

Cours HTML 5

HTML

1. Introduction

1.1. Le Web ?

- Terme créé en 1990 par Tim Berners-Lee.
- Premier logo du WWW dessiné par Robert Caillau.

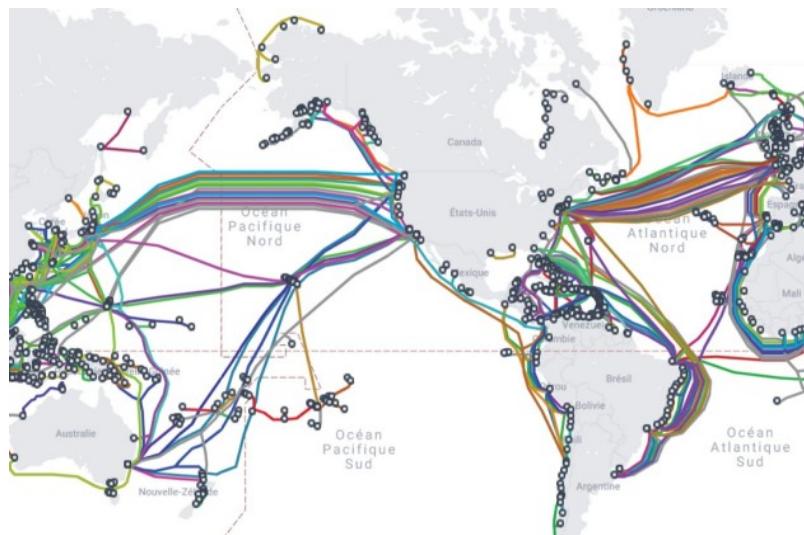


- Inventeur d'Internet et du premier navigateur (WorldWideWeb).
- WWW = World Wide Web = Toile mondiale.
- Technologie informatique permettant de consulter grâce à un navigateur des pages regroupées sur des sites en passant par le réseau internet.
- C'est une application d'Internet, comme l'email, la messagerie instantanée, le P2P, le partage de fichiers en passant par le réseau internet.
- Le web 2.0 / 3.0 etc ... = nouvelles versions successives du web.
- Fait de l'internaute une personne active sur la toile.
- Un web plus simple pour l'internaute, plus collaboratif et participatif.

1.2. Internet ?

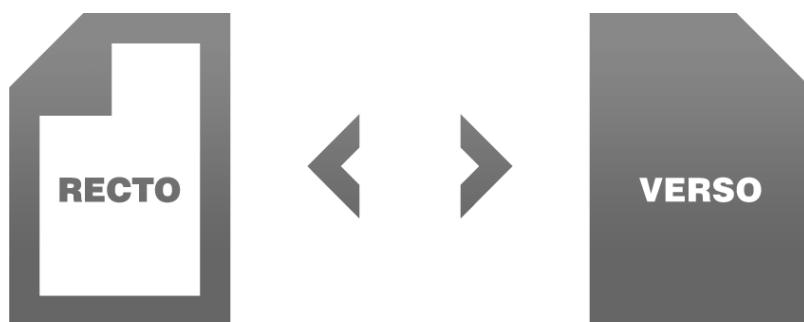
- Fin 1960 (guerre froide), les autorités militaires américaines mesurent la vulnérabilité de leur système de communication.
- Le département américain de la Défense travaille à la conception d'un système de réseau fiable.
- Première étape de la création d'Internet: ARPAnet.
- 1972, Ray Tomlinson (MIT) : courrier électronique.
- La même année : Première démonstration publique d'ARPAnet à l'International Computer Communication Conference (ICCC).

- interconnecte les réseaux informatiques du monde entier.
- connecte les périphériques à l'aide de fils de cuivre, de réseaux sans fil ou de câbles à fibres optiques.



1.3. Page Web ?

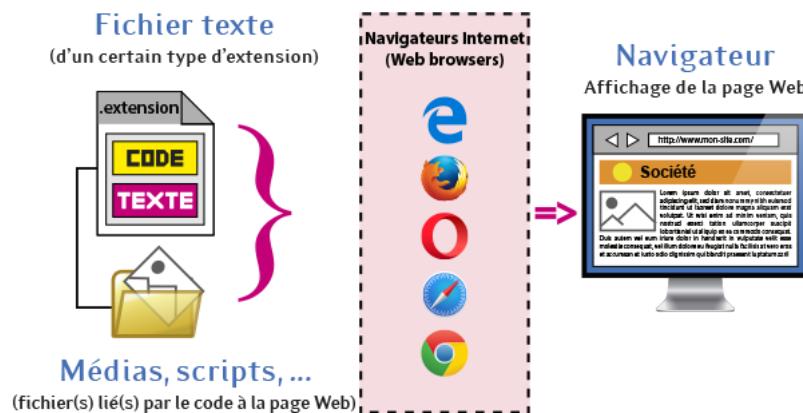
1.3.1. Concept



- recto = texte (lecture facile pour un être humain)
- verso = code HTML (lecture difficile pour un novice)

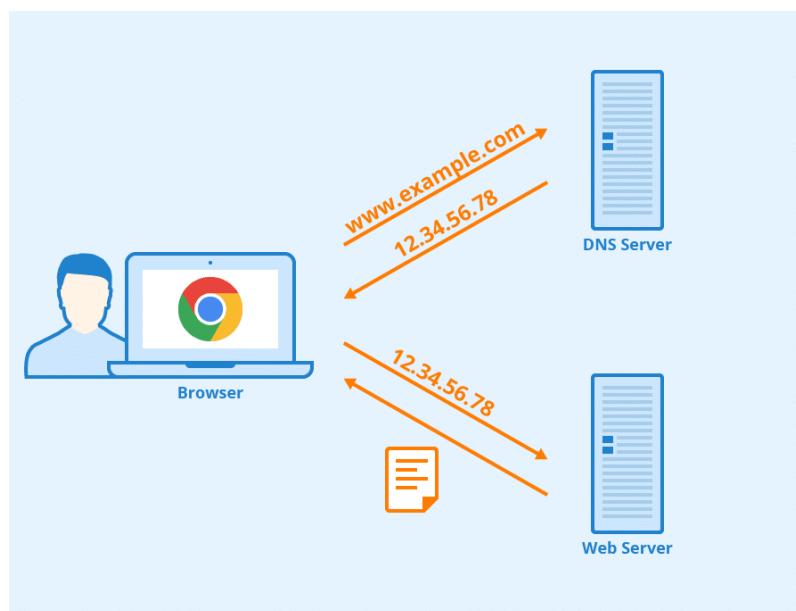


Le navigateur joue le rôle d'interprète ; il reçoit les instructions contenues sur le recto de la page, puis restitue ces informations sur le verso, mais cette fois de manière visuelle et compréhensible par l'être humain.



1.3.2. DNS

source : <https://www.phonandroid.com/meilleurs-serveurs-dns-gratuits.html>



DNS est l'acronyme de “Domain Name System”, un terme que l'on peut traduire par “système de noms de domaine”. Les interactions qu'il gère sont primordiales dans le fonctionnement d'internet, puisque le DNS traduit les noms de domaines en adresses IP. Quand vous tapez **google.com** dans votre barre d'URL, cela ne signifie pas grand-chose pour le réseau. Car la véritable identification de cette adresse est en fait **216.58.198.206**.

1.3.3. Anatomie d'une page web

1.3.3.1. Landing page

Une landing page, ou page de renvoi, désigne une page internet dont le but est de faciliter une action spécifique.

On parle de «page de renvoi» parce que le visiteur y est renvoyé dans le cadre du parcours de vente, que ce soit via un lien dans un email, un article de blog, ou encore une publicité.



1 - Titre de la page

2 - Sous-titre clair et concis

- Étant une des premières choses qu'un visiteur va lire, le titre d'une page de destination ne doit pas être confusant ou ennuyeux mais au contraire inciter le visiteur à regarder de plus près
- Le sous-titre doit donner un point précis qui est lié à la teneur du site, afin d'attirer l'attention d'un lecteur

3 - Rédaction parfaite

- Pour ne pas risquer d'entamer la confiance du visiteur, la rédaction doit être parfaite sans faute d'orthographe ou de grammaire

4 - Indicateur de confiance

- Ne pas hésiter à insérer des témoignages de clients, des articles de presse qui parlent de votre site, des prix concours etc... Ces éléments sont pour les visiteurs des gages de confiance

5 - Appel à l'action

- Une fois que le visiteur a lu le titre de la page d'atterrissement, il est essentiel qu'il sache quoi faire
- C'est le « do it now » : « Faites-le maintenant »

6 - L'appel au clic

- Identifier les mots clés qui sont utilisés par vos clients ou visiteurs (cliquer ici, acheter maintenant, gratuit, télécharger etc...)
- Un bouton de conversion doit se démarquer et être placé juste après l'appel à l'action ou être lui-même un bouton d'appel à l'action. Dans tous les cas, pour être visible il doit être gros et de couleur flashy

7 - Modérer le nombre de liens

- Les liens reliant l'utilisateur à un trop grand nombre de sites ou de pages risquent de le distraire et avoir un impact négatif sur les conversions
- Beaucoup de liens peuvent donner un sens sur une page d'accueil. En revanche pour une page de destination, la simplicité est la clé de réussite

8 - Utiliser des images et des vidéos

- Rajouter des vidéos de présentations produits, de témoignages de clients ainsi que des images de produits sur une page d'accueil
- Cela peut avoir un impact positif sur le visiteur ou pousser un acheteur à en savoir plus sur le produit présenté

9 - Restez au dessus de la ligne de flottaison

- La ligne de flottaison ou fold en anglais représente la ligne imaginaire à partir de laquelle l'utilisateur doit faire défiler la page. Le contenu se situant en dessous de la ligne de flottaison ne sera visible que si l'internaute fait défiler la page
- Il est donc important que tous les messages importants comme les appels au clic figurent au dessus de la ligne de flottaison
- Ne jamais placer un bouton à un endroit qui ne serait pas immédiatement visible par le visiteur

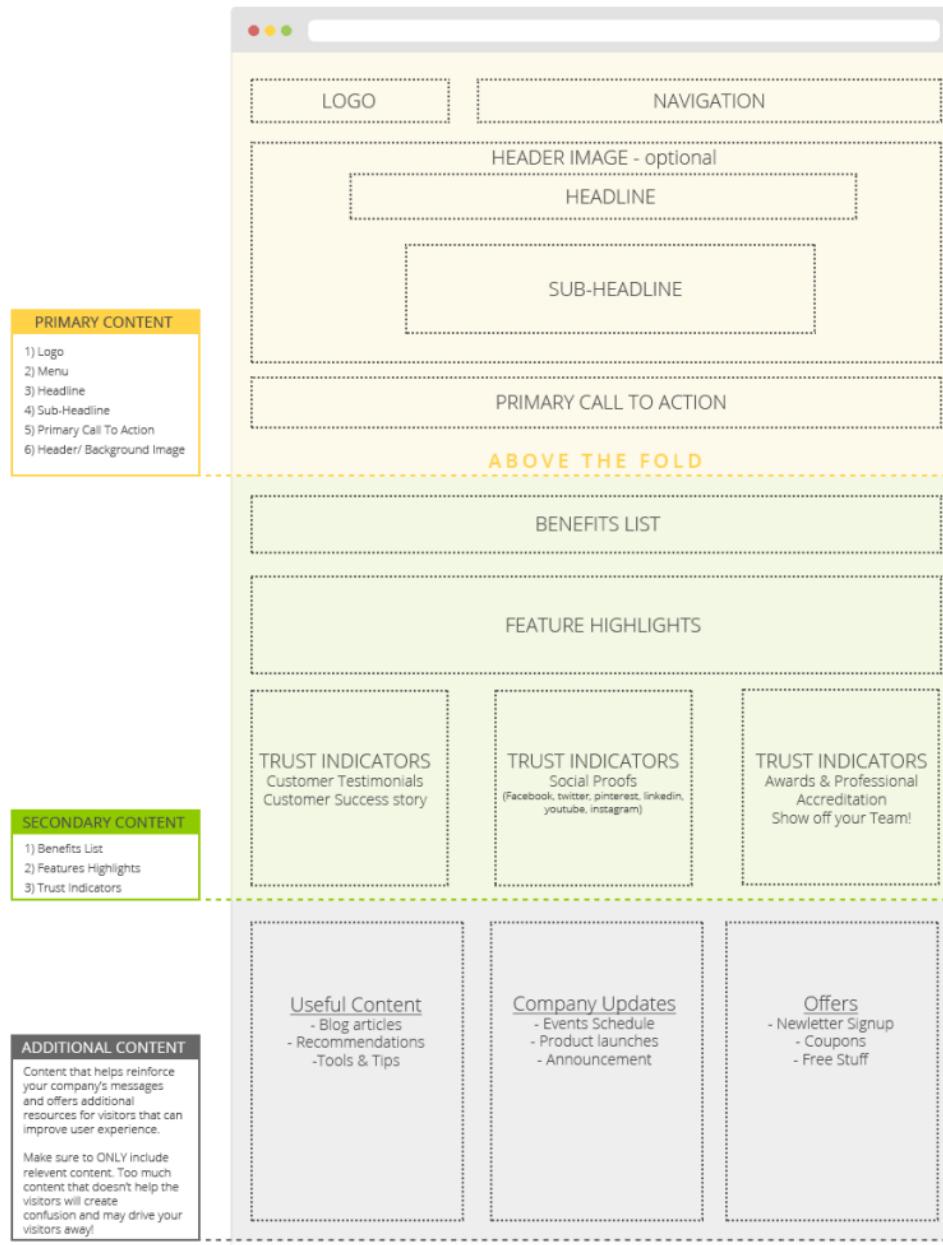
10 - Faire des essais

- Optimiser une page de destination pour améliorer le taux de conversion au fil du temps
- Pour cela, faites des tests avec différentes images, appel à l'action, design du site et mesurez ce qui a le plus d'impact sur vos visiteurs.

1.3.3.2. Page d'accueil

Les pages d'accueil sont généralement faites pour permettre plusieurs types d'interaction.

Un visiteur pourra la consulter pour acheter un produit, pour obtenir des informations sur une entreprise, pour trouver ses coordonnées ou pour postuler à un emploi. La page d'accueil doit pouvoir répondre à tous ces besoins.



1.4. Objectifs d'une page web

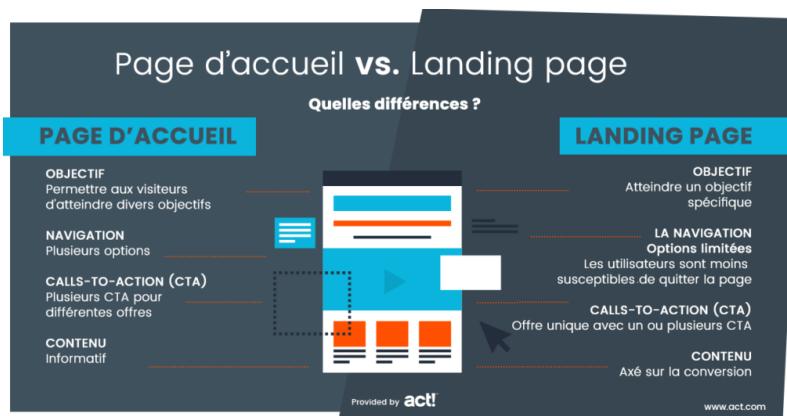
1.4.1. Objectifs utilisateurs

Les éléments qui sont sur une page d'accueil sont présents en général pour cultiver une bonne expérience utilisateur ainsi qu'une espèce de guide pour atteindre les objectifs que les utilisateurs peuvent s'être fixés. Votre but principal sera de procurer une route claire et facile pour les produits que les utilisateurs seront plus enclins à acheter.

1.4.2. Objectifs de l'entreprise

En plus de faciliter l'atteinte des objectifs de vos utilisateurs, votre **page d'accueil** doit également pouvoir satisfaire l'atteinte des objectifs de votre entreprise.

1.4.3. Landing Page vs Page accueil

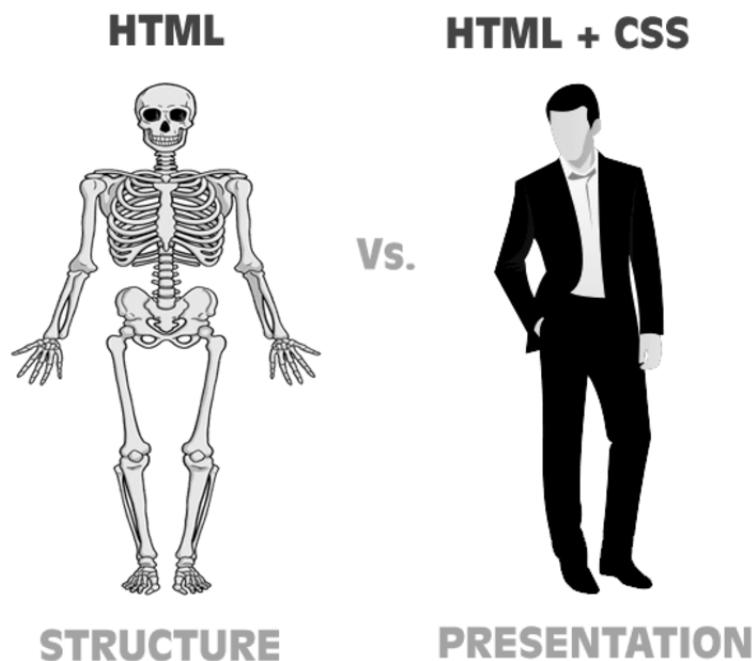


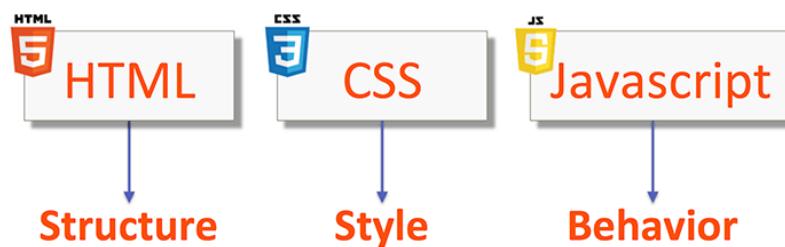
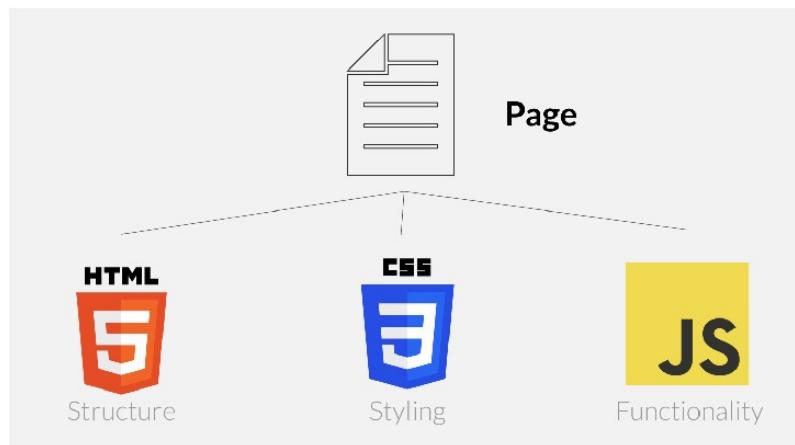
Lorsque les gens visitent votre site internet, savent-ils immédiatement qui vous êtes et ce que vous faites? Une bonne page d'accueil d'un site internet d'entreprise est capable de rapidement et efficacement communiquer ces informations et la raison d'être de l'entreprise.

	PAGE ACCUEIL	LANDING PAGE
Objectif	Permettre plusieurs types d'interaction. Consulter pour acheter un produit. Obtenir des informations sur une entreprise. Trouver ses coordonnées. Postuler à un emploi.	Collecter des informations sur leurs visiteurs ou de clôturer une vente. Télécharger un eBook. S'inscrire à un webinaire. Demander la démo gratuite d'un produit. Faire un achat.
Navigation	Inclut un ou plusieurs menus qui permettent aux visiteurs de se diriger vers les informations qu'ils recherchent.	Se passe de menu pour ne pas distraire le visiteur ni l'éloigner de son contenu. Elle ne propose qu'un seul type de navigation : le call-to-action .
Contenu	Contenu informatif, qui vise notamment à informer le visiteur sur la société et ses activités.	On se focalise plutôt sur la conversion du visiteur : Une page de renvoi consacrée à un eBook communiquera l'intérêt de télécharger le document, alors qu'une page de renvoi conçue pour faire la promotion d'un logiciel encouragera le visiteur à s'inscrire à une démo gratuite.
Calls-to-action (CTA)	Encourager le visiteur à : découvrir les produits. s'abonner à une newsletter. accéder à une démo gratuite.	Un seul type de CTA (parfois répété sous des formes différentes) afin de ne pas distraire de l'objectif ultime de la page.

1.5. Les languages Front-End

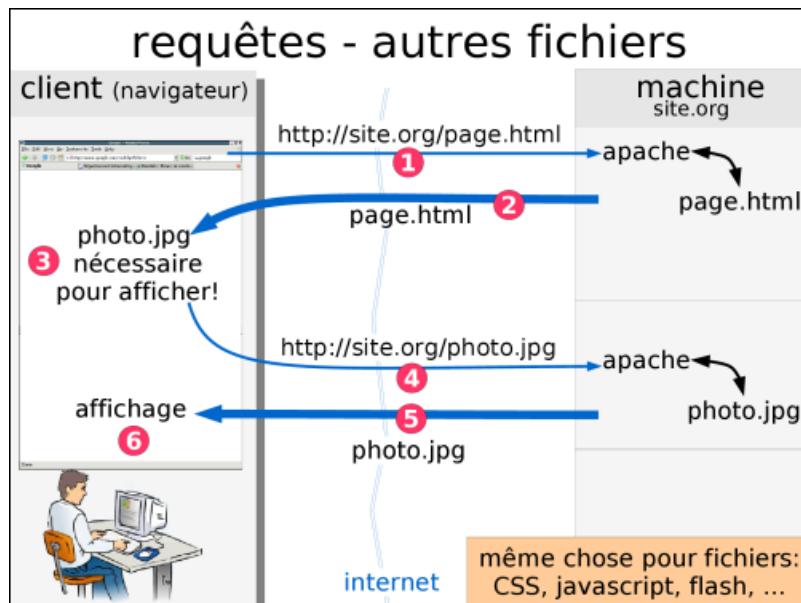
- HTML = langage de balisage
- CSS = langage de style
- JS = langage d'interaction





1.6. Affichage d'une page html en 7 étapes

1. Le navigateur envoie une requête http au serveur google.fr
2. Le serveur fabrique le code HTML de la page demandée et le compresse.
3. Le serveur envoie le code HTML comme réponse à la requête du navigateur.
4. Le navigateur reçoit en réponse le code HTML et le décomprime.
5. Le navigateur interprète le code HTML ligne par ligne & initie affichage de la page.
6. Le navigateur envoie de nouvelles requêtes vers le serveur (images, styles, scripts).
7. Le rendu de la page est terminé.



1.7. Créer une page web

* Editeur WYSIWYG (What You See Is What You Get) 🧑

* CMS (Content Management Service WORDPRESS) 🧑

* Coder 😊

* Payer un développeur 😱

1.8. Publier son site

- Trouver un nom de domaine.
- Trouver un hébergeur.
- Transférer son site en ligne.

1.9. Outils utilisés dans cette formation

Visual Studio Code + extensions :

- live html preview
- prettier
- live server

Navigateur : chrome

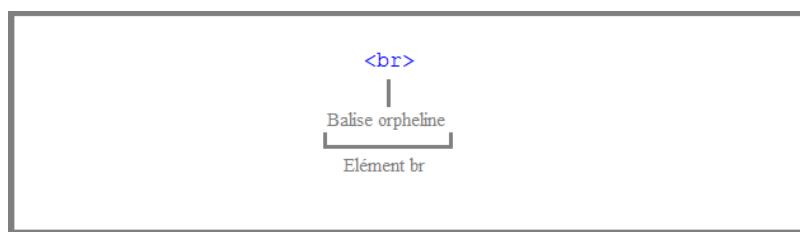
Editeur de code en ligne :

- <https://htmlcodeeditor.com/>
- <https://codesandbox.io>

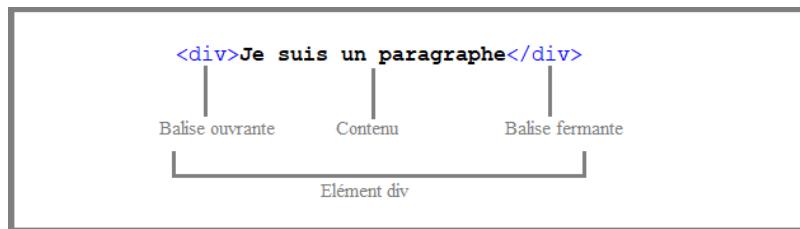
2. HTML

2.1. Balises HTML

2.1.1. balise orpheline



2.1.2. balise en paires



2.1.3. balise avec attribut

Anatomie d'un élément HTML



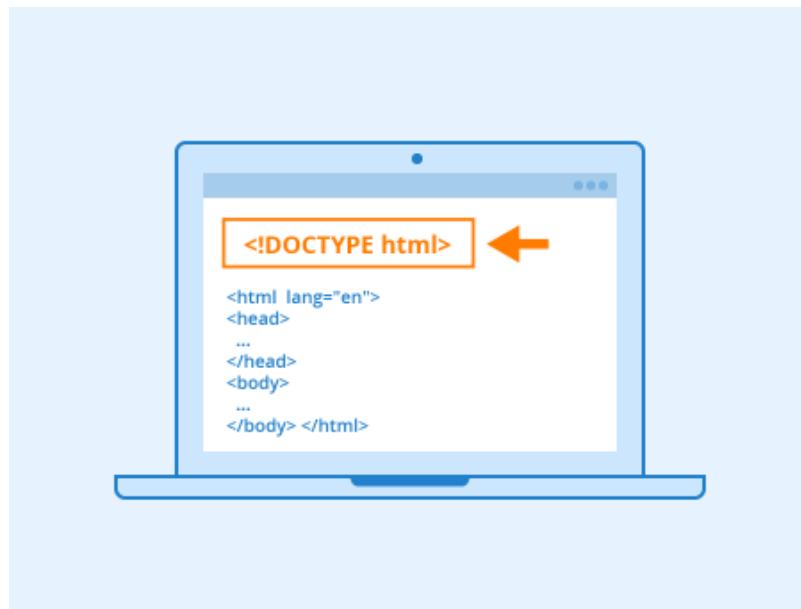
2.2. Structure minimale d'une page htmlBalises HTML

2.2.1. Struture minimale d'une page HTML

```
<!DOCTYPE html>
<html lang="fr">
<head>
<meta charset="utf-8">
<title>Titre de la page</title>
<meta name="description" content="Structure minimale d'une page html" />
</head>
<body>
<!-- Le contenu de la page qui s'affiche --&gt;
&lt;/body&gt;
&lt;/html&gt;</pre>
```

2.2.2. DOCTYPE

La balise de déclaration `<! DOCTYPE>` (insensible à la casse) est utilisée par le navigateur Web pour comprendre la version du HTML utilisé dans le document. La version actuelle de HTML est 5 et utilise la déclaration suivante :



2.2.3. Commentaires

```
<!-- Un commentaire sur une seule ligne -->
<!--
Un commentaire qui débite sur une ligne
et qui se prolonge sur la ligne suivante
-->
<p>Je suis un paragraphe.</p>
```

2.2.4. Indentation

code non indenté :

```
<!DOCTYPE html>
<html lang="fr">
<head>
<meta charset="utf-8">
<title>Titre de la page</title>
<meta name="description" content="Structure minimale d'une page html" />
</head>
<body>
<!-- Le contenu de la page qui s'affiche -->
</body>
</html>
```

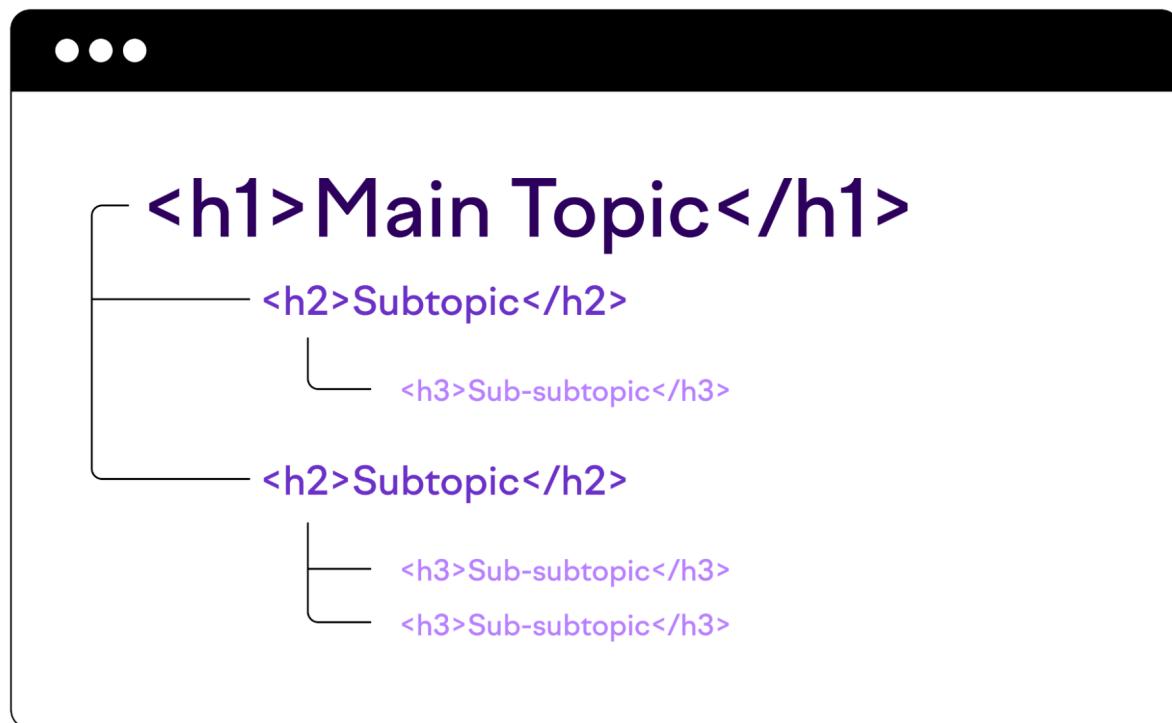
code indenté :

```
<!DOCTYPE html>
<html lang="fr">
<head>
<meta charset="utf-8">
<title>Titre de la page</title>
<meta name="description" content="structure minimale d'une page html" />
</head>
<body>
<!-- Le contenu de la page qui s'affiche -->
</body>
</html>
```

2.3. Contenu ordonné

2.3.1. hiérarchie des titres

```
<h1>Titre de niveau 1</h1>
<h2>Titre de niveau 2</h2>
<h3>Titre de niveau 3</h3>
<h4>Titre de niveau 4</h4>
<h5>Titre de niveau 5</h5>
<h6>Titre de niveau 6</h6>
```



2.3.2. paragraphes

```
<p>Je suis le paragraphe n°1</p>
<p>Je suis le paragraphe n°2</p>
```

2.3.3. ligne horizontale

```
Ligne
<hr>
```

2.4. Mise en forme du texte

2.4.1. Les principales balises de mise en forme

Ce tableau offre un aperçu rapide des balises les plus utilisées pour structurer et mettre en forme le contenu web, soulignant leur fonction principale et leur impact sémantique.

Balise	Rendu à l'écran	Fonction & Impact Sémantique
<code></code>	Texte en gras	Ne donne aucune dimension sémantique au texte.
<code></code>	Texte d'une certaine importance	Indique une importance ou une gravité pour le texte, ajoutant une signification sémantique.
<code><i></code>	<i>Texte en italique</i>	Utilisé pour une voix ou une humeur différente du contenu environnant, sans ajouter de signification spéciale.
<code></code>	<i>Texte accentué</i>	Met en valeur l'importance du texte, souvent interprété comme en italique.
<code><cite></code>	<i>Texte cité</i>	Utilisé pour faire référence à une source, souvent des titres d'œuvres ou des noms.
<code><mark></code>	Texte surligné	Utilisé pour mettre en évidence une partie du texte à des fins de référence.
<code><small></code>	Texte plus petit	Réduit la taille de la police du texte, utilisé pour les petits caractères ou le texte de discréption.
<code><sub></code>	H ₂ O	Affiche un texte en indice, utile pour les formules chimiques ou mathématiques.
<code><sup></code>	E=mc ²	Affiche un texte en exposant, utilisé pour les notes de bas de page ou les formules mathématiques.
<code></code>	Texte supprimé	Indique un texte qui a été supprimé de la documentation.
<code><ins></code>	Texte inséré	Marque les ajouts ou les modifications apportées à un document.
<code><u></code>	<u>Texte souligné</u>	Souligne le texte, mais est souvent utilisé pour indiquer une orthographe incorrecte ou des conventions spéciales.
<code><abbr></code>	Ex	Indique une abréviation ou un acronyme, le titre fournit l'explication complète.
<code><blockquote></code>	Cite un bloc de texte d'une autre source.	Utilisé pour citer de longs extraits de texte d'autres sources, souvent indenté.
<code><q></code>	«Texte cité»	Pour les citations inline, ajoute automatiquement des guillemets dans certains navigateurs.
<code><kbd></code>	Ctrl + S	Représente une entrée utilisateur au clavier, du texte vocal ou toute autre forme d'entrée de commande. Utilisé dans la documentation ou les tutoriels pour indiquer au lecteur les entrées spécifiques à fournir.
<code><pre></code>	Texte préformaté	Affiche le texte avec une mise en forme fixe et préserve les espaces et les sauts de ligne.
<code><code></code>	<code><code>Code informatique</code></code>	Utilisé pour afficher une portion de code informatique avec une distinction visuelle.

2.4.2. Les listes

2.4.2.1. liste non ordonnée

```
<ul>
    <li>Item n°1</li>
    <li>Item n°2</li>
    <li>Item n°3</li>
</ul>
```

- Item n°1
- Item n°2
- Item n°3

2.4.2.2. liste ordonnée

```
<ol>
    <li>Item n°1</li>
    <li>Item n°2</li>
    <li>Item n°3</li>
</ol>
```

1. Item n°1
2. Item n°2
3. Item n°3

2.4.2.3. liste imbriquée

```
<ul>
    <li>Item 1</li>
        <li>
            <ul>
                <li>Item 1.1</li>
                <li>Item 1.2</li>
            </ul>
        </li>
    <li>Item 2</li>
</ul>
```

- Item 1
 - Item 1.1
 - Item 1.2
- Item 2

2.4.2.4. liste de définition

```
<dl>
  <dt>Coffee</dt>
  <dd>- black hot drink</dd>
  <dt>Milk</dt>
  <dd>- white cold drink</dd>
</dl>
```

Coffee

-black hot drink

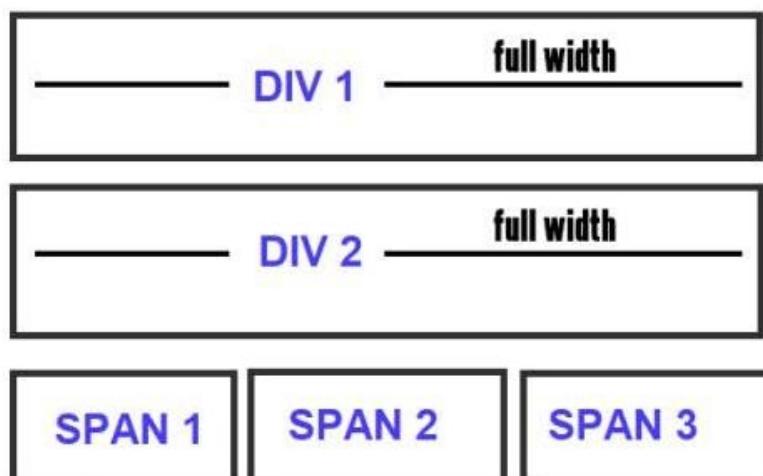
Milk

- white cold drink

	Defines an unordered list
	Defines an ordered list
	Defines an unordered list
	Defines a list item
<dl>	Defines a description list
<dt>	Defines a term in a description list
<dd>	Describes the term in a description list

2.5. Les blocs de contenu

2.5.1. div et span



BLOCK-LEVEL ELEMENTS:



INLINE ELEMENTS:



2.6. les images

2.6.1. les formats d'images

TOP 4 IMAGE FORMATS FOR THE WEB [2018]

	JPG	GIF	PNG	SVG
VECTOR				✓
RASTER	✓	✓	✓	
TRANSPARENCY			✓	✓
ANIMATION		✓	✓	✓
LOSSY	✓			



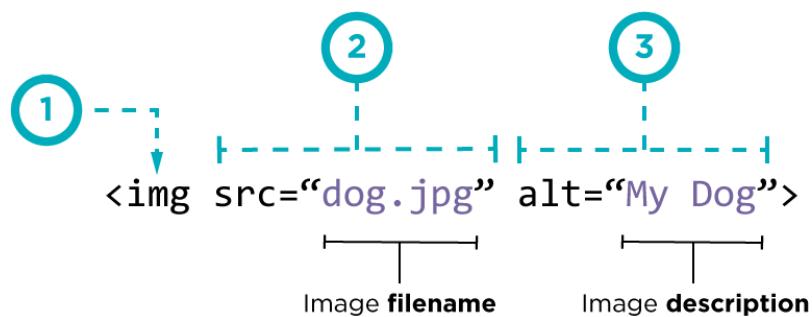
2.6.2. la balise img

HTML IMAGE TAG

```

```

Tag Name Attribute Name Attribute Value Attribute Name Attribute Value

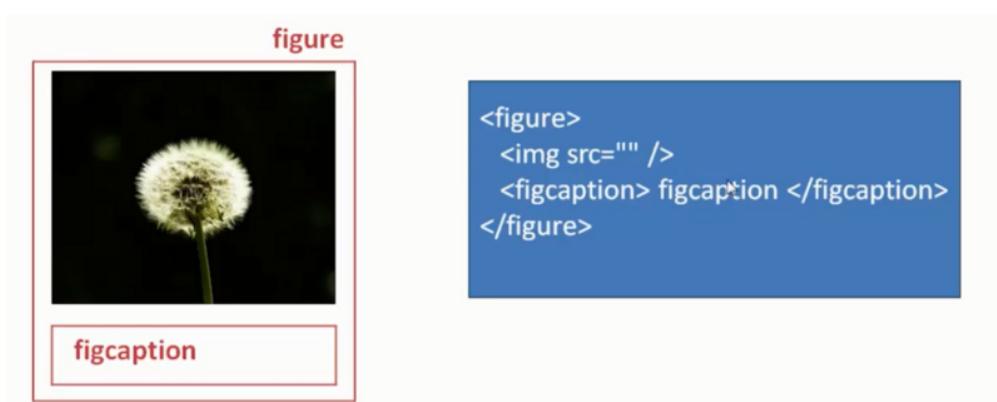


1. Créez une balise d'image en utilisant l'abréviation `img`. Ceci est considéré comme une balise à *fermeture automatique*, car il n'a pas besoin d'envelopper le texte comme le font de nombreuses autres balises. La `/` droite avant la fin `>` est facultative, mais nous rappelle que cette balise n'a pas besoin de balise fermante.
2. L' attribut `src` est l'abréviation de source . Cela indique au tag quelle image charger. Dans ce cas, la page cherchera une image avec le nom de fichier `dog.jpg` dans le même répertoire que la page. Les noms de fichiers image incluent des extensions qui indiquent à l'ordinateur avec quel type d'image il travaille. Les extensions courantes sont `.jpg` , `.jpeg` et `.png` . Assurez-vous de mettre des guillemets autour du nom de votre fichier image.

3. L' attribut alt est l'abréviation de texte alternatif . Bien que vous ne voyiez pas ce texte sur votre page Web, il fournit une sauvegarde au cas où votre image ne se téléchargerait pas correctement ou pour les utilisateurs malvoyants. Dans cet exemple, si votre navigateur ne parvenait pas à charger l'image, vous verriez



2.6.3. figure / figcaption

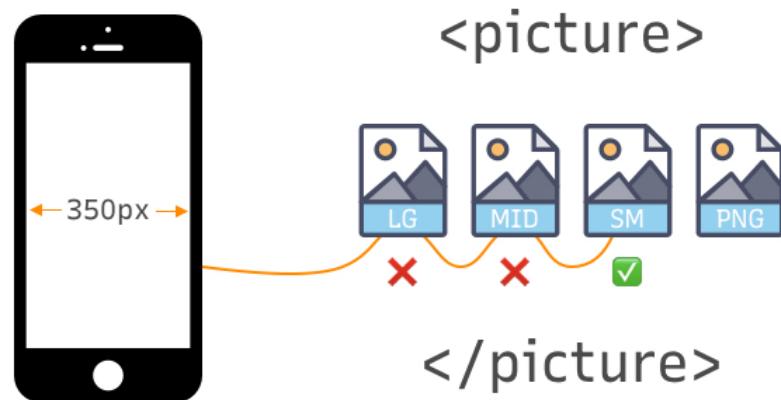


2.6.4. picture

L'élément `<picture>` est une balise qui permet aux développeurs web de spécifier plusieurs sources pour une image, afin que le navigateur puisse choisir la source d'image la plus appropriée à afficher selon les critères comme la taille de l'écran, le type de fichier, ou la résolution de l'écran.

Cette balise est particulièrement utile pour créer des sites web réactifs (responsive) qui s'adaptent à divers appareils et dimensions d'écran.

```
<picture>
  <source media="(min-width: 1024px)" srcset="banner-large.jpg" />
  <source media="(min-width: 700px)" srcset="banner-medium.jpg" />
  <source media="(max-width: 699px)" srcset="banner-small.jpg" />
  
</picture>
```



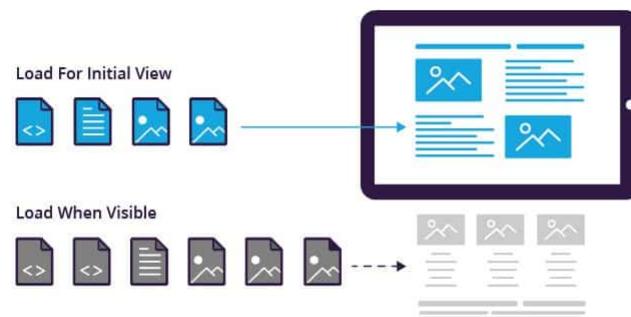
2.6.5. lazy loading

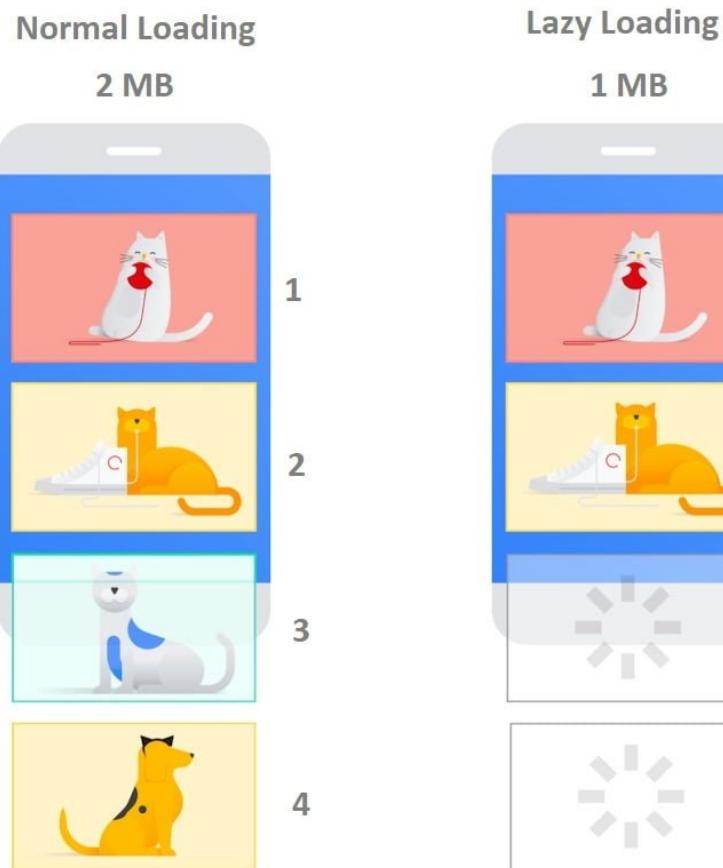
Quand une page web est chargée par un navigateur, par défaut, toutes les ressources de la page sont chargées immédiatement.

Avec le lazy loading, les ressources hors de l'écran (qui ne sont pas encore visibles pour l'utilisateur) ne sont pas chargées immédiatement. Elles ne le seront que lorsque l'utilisateur se rapproche de leur emplacement en faisant défiler la page.

```

```

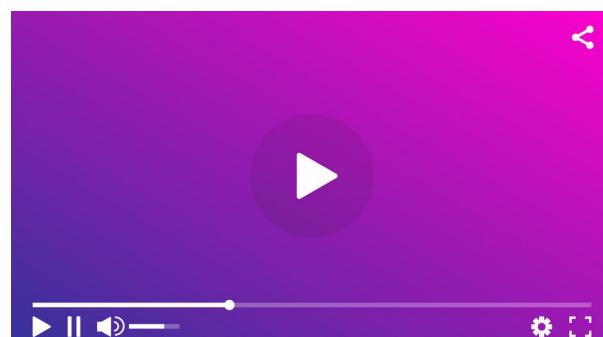




3. Les vidéos

Pour intégrer une vidéo dans une page web en utilisant HTML, vous utilisez l'élément `<video>`. Cet élément permet de lire des vidéos directement sur la page, sans nécessiter de plugin externe.

```
<video controls loop autoplay muted width="250">
  <source src="chemin/vers/la/video.mp4" type="video/mp4">
  <source src="chemin/vers/la/video.ogg" type="video/ogg">
  Votre navigateur ne prend pas en charge la balise vidéo.
</video>
```



Pour assurer une compatibilité maximale avec différents navigateurs, fournissez votre vidéo dans plusieurs formats (comme `.mp4` et `.ogg`).

3.1. Les liens

3.1.1. les ancre nommées / lien d'évitement

3.2. Lien de retour en haut de page

Un lien de retour en haut de page ne diffère pas des liens internes classiques. Il faut créer une ancre, puis pointer vers cette ancre à chaque fois que l'on veut faire un lien de retour. Pour créer l'ancre, on peut placer au début de l'élément `body` ou du premier `div` structurant le code suivant:

```
<a id="haut"></a>
<!-- On peut bien sûr choisir n'importe quel autre nom que "haut" --&gt;</pre>

```

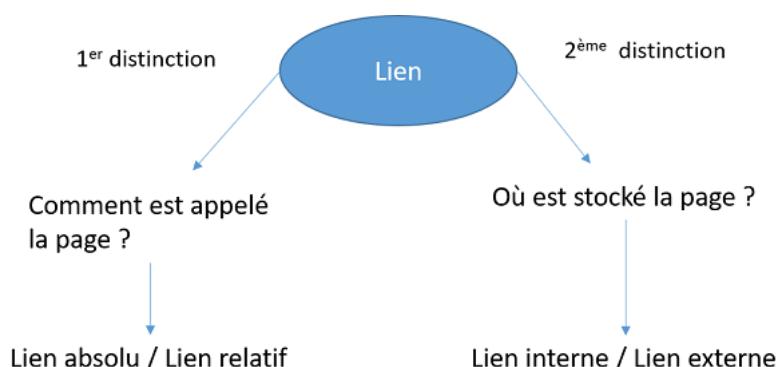
On pourra ensuite faire un lien vers cette ancre en précisant dans l'attribut `href` le nom de l'ancre précédé de `#` :

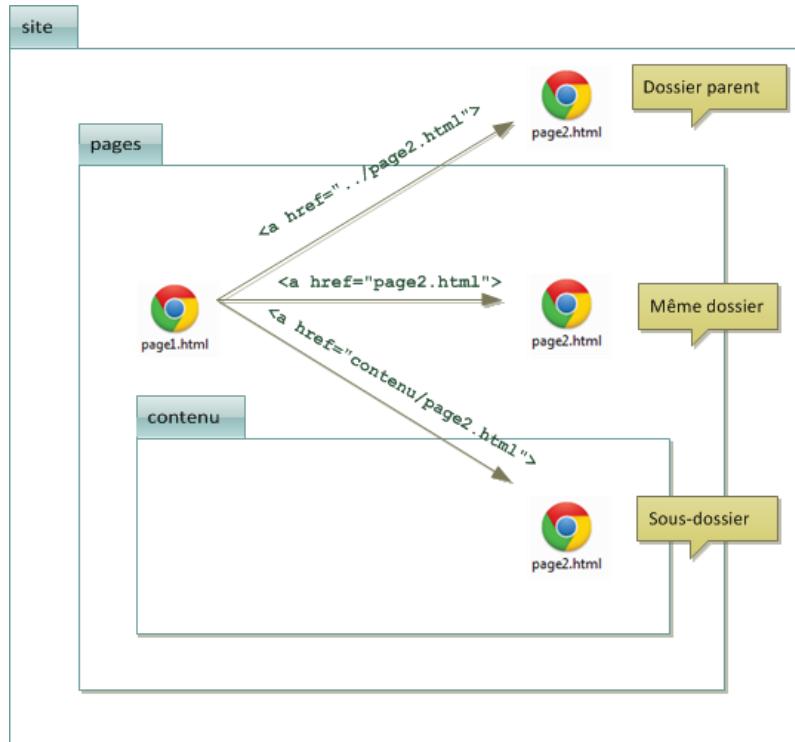
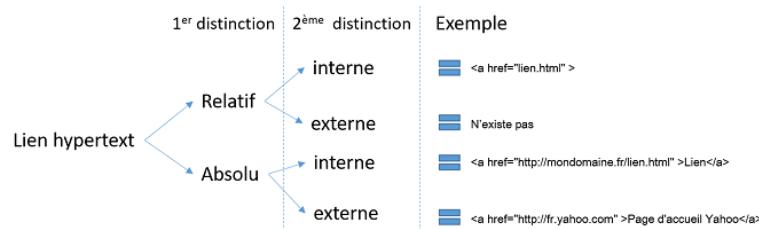
```
<a href="#haut">haut de page</a>
```

Avec la plupart des navigateurs, aucune ancre n'est nécessaire pour retourner en haut de page : le dièse `#` seul suffit à revenir au début du contenu.

```
<a href="#">haut de page</a>
```

Attention cependant, jusqu'à Opera 9 inclus ce fonctionnement n'est pas compris nativement, il faudra lui préférer la méthode de création d'une ancre en bonne et due forme, puis faire pointer les liens de retour en haut de page vers cette ancre.





3.2.1. Les attributs target

L'attribut **target** dans les éléments **<a>** en HTML spécifie où ouvrir le lien ciblé.

Les valeurs **_blank** et **_self** sont deux des options les plus couramment utilisées pour cet attribut.

- **_blank**

Ce lien ouvrira <https://www.example.com> dans un nouvel onglet de la fenêtre actuelle du navigateur.

```
<a href="https://www.example.com" target="_blank">Visitez Exemple.com</a>
```

- **_self**

Ce lien mènera à </contact.html> dans le même onglet et la fenêtre actuelle du navigateur.

Cependant, puisque `_self` est la valeur par défaut, l'attribut `target="_self"` est souvent omis.

```
<a href="../contact.html" target="_self">Contactez-nous</a>
```

3.2.2. L'attribut download

L'attribut `download` dans les éléments `<a>` en HTML indique au navigateur de télécharger l'URL ciblée au lieu de naviguer vers celle-ci. Cet attribut peut être particulièrement utile lorsque vous souhaitez fournir des fichiers à télécharger directement depuis votre site web, tels que des documents PDF, des images, ou des fichiers zip.

```
<a href="downloads/toto.pdf" download="Gateau_Chocolat">Mon fichier PDF</a>
```

3.2.3. mailto

L'attribut `mailto:` est utilisé dans les liens HTML (`<a>`) pour créer un moyen rapide pour les visiteurs d'envoyer un email en cliquant sur le lien.

```
<a href="mailto:exemple@domaine.com">Envoyez-moi un email</a>
```

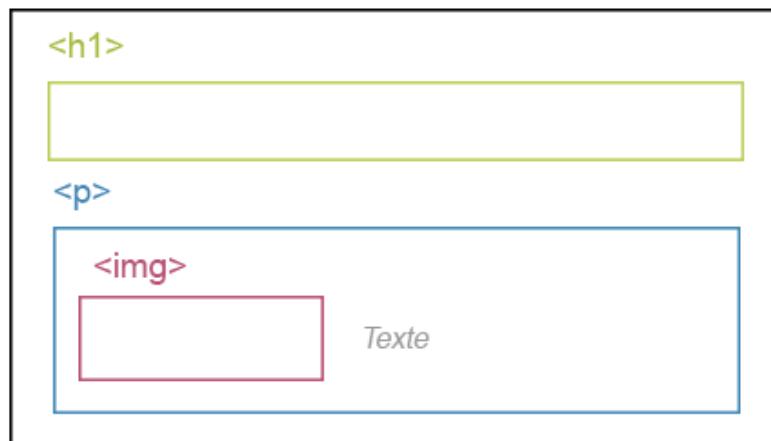
Le fichier téléchargé sera automatiquement renommé en "Gateau_Chocolat.pdf".

3.3. le flux

3.3.1. Emboîtements

Tous les éléments possèdent des règles définissant clairement : dans quel autre élément (parent) ils peuvent être placés, et quels autres éléments (enfants) ils peuvent contenir. Par ailleurs, des balises bien précises sont auto-fermant, c'est-à-dire des éléments vides et ne doivent -ou ne peuvent- rien contenir du tout ; c'est le cas de ``. Ces règles ne doivent pas être vues comme une contrainte, mais comme le fruit d'une longue réflexion et d'un usage naturel. Il serait par exemple absurde de placer un élément `<input>` dans un `` ou un `<hgroup>` dans un `<select>`.

<article>



En respectant ces principes, votre document sera valide et aura toutes les chances d'être bien analysé syntaxiquement, interprété et compris par tous les navigateurs.

De nouvelles exceptions voient le jour toutefois : les liens `<a>` peuvent désormais englober un ou plusieurs éléments de type bloc, contrairement à ce qui était indiqué par les précédentes versions de HTML.

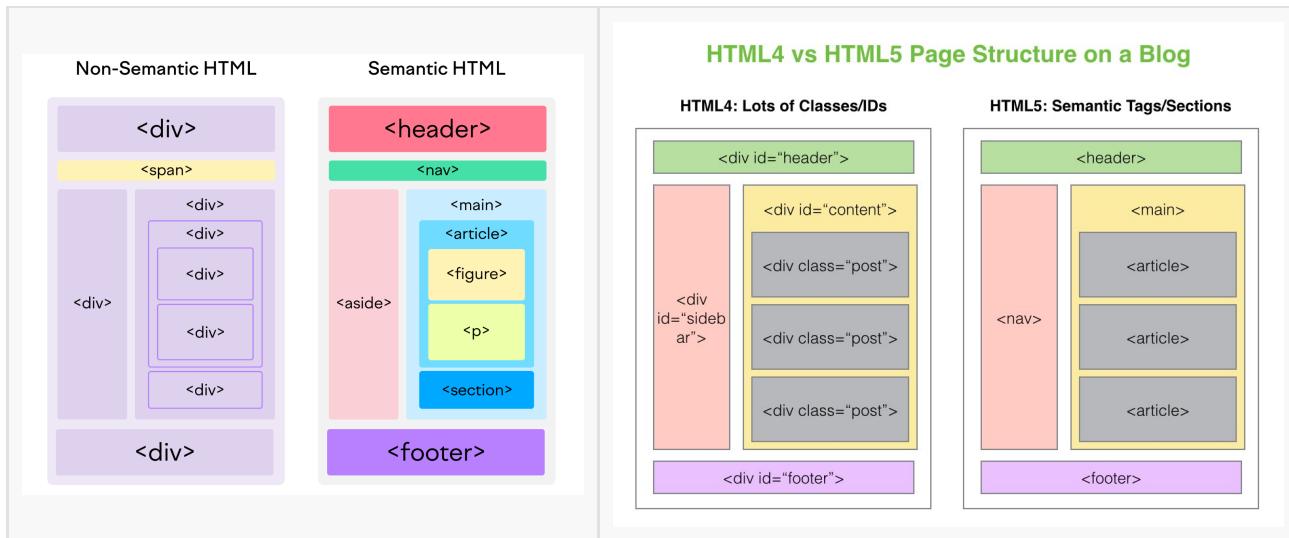
3.4. Les balises sémantiques

Une balise sémantique est une balise qui donne une indication sur le contenu qu'elle entoure. Comprenez que les balises `<div>` ou `` ne peuvent donner aucune indication sur les éléments qu'elles contiennent du fait que la documentation HTML ne donne pratiquement aucune limite au type de contenu possible entre ces balises, ce ne sont donc pas des balises sémantiques.

Certaines balises ont alors été ajouté à HTML pour permettre un découpage plus précis de votre page web en délimitant des zones spécifiques.

L'un des autres atouts de ce nouveau type de balisage, est le fait qu'il permette une meilleure accessibilité des contenus pour les personnes nécessitant des logiciels de lecture de site pour consulter le web.

Les balises sémantiques permettent de donner un sens aux éléments qu'elles contiennent, et donc aux pages web.

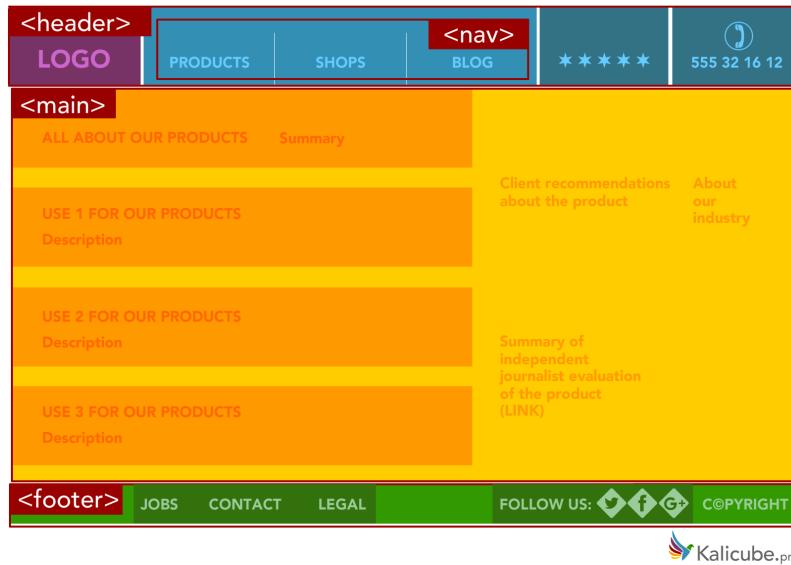


3.4.1. L'entête du site : `<header>`

- Sur un site internet, on peut observer la partie haute, appelée ‘entête’ ou ‘header’ en anglais, que l’on retrouve sur l’ensemble des pages du site.
- Cette zone peut alors être définie à l’aide de la balise `<header>`.
- De manière générale, on retrouvera dans cette zone le logo, et au minimum, la navigation principale du site.
- La balise `<header>` peut apparaître plusieurs fois sur la page, elle peut également être utilisée pour délimiter l’entête d’un article de blog ou d’une section par exemple.
- Cependant conservez une structure simple pour débuter et n’utilisez la balise `<header>` que pour **délimiter l’entête générale du site web**.
- Une balise `<header>` ne peut pas être contenue dans une autre balise `<header>`.

3.4.2. Les menus de navigation : `<nav>`

- Dans la balise `<header>` vu précédemment, nous pouvons ajouter une zone servant à **délimiter la navigation principale du site web**. Cette zone peut être définie à l’aide de la balise sémantique `<nav>`.
- Il faut tout de même savoir que la balise `<nav>` n’est pas forcément placée dans le header d’un site, on peut la retrouver n’importe où et à plusieurs endroits de la page comme par exemple dans le footer ou dans une barre latérale de contenu (sidebar).

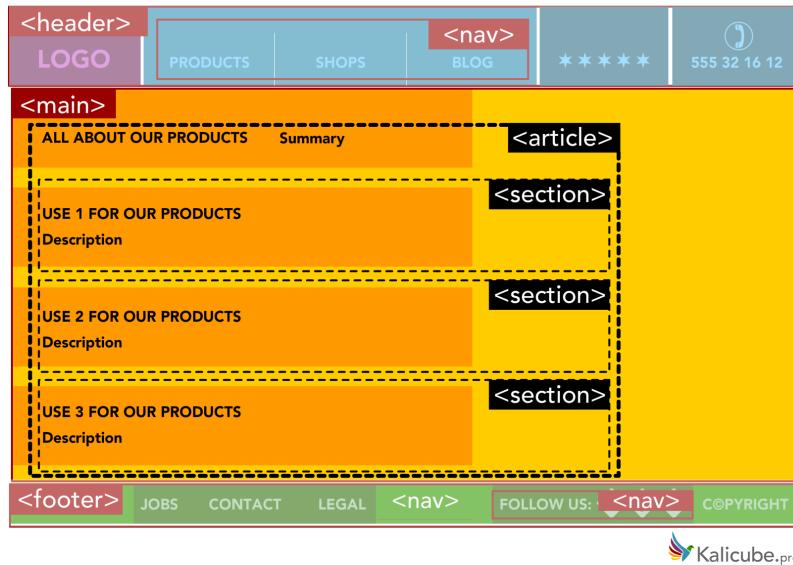


3.5. Le contenu principal : `<main>`

- La balise `<main>` sert à définir la zone (souvent grande) de votre page qui possède le contenu utile de la page. Dans l'idéal, le contenu de la balise `<main>` devrait être unique à la page.
- C'est dans cette balise que vous allez coder la majorité du contenu de votre site web.
- N'utiliser qu'une seule balise `<main>` par page.

3.6. Le contenu indépendant : `<article>`

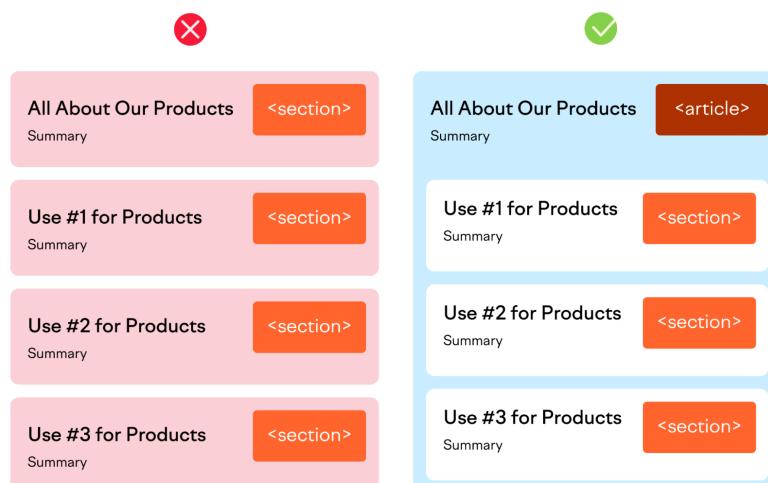
- Cette balise va permettre de signaler aux robots d'indexation quel éléments sont en lien avec le sujet de la page.
- Vous pouvez glisser dans la balise `<article>`, le titre, les informations sur l'auteur ou la date de publication, la catégorie, le corps de l'article, et les articles annexes d'une même thématique.
- La bonne tenu de la balise article va également dépendre de l'objectif de votre site web.
- Cette balise a d'autant plus d'importance sur un site e-commerce, un site d'actualité ou un blog.



Kalicube.pro

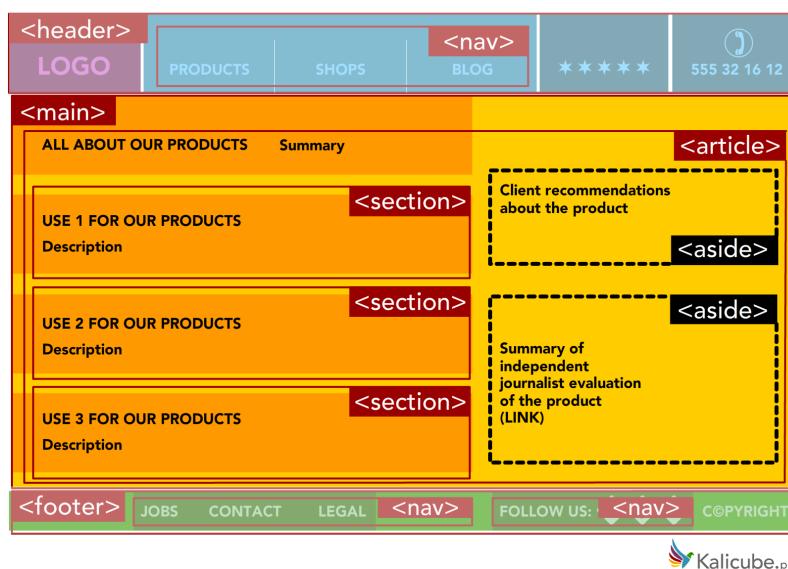
3.7. Un découpage plus précis : `section`

- Le contenu de notre site présent dans une des balises `<main>` ou `<article>` peut être plus ou moins complexe, c'est pour cela qu'il est important de correctement structurer le contenu principal de la page en plusieurs zones.
- Pour cela, il existe la balise `<section>` permettant un découpage plus thématique que les balises `<main>` et `<article>` vues précédemment.
- Chaque nouvelle balise `<section>` indique aux robots d'indexations une variétés de contenu différents pour une même page.
- Il est possible d'avoir un découpage encore plus précis en utilisant cette fois la balise générique `<div>` (pour division) et `` (pour ‘espace’ / ‘zone’) à l'intérieur des sections.
- Ce n'est pas parce qu'il existe maintenant beaucoup plus de balises de structure qu'il ne faut plus utiliser la balise `<div>`, cette dernière reste toujours extrêmement utile et constitue souvent la majorité de la structure d'une page.



3.8. Contenu annexe : <aside>

- Pour certaines pages, il est possible de proposer un encart de contenu non relié directement au contenu principal de la page.
- Vous pourrez alors encadrer ce contenu à l'aide de la balise `<aside>` tout simplement.
- La balise `<aside>` est souvent utilisée pour délimiter des sidebars (les barres latérales de contenu annexes que l'on retrouve souvent sur des blogs), ou des éléments complémentaires au contenu principal de la page.
- Il n'y a pas de règle quant à l'endroit où utiliser cette balise, libre à vous de choisir en fonction de votre site web.



3.9. Le pied de page : <footer>

- Cette balise sert principalement à délimiter la pied de page.
- On retrouvera alors souvent cette balise à côté de la balise <main>.
- Dans le **footer** on insère en général des liens de navigation, le logo du site et le lien vers les mentions légales.
- On peut également placer des balises <nav> dans le **footer** afin de délimiter des zones de navigation secondaire.
- On peut également utiliser la balise <footer> pour baliser une fin d'article ou de section, de la même manière que le fait le header.
- On pourra alors y placer les crédits et **les informations annexes ayant une importance moindre** que le contenu principal.

4. Les formulaires

Un formulaire HTML est utilisé pour collecter différentes sortes d'informations de l'utilisateur du site web. Il peut contenir des champs de texte, des boutons radio, des cases à cocher, et plus encore.

Les formulaires sont essentiels pour les interactions utilisateur, incluant les inscriptions, les connexions, et la soumission de commentaires.

4.1. Le duo <label> et <input>

Ce duo joue un rôle crucial dans la création de formulaires web accessibles et conviviaux.

4.1.1. La balise <label>

Cette balise est utilisée pour définir une étiquette pour plusieurs éléments de formulaire (**input**, **select**, **textarea**).

- Elle améliore l'**accessibilité** (voir section sur l'Accessibilité) en permettant aux utilisateurs de cliquer sur le texte de l'étiquette pour se focaliser ou activer le contrôle de formulaire associé. Cela est particulièrement utile pour les petits éléments de formulaire comme les cases à cocher ou les boutons radio.

- L'attribut `for` de la balise `<label>` doit correspondre à l'`id` de l'élément de formulaire qu'elle décrit, créant ainsi une association explicite entre l'étiquette et le champ de formulaire :

Exemple :

```
<label for="prenom">Prénom :</label>
<input type="text" id="prenom" name="prenom">
```

4.1.2. La balise `<input>`

Cette balise est utilisée pour créer un contrôle interactif dans un formulaire web qui permet à l'utilisateur de saisir des données.

- Elle peut prendre divers types spécifiés par l'attribut `type`, :
 - `text`, `password`, `email`, `checkbox`, `radio`, et plus encore, chacun définissant le genre d'information que l'utilisateur peut entrer.
- L'`id` est un identifiant unique attribué à chaque élément d'entrée, qui peut être utilisé pour associer la balise `label` correspondante.

Exemple :

4.2. Zone de texte : `<textarea>`

La balise `<textarea>` est utilisée en HTML pour créer une zone de texte multi-lignes, permettant aux utilisateurs de saisir du texte sur plusieurs lignes (ce qui n'est pas le cas de la balise `<input>`).

Elle est souvent employée pour des champs de saisie plus larges comme les commentaires ou les descriptions.

Attributs courants :

- `rows` et `cols` spécifient respectivement le nombre de lignes et de colonnes dans la zone de texte, déterminant ainsi sa taille.
- `name` identifie la zone de texte, permettant de transmettre ses données au serveur lors de la soumission du formulaire.

Exemple :

```
<label for="commentaire">Commentaire :</label>
<textarea id="commentaire" name="commentaire" rows="4" cols="50" placeholder="Entrez votre commentaire ici...">
</textarea>
```

4.3. L'attribut `required`

L'attribut `required` en HTML est un attribut booléen qui peut être ajouté à la plupart des éléments de formulaire (`<input>`, `<select>`, `<textarea>`, etc.) pour indiquer que l'utilisateur doit remplir ce champ avant de soumettre le formulaire.

Si le formulaire est soumis et que le champ marqué comme `required` est vide, le navigateur affiche un message d'erreur et empêche la soumission du formulaire jusqu'à ce que l'utilisateur ait fourni la valeur nécessaire.

Champ de texte requis :

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

Zone de texte requise :

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

4.4. Bouton de soumission

Le bouton de soumission dans un formulaire HTML est un élément crucial qui permet aux utilisateurs d'envoyer les données du formulaire à un serveur pour traitement.

4.4.1. Bouton de Soumission Standard

En HTML il est possible de créer un bouton de soumission de deux manières :

En utilisant la balise `<input>` :

```
<input type="submit" value="Envoyer">
```

En utilisant la balise `<button>` :

```
<button type="submit">Envoyer</button>
```

4.5. Récapitulatif

La création de formulaires HTML, met l'accent sur l'utilisation des éléments `label` et `input` pour améliorer l'accessibilité et l'expérience utilisateur.

Les champs spécifiques comme les zones de texte (`<textarea>`) permettent des saisies multi-lignes, et l'importance de l'attribut `required` a été soulignée pour assurer que les champs essentiels soient remplis avant la soumission du formulaire.

Exemple de formulaire complet :

```
<form action="" method="post">
  <div>
    <label for="prenom">Prénom :</label>
    <input type="text" id="prenom" name="prenom" required />
  </div>
  <div>
    <label for="email">Email :</label>
    <input type="email" id="email" name="email" required />
  </div>
  <div>
    <label for="message">Votre message :</label>
    <textarea id="message" name="message" rows="5" required></textarea>
  </div>
  <div>
    <button type="submit">Envoyer</button>
  </div>
</form>
```

5. Les tableaux

Les tables sont un moyen efficace en HTML de présenter des données en lignes et colonnes. Cependant, pour les rendre accessibles à tous les utilisateurs, y compris ceux qui utilisent des lecteurs d'écran, il est crucial d'utiliser correctement les balises et attributs HTML.

Synthèse des éléments d'un tableau en HTML :

Élément	Description
<code><table></code>	L'élément conteneur pour une table HTML. Tous les autres éléments de table (tels que <code><tr></code> , <code><td></code> , <code><th></code>) doivent être insérés à l'intérieur de cet élément.
<code><th></code>	Représente une cellule d'en-tête dans une table, généralement utilisée dans <code><thead></code> pour définir les en-têtes de colonnes ou de lignes. L'attribut <code>scope</code> (<code>col</code> , <code>row</code>) spécifie si la cellule d'en-tête est pour une colonne ou une ligne.
<code><tr></code>	Définit une ligne dans une table (table row). Les lignes contiennent des cellules, qui peuvent être des cellules de données (<code><td></code>) ou des cellules d'en-tête (<code><th></code>).
<code><td></code>	Représente une cellule de données (table data) standard dans une table. Ces cellules sont les contenus de votre table.
<code><thead></code>	Utilisé pour grouper l'en-tête de contenu d'une table. Il est bon pour la sémantique et aide les technologies d'assistance à comprendre la structure de la table.
<code><tbody></code>	Utilisé pour grouper le corps de contenu d'une table, séparant ainsi les données principales des en-têtes (<code><thead></code>) et du pied de table (<code><tfoot></code>).
<code><tfoot></code>	Utilisé pour grouper le pied de contenu d'une table. Les navigateurs peuvent utiliser cet élément pour permettre à l'utilisateur de naviguer plus facilement dans de grandes tables, affichant le <code><tfoot></code> de manière cohérente pendant le défilement.
<code>scope="col"</code>	Attribut de <code><th></code> indiquant que l'en-tête s'applique à une colonne. Aide à établir des relations entre les en-têtes de colonne et les cellules de données pour une meilleure accessibilité.
<code>scope="row"</code>	Attribut de <code><th></code> indiquant que l'en-tête s'applique à une ligne. Utilisé principalement dans les corps de table (<code><tbody></code>) pour associer des en-têtes de ligne avec les cellules de données de cette ligne.

Exemple d'un tableau complet et accessible sans mise en forme css :

```

<table>
  <caption>
    Dépenses mensuelles
  </caption>
  <thead>
    <tr>
      <th scope="col">Catégorie</th>
      <th scope="col">Janvier</th>
      <th scope="col">Février</th>
      <th scope="col">Mars</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row">Nourriture</th>
      <td>300€</td>
      <td>280€</td>
      <td>350€</td>
    </tr>
    <tr>
      <th scope="row">Logement</th>
      <td>700€</td>
      <td>700€</td>

```

```
<td>700€</td>
</tr>
<tr>
  <th scope="row">Transport</th>
  <td>150€</td>
  <td>120€</td>
  <td>160€</td>
</tr>
</tbody>
<tfoot>
<tr>
  <th scope="row">Total</th>
  <td>1150€</td>
  <td>1100€</td>
  <td>1210€</td>
</tr>
</tfoot>
</table>
```

Rendu écran :

Dépenses mensuelles			
Catégorie Janvier Février Mars			
Nourriture	300€	280€	350€
Logement	700€	700€	700€
Transport	150€	120€	160€
Total	1150€	1100€	1210€