Texte / Checkbox etc..)

Nous allons dans un premier faire un input de type texte en lui donnant l'attribut `type=""` :

```
<form>  
  <input type="text" name="nom">  
</form>
```

Une petite précision, l'attribue `name=""` sur un input est obligatoire. C'est ce qui va permettre au serveur d'identifier le champ et de spécifier l'information rentrées par l'utilisateur pour qu'il soit capable de faire ce qu'il doit en faire.

Nous avons maintenant un champ de type texte dans lequel on peut écrire, mais rien ne se passera puisque le back-end n'existe pas, il n'y aucune logique attribué à ce champ.

Les formulaires

Mon champ est maintenant créé, il est possible de lui spécifier un titre, ce qu'on appelle un label.

Pour cela, il suffit de créer une balise `<label>...</label>` et lui donner le nom du champ. Mais il faut également utiliser l'attribut `for=""` pour spécifier à quel input est rattaché ce label. Pour cela, il va donc falloir donner un id à notre input

```
<form>
  <label for="nom">Nom</label>
  <input id="nom" type="text" name="nom">
</form>
```

Un autre attribut très sympa sur les inputs, c'est le placeholder. C'est un attribut qui permet de spécifier à l'utilisateur ce qui est attendu de lui. Par exemple, on peut écrire "Entrez votre nom..."

```
<form>
  <label for="nom">Nom:</label>
  <input id="nom" type="text" name="nom" placeholder="Entrez votre nom...">
</form>
```

Les formulaires

Une fois que j'ai créé mes champs, il est possible de dédier une zone de texte au client pour qu'il puisse nous adresser un message. Pour cela, on utilise la balise `<textarea>`.

```
<form>
  <label for="nom">Nom:</label>
  <input id="nom" type="text" name="nom" placeholder="Entrez votre nom...">

  <label for="message">Message:</label>
  <textarea id="message"></textarea>
</form>
```

Il existe 2 attributs qui permettent de modifier la longueur et la largeur de cette zone de texte. Il s'agit de :

- `cols=""`
- `rows=""`

```
<form>
  <label for="nom">Nom:</label>
  <input id="nom" type="text" name="nom" placeholder="Entrez votre nom...">

  <label for="message">Message:</label>
  <textarea id="message" cols="50" rows="20"></textarea>
</form>
```

Les formulaires

Vous connaissez maintenant les 2 balises principales d'un formulaire. Les saisies possibles et le comportement de l'élément `<input>` dépend fortement de la valeur indiquée dans son attribut `type`.
La balise `input` a énormément d'attributs, je ne vous listerai que les principaux :

`autocomplete`

Une chaîne de caractères qui indique le type d'autocomplétion à utiliser.

`autofocus`

Un attribut booléen qui passe le focus sur le champ lorsque le formulaire est affiché.

`disabled`

Un attribut booléen qui indique si le champ doit être désactivé.

`form`

L'ID du formulaire auquel le champ est rattaché.

`list`

L'ID d'un élément `<datalist>` qui fournit une liste de suggestions.

`name`

Le nom du champ qui sera rattaché à la donnée envoyée via le formulaire.

`readonly`

Un attribut booléen qui indique si le champ peut être édité ou non.

`required`

Un attribut booléen qui indique que le champ doit être renseigné avant de pouvoir envoyer le formulaire.

`tabindex`

Une valeur numérique qui indique à l'agent utilisateur l'ordre selon lequel naviguer au clavier grâce à la touche Tab.

`type`

Une chaîne de caractère qui indique le type de champ représenté par l'élément `<input>`.

`value`

La valeur du champ.

Focus sur l'attribut type="..."

L'attribut type="" dispose d'énormément de possibilité, qu'il faut connaître pour la création de formulaire en tout genre.

type="button"

Permet l'affichage d'un champ de type bouton.
Doit être suivi de l'attribut value=""

type="checkbox"

Permet l'affichage d'une checkbox

type="password"

Permet l'affichage d'un champ de type password. Le texte devient caché

type="radio"

Permet l'affichage d'un bouton radio

type="submit"

Permet l'affichage d'un bouton submit.
Ex : Envoyer le formulaire

type="color"

Permet de choisir une couleur

type="date"

Permet l'affichage d'un calendrier pour choisir une date

type="email"

Permet l'affichage d'un champ de type e-mail

type="range"

Permet à l'utilisateur d'indiquer une valeur numérique comprise en 2 bornes

type="text"

Permet l'affichage d'un champ de type text

La balise <datalist>

L'élément HTML <datalist> contient un ensemble d'éléments <option> qui représentent les valeurs possibles pour d'autres contrôles.

```
<label for="FFDatalist">Choisissez un héros:</label>
  <input list="heros" id="FFDatalist" name="heros" />
  <datalist id="heros">
    <option value="Cloud">
    <option value="Sephiroth">
    <option value="Djidane">
    <option value="Squall">
    <option value="Capitaine Onion">
  </datalist>
```

La balise <select>

L'élément HTML <select> représente un contrôle qui fournit une liste d'options parmi lesquelles l'utilisateur pourra choisir.

```
<label for="FFSelectOption">Choisissez un héros:</label>
  <select name="FFheros" id="FFSelectOption">
    <option value="">--Choisissez un héros:--</option>
    <option value="cloud">Cloud</option>
    <option value="sephiroth">Sephiroth</option>
    <option value="djidane">Djidane</option>
    <option value="squall">Squall</option>
    <option value="onion">Captain Onion</option>
  </select>
```

La balise <optgroup>

L'élément HTML <optgroup>, utilisé dans un formulaire, permet de créer un groupe d'options parmi lesquelles on peut choisir dans un élément <select>.

```
<label for="FFHeros">Choisissez un heros:</label>
<select id="FFHeros">
  <optgroup label="Final Fantasy VII">
    <option value="sephiroth">Sephiroth</option>
    <option value="tifa">Tifa</option>
    <option value="aeris">Aeris</option>
    <option value="cloud">Cloud</option>
  </optgroup>
  <optgroup label="Final Fantasy VIII">
    <option value="squall">Squall</option>
    <option value="linoa">Linoa</option>
    <option value="selphie">Selphie</option>
    <option value="seifer">Seifer</option>
  </optgroup>
</select>
```

Exercice n°9

Vous travaillez pour une agence de voyage. Celle-ci vous a délégué l'intégration d'un formulaire de contact.

Vous devez donc créer un formulaire comprenant les informations suivantes :

- Nom / Prénom
- E-mail
- La possibilité de choisir son pays (6 pays suffiront)
- Une zone de texte pour que le client puisse rédiger son message
- Un bouton "Envoyé"