

HTML

LES FONDAMENTAUX

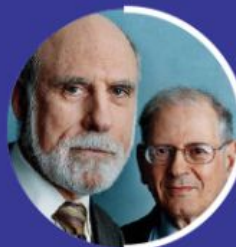
Histoire du Web



1969 : LA DARPA CRÉE

ARPANET

Le 1er réseau à transfert de paquets développé par la Defense Advanced Research Projects Agency - programme de défense nationale Américain



ANNÉES 1980

Vinton Gray Cerf & Bob Kahn inventent le **protocole TCP/IP**

En combinant leurs deux protocoles, ils inventent le principe qui reste la base de l'Internet d'aujourd'hui
Le 1er janvier 1983, ARPANET adopte le protocole TCP/IP

ARPANET devient Internet pour servir de support au développement de nouvelles applications basées sur le transfert de données

INTERNET

WORLD WIDE WEB

Email

Live Chat

Forum

Peer to Peer

...



13 MARS 1989

Sir Timothy Berners-Lee :
inventeur et guru du Web

Sir Timothy Berners-Lee propose de développer un système hypertexte organisé en toile, afin d'améliorer la diffusion des informations dans le cadre d'un projet du CERN de Genève sur l'acquisition et le traitement des données qu'il rendra public le 6 août 1991



W3C

En octobre 1994, il fonde le World Wide Web Consortium, organisme à but non lucratif chargé de normaliser et de promouvoir la compatibilité des technologies du Web

1989-1992 Création du langage ←
1993 Définition de HTML+ →
1994 Développement des outils en sémantique ←
1997 Validation de HTML 4.0 par le W3C →
2000 / 2006 Abandon du HTML en faveur du XHTML ←
2007 Le W3C relance le développement d'un nouveau HTML →
2007 / 2014 Développement d'outils de compatibilités de langage ←

28 OCTOBRE 2014

HTML5



Balises sémantiques adaptées



Composants spécifiques



Compatible avec les langages



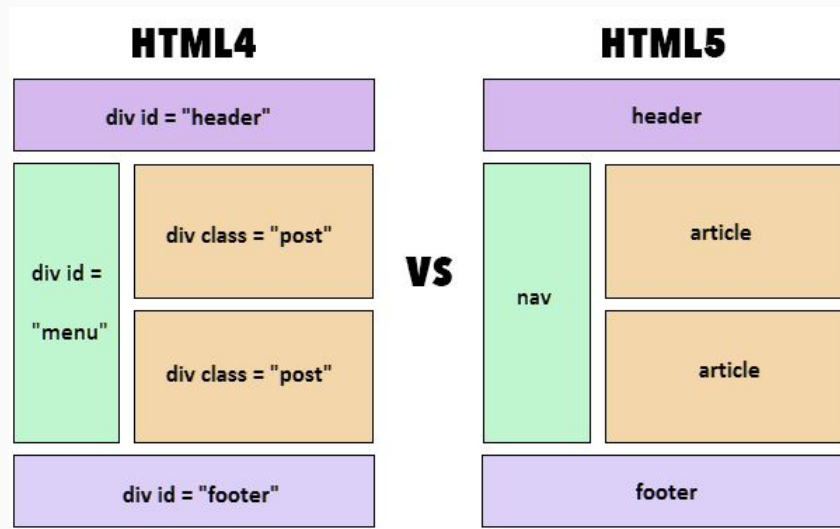
Centré sur l'utilisateur



Développé en permanence

Après l'abandon du XHTML 2.0, le World Wide Web Consortium travail sur la normalisation du **HTML5** qui à pour but de décrire la sémantique des documents ou les applications en ligne et d'enrichir les interfaces utilisateurs avec des contrôles spécifiques en assurant la compatibilités avec les différents langage de l'Internet

HTML4 vs HTML5



Squelette de base d'une page HTML

Le squelette d'une page HTML est la structure qui permet de créer et d'organiser du contenu. Pour qu'il soit reconnu par les navigateurs, il doit respecter une orthographe et un ordre strict.

La balise `<!DOCTYPE html>` correspond au type de document, cela initialise le HTML 5

`<html>` Ouverture de la page

`<head>` Configuration de la page
Toujours en premier
(méta données, titre, css, etc.)

`<body>` Tout le contenu de la page

`</html>` Fermeture de la page

```
<!DOCTYPE html>

<html lang="fr">

  <head>
    <!-- Entête de la page -->
    <meta charset="UTF-8" />
    <title>Document</title>
  </head>

  <body>
    <!-- Corps de la page -->
    <p>Hello World!</p>
  </body>

</html>
```

Balise <head>

En partant du haut le squelette commence par la tête. En HTML, la balise représentant la tête contient les informations nécessaires au fonctionnement du corps de la page.

<head> Ouverture de la balise

Charset - Détermine la façon dont le texte est transmis et stocké

UTF-8 - Universal Character Set Transformation Format - 8 bits

Keywords - Mots clés de la page

Viewport - Surface de la fenêtre pour l'affichage mobile

Title - Titre de la page

Link - Définit la relation entre le document courant et une ressource externe. Ici une feuille de style CSS

</head> Fermeture de la balise

```
<head>
  <meta charset="UTF-8" />

  <meta name="keywords" content="exercise,html,css" />

  <meta name="viewport" content="initial-scale=1" />

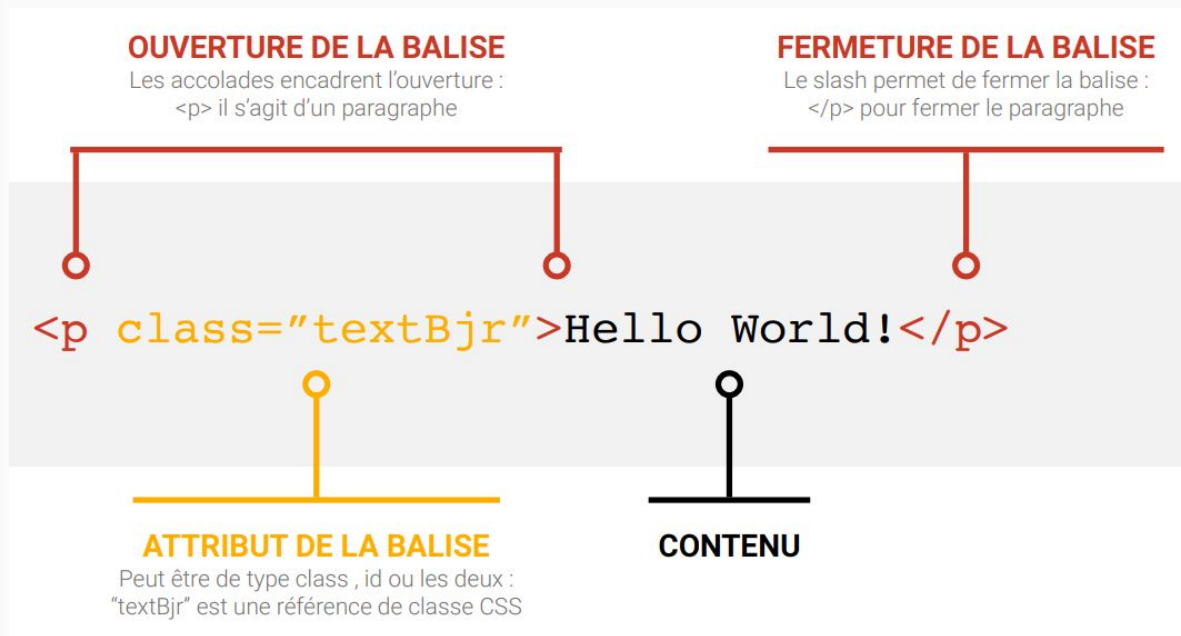
  <title>Hello World</title>

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

Structure des balises HTML

Les balises HTML peuvent être de type block ou in-line.

Leurs comportements sont différents mais leurs structures restent similaires. Il faut toujours ouvrir et fermer une balise HTML et lui ajouter un contenu puis en option un attribut.



Exceptions

Certaines balises ne nécessitent pas d'être fermées ou encore d'avoir du contenu mais seulement des attributs.

La balise `
` pour les sauts de lignes ne se ferme pas.

La balise `<input>` ou encore `` ne se ferment pas, n'ont pas de contenu excepté leurs attributs.

Les types de balises HTML

Les types de balises HTML ont chacune une fonction particulière: le type bloc permet de créer des sections et le type in-line permet de créer des sous-sections. Les balises de types blocs englobent des balises de type in-line.

BALISES DE TYPE BLOC

Elles occupent toute la largeur disponible. Lorsque deux éléments blocs se suivent dans une page, ils sont positionnés (par défaut) l'un sous l'autre.

<section> Balise sectionnante HTML5
<h1> Balise de titre (1 à 6)
<p> Balise de paragraphe

```
<section>

  <h1>Duis aute irure dolor in reprehenderit</h1>

  <p>Lorem ipsum sit amet</p>

</section>
```

BALISES DE TYPE INLINE

Elles n'occupent que la largeur indispensable à l'affichage du contenu et ne provoque pas de retour à la ligne.

**** Rend un texte gras
<i> Rend le texte italique
**** Balise pour charger une image

```
<section>

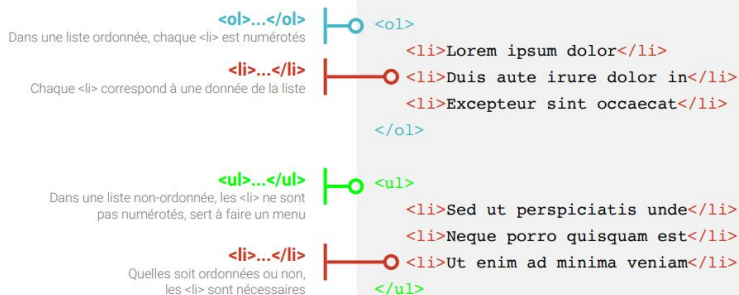
  <p><strong>2010-2015 :</strong> Lorem ipsum sit amet,
  dolor <i>Consectetur</i></p>

</section>
```

Listes

Une liste HTML est un paragraphe structuré contenant une suite de lignes, ces lignes peuvent être ordonnées ou non-ordonnées. Pour fonctionner, une liste a besoin d'un conteneur `` ou `` dans lequel sont associées des balises ``.

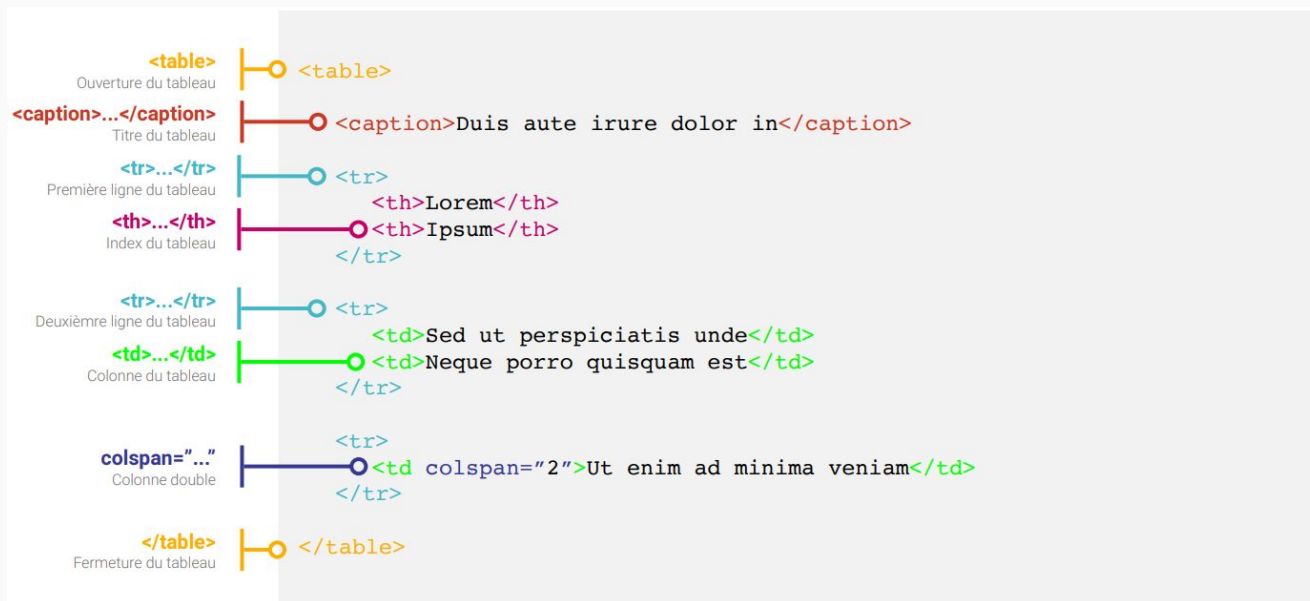


Les liens HTML peuvent être de deux types: interne ou externe. Les liens internes servent à gérer la navigation sur plusieurs pages d'un même site alors que les externes affichent des pages qui ne font pas partie du site, les liens externes doivent être spécifiés à l'utilisateur.



Tableaux

Les tableaux HTML ont un fonctionnement proche des listes mais sont composés de plus de balises. Il est possible d'afficher un tableau avec différentes options comme un titre ou un index mais également un header et un footer.



Pour réaliser un formulaire HTML il faut utiliser une suite de balises HTML pour permettre à l'utilisateur de renseigner des informations. Comme pour une liste ou un tableau, un formulaire est englobé dans un conteneur dans lequel sont associées des balises `<label>` et `<input>`.

`<form ... >`
Ouverture du formulaire, la méthode et l'action font référence à la programmation PHP du formulaire

`<form method="post" action="connexion.php">`

`<label ... >`
Les labels sont associés aux inputs, la valeur de `for="..."` doit être égale au `id="..."` du input.

`<label for="lePseudo">Votre pseudo</label> :`
`<input type="text" name="pseudo" id="lePseudo" />`

`<input ... >`
Les inputs permettent à l'utilisateur de renseigner ces informations et peuvent être de différents types.

`<label for="lePass">Votre mot de passe :</label>`
`<input type="password" name="pass" id="lePass" />`

`<input type="submit" ... >`
Ce input est nécessaire pour valider un formulaire.

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

`</form>`
Fermeture du formulaire, tous les inputs doivent être placés entre les balises `<form>...</form>`.

`</form>`

À suivre...