

LES BASES HTML 5



1- DÉFINITION HTML

- **HTML** : **H**yper**T**ext **M**arkup **L**anguage
- Est un langage de **balisage** (dit aussi langage de **marquage**) qui permet de structurer le contenu des pages web
- HTML n'est pas un langage de programmation, mais c'est juste un langage de description de documents. Il permet de faire deux types d'actions essentielles :
- Définir des zones de contenu
- Placer des liens permettant de naviguer entre les pages.
- Le navigateur reçoit le HTML, l'interprète, et produit la page telle qu'affichée au client.
-



2- HISTORIQUE

- 1989-1992 : HTML est une des 3 inventions à la base du World Wide Web, avec le HTTP et les
- adresses web (URL). Inventés par Tim Berners-Lee
- 1993 : HTML 1.0
- 1995-1996 : HTML 2.0
- 1997 : HTML 3.2 et HTML 4.0
- 1999 : HTML 4.01
- 2000-2006 : XHTML une version extensible de HTML
- 2007 à nos jours: HTML 5 et abandon du XHTML



3- SÉPARATION ENTRE CONTENU ET MISE EN FORME

- Une page web créée avec HTML 5 doit être pensée en distinguant deux parties:
- Un contenu, structuré au moyen des éléments HTML : à ce stade, on décrit principalement l'organisation des informations à fournir à l'utilisateur (grandes divisions, titres, paragraphes, tableaux, images et liens, etc.).
- Une feuille de style CSS, définissant la mise en page de ces éléments (polices et tailles de caractères, bordures, marges, couleurs, positionnement dans la page, etc.).



HTML 5 LANGUAGE DE BALISAGE



- **Une balise** : est une unité syntaxique qui permet de délimiter une partie du texte.
- Elle indique la nature du texte qu'elle encadre.
- Exemple :



ÉLÉMENT HTML

- est la partie du document délimitée par les balises.
- La Balise d'ouverture marquera le début d'un élément, tel que `<div>`. La balise de clôture marquera la fin d'un élément et commencera par une barre oblique, comme `</div>`.



Exemple : élément HTML titre

`<title>` Ceci est un titre du document `</title>`



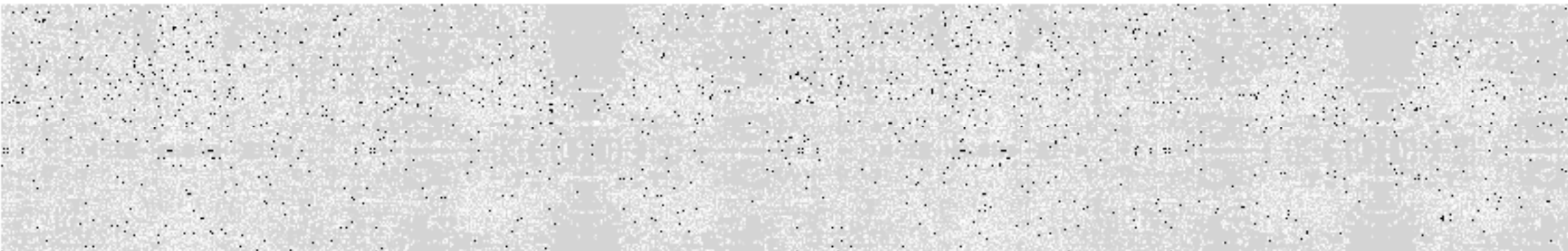
ATTRIBUT

- est une information auxiliaire définissant une propriété de l'élément HTML.





LES RÈGLES DE BASE HTML 5



Un document HTML doit respecter certaines règles simples:

- Les éléments et les attributs sont insensibles à la casse. Par exemple, `<body>` et `<BODY>` sont acceptés.
- Les éléments non vides doivent avoir une balise d'ouverture et une balise de fermeture.
- Les éléments vides ne comportent qu'une seule balise et il est plus correct de les terminer par les caractères `/>`



- Les éléments ne doivent pas se chevaucher et donc obéir au principe premier ouvert, dernier fermé.
- Tous les attributs doivent avoir une valeur incluse entre guillemets ("). Les différents attributs du même élément doivent être séparés par au moins un espace.
- Une valeur doit être donnée à tous les attributs utilisés, y compris à ceux dont la valeur est unique



Un document respectant toutes ces conditions est dit « *bien formé* »

- Un document bien formé est dit « *conforme* » s'il respecte les règles d'inclusion des éléments les uns dans les autres, telles qu'elles sont définies dans la spécification HTML 5.
- Le respect de ces contraintes est impératif pour que le document soit déclaré conforme par le validateur.



STRUCTURE D'UN DOCUMENT HTML 5

- Toute page écrite en HTML5 doit comporter une certaine structure, un « squelette » qui sera toujours le même.

```
1 <!DOCTYPE html>
2   <html>
3     <head>
4       <title>Ma première page valide en HTML5</title>
5       <meta charset="utf-8"/>
6     </head>
7     <body>
8
9     </body>
10  </html>
11
```



Structure d'une page HTML 5

`<!DOCTYPE html>` ← Précise qu'il s'agit d'un document HTML 5
(obligatoire)

`<!-- commentaire -->` ← Commentaire (optionnel)

`<html>` ← La racine du document contient
obligatoirement deux fils **head** et **body**

```
<head>
  <title> Titre de la page </title>
  .....
</head>
<body>
  .....
</body>
</html>
```

Structure d'une page HTML 5

```
<!DOCTYPE html>
```

```
<!-- commentaire -->
```

```
<html lang="fr" dir="ltr">
```

```
<head>
```

```
  <title> Titre de la page </title>
```

```
  .....
```

```
</head>
```

L'entête : Fournit des renseignements sur la page

```
<body>
```

```
  .....
```

```
</body>
```

Le corps : représente le contenu de la page (texte, image, tableau,...)



- **Doctype** : sert à indiquer le type de balisage de document. En HTML5 on met simplement html, sans oublier le point d'exclamation avant le Doctype
-
- **Html** : l'élément **<html>** est parent de tous les autres éléments, soit **<head>** et **<body>**. Il possède des attributs comme **lang** qui indique la langue utilisée par défaut dans la page, et **dir** pour préciser le sens de lecture (**dir = 'ltr'** → de gauche à droite et **dir = 'rtl'** de droite à gauche).
-



- **Head** : contient le titre de la page, et des méta-données (données invisibles par l'utilisateur)

```
<head>  
  <title> Titre de la page </title> ← Le titre de la page (obligatoire)  
  .....  
</head>
```



Structure d'une page HTML 5

```
<!DOCTYPE html>
<html>
  <head>
    <title> Page HTML 5 minimale </title>
  </head>
  <body>
    <p> salut ! <br> C'est ma première page</p>
  </body>
</html>
```



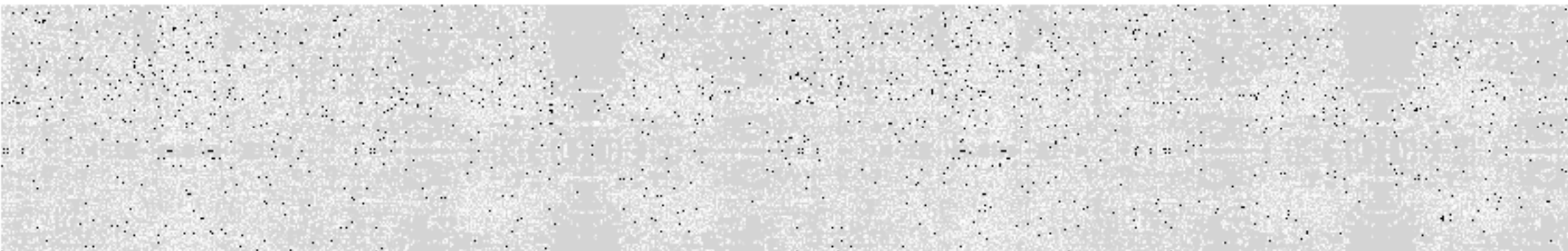
EXEMPLE

Voici une
première page
web avec le
minimum de
balises,





EN TÊTE DE LA PAGE



En-tête de la page

<head>

<title> Page HTML simple </title>

<meta Charset="UTF-8">

</head>

cette balise **META** permet d'avoir une information sur l'encodage des caractères utilisés (**UTF-8**)

il est recommander de la placer au début de la section **<head>**, surtout avant **<title>**,

En-tête de la page

```
<head>
```

```
<meta Charset="UTF-8">
```

```
<title> Page HTML simple </title>
```

```
<meta name=" keywords" content="Web, HTML5, CSS">
```

```
</head>
```

Des mots clés pour l'indexation de la page

La balise META keywords, permet d'indiquer un certain nombre de mots-clés à associer à la page

En-tête de la page

```
<head>
```

```
<meta Charset="UTF-8">
```

```
<title> Page HTML simple </title>
```

```
<meta name=" keywords" content="Web, HTML5, CSS3">
```

```
<meta name="description" content="description de ma page">
```

```
</head>
```



Description de la page affichée par les moteurs de recherche

Cette META permet de fournir une courte description de la page web. Cette description est notamment utilisée par les principaux moteurs de recherche, pour fournir le court texte d'aperçu des pages de résultats.

En-tête de la page

<head>

<meta Charset="UTF-8">

<title> Page HTML simple </title>

<meta name="keyword" content="Web , HTML5, CSS3">

<meta name="description" content="description de ma page">

<meta http-equiv="refresh" content="10">

</head>



Rafrachissement (Rechargement) de la page toutes
les 10 secondes

L'attribut *refresh* permet d'ordonner au navigateur de recharger la
page toutes les **n** secondes

En-tête de la page

<head>

<meta Charset="UTF-8">

<title> Page HTML simple </title>

<meta name="keywords " content="Web, HTML5, CSS3">

<meta name="description " content="description de ma page">

<meta http-equiv="refresh" content="10">

<meta name="Date" content=" Mon, 13 February 2017 10:00:00">



</head>

Date de création ou mise à jour de la page

Les attributs de < meta >

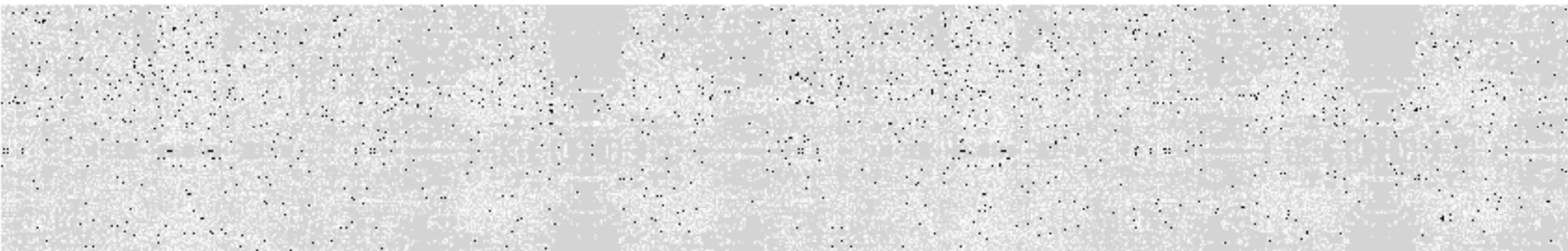
Attribut	Valeurs	Rôle
name	application-name author description generator keywords ou autres valeurs enregistrées	Méta-informations relatives à la page : nom de l'application web, auteur, description du contenu, programme de conception utilisé, mots-clés relatifs au contenu.
http-equiv	refresh default-style content-type	Rafraîchissement de la page. Feuille de style préférée. Déclaration de la page de code et du type MIME.
content	texte	En combinaison avec <code>name</code> et <code>http-equiv</code> , confère sa valeur à la balise.
charset	encodage des caractères	Déclaration de la page de code qui doit être utilisée pour l'interprétation, si celle-ci est différente de l'ASCII.





LE CORPS DE LA PAGE

Les titres



Les Titres

On doit utiliser six niveaux de titres différents.

Pourquoi?

Parce que, dans un document, on peut dire « ceci est un titre très important », « ceci est un titre un peu moins important », « ceci est un titre encore moins important », etc.

On a donc six (06) balises de titres différentes:

- `<h1> </h1>` : signifie « titre très important »
- `<h2> </h2>` : signifie « titre important »
- `<h3> </h3>` : signifie « sous-titre » c'est un titre un peu moins important
- `<h4> </h4>` : signifie « titre moins important »
- `<h5> </h5>` : signifie « titre pas important »
- `<h6> </h6>` : alors là pas important du tout

Les Titres

Exemple :

`<Body>`

`<h1>` Titre du 1^{er} niveau `</h1>`

`<h2>` Titre du 2^{ème} niveau `</h2>`

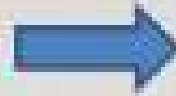
`<h3>` Titre du 3^{ème} niveau `</h3>`

`<h4>` Titre du 4^{ème} niveau `</h4>`

`<h5>` Titre du 5^{ème} niveau `</h5>`

`<h6>` Titre du 6^{ème} niveau `</h6>`

`</body>`



Titre du 1er niveau

Titre du 2ème niveau

Titre du 3ème niveau

Titre du 4ème niveau

Titre du 5ème niveau

Titre du 6ème niveau



LE CORPS DE LA PAGE

Mise en forme

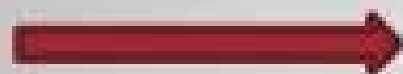
Corps de la page

Paragraphe

<body>

<p title="Description 1" > Premier paragraphe </p>

<p title = "Description 2" > Deuxième paragraphe </p>



Premier paragraphe

Deuxième paragraphe

Corps de la page

Paragraphe

<body>

<p title="Description 1" > Premier paragraphe </p>

<p title = "Description 2" > Deuxième paragraphe </p>



Premier paragraphe

Description 1

Deuxième paragraphe

Corps de la page

Paragraphe

<body>

<p title="Description 1" > Premier paragraphe </p>

<p title = "Description 2" > Deuxième paragraphe </p>



Premier paragraphe

Deuxième paragraphe

Description 2

Corps de la page

Texte en gras

`` Ceci est un texte en gras ``



Ceci est un texte en gras.

Texte en italique

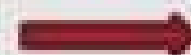
`<i>` Ceci est un texte en italique `</i>`



Ceci est un texte en italique.


Texte en petite taille

`<p>` Ceci est un texte `<small>` petit `</small>` , `</p>`



Ceci est un texte petit.

Texte en indice

`<p> la variable X _i .</p>`  La variable X_i .

Texte marqué

`<p> <mark> Remarque : </mark> </p>`  Remarque:

Alignement du texte (attribut align):

`<p align="center"> Paragraphe avec texte centré </p>`

Remarque : l'attribut **align** peut avoir les valeurs **center**, **right**, **left**.



STRONG, EMPHASIS ET MARK

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Ma premiere page en HTML</title>
5     <meta charset="utf-8"/>
6   </head>
7   <body>
8     <h1>Cette fois, on va parler des éléments strong, em et mark</h1>
9     <p>Je suis un paragraphe contenant un mot très <strong>important</strong></p>
10    <p>Moi aussi je contiens <em>trois mots importants</em>, mais un peu moins imp
11    <p>Moi, je possède un <mark>contenu qui doit ressortir visuellement</mark></p>
12  </body>
13 </html>
```

Ma premiere page en HTML

file:///Users/Pham/Desktop/Cours%20HTML/1pre.html

Cette fois, on va parler des éléments strong, em et mark

Je suis un paragraphe contenant un mot très **important**

Moi aussi je contiens *trois mots importants*, mais un peu moins importants tout de même

Moi, je possède un **contenu qui doit ressortir visuellement**



LES LISTES

Les listes ordonnées, non
ordonnées, liste de définition

- Les listes HTML sont prévues pour dresser des énumérations possédant une valeur sémantique.
- Les balises , , <DL> déterminent 3 type des listes (ordonnée, non ordonnée et définition de liste).
- Associée à ces balises, une autre balise ... précise quels sont les différents éléments de la liste.



Les listes

Liste non ordonnée (à puces): Élément HTML **ul** + **li**

<p>

 Plan du cours :

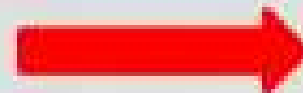
 Introduction au Web

 HTML 5

 CSS 3

 JavaScript

</p>

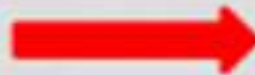


Plan du cours:

- Introduction au Web
- HTML 5
- CSS 3
- JavaScript
-

LISTE ORDONNÉE

```
<p>  
  <b> Plan du cours : </b>  
  <ol>  
    <li> Introduction au Web </li>  
    <li> HTML 5 </li>  
    <li> CSS 3</li>  
    <li> JavaScript </li>  
    <li> ..... </li>  
  </ol>  
</p>
```



Plan du cours:

Cliquez pour ajouter du texte

1. Introduction au Web
2. HTML 5
3. CSS 3
4. JavaScript
5. ...



Les listes

Paramètres d'une liste ordonnée (numérotée)

➡ Le type de la numérotation : Attribut *type*

- *type="1"*: 1,2,3,... (Numérotation décimale)
- *type="a"*: a, b, c, ... (Numérotation alphabétique minuscule)
- *type="A"*: A, B, C, ... (Numérotation alphabétique majuscule)
- *type="i"*: i, ii, iii, ... (Numérotation en chiffres romains minuscules)
- *type="I"*: I, II, III, ... (Numérotation en chiffres romains majuscules)

➡ La valeur initiale de la numérotation : Attribut *start*

➡ Une numérotation descendante : Attribut booléen *reversed*

Les listes

Paramètres d'une liste ordonnée (numérotée)

Exemple

<p>

 Plan du cours :

<ol type="I" start="3">

 Introduction au Web

 HTML 5

 CSS 3

 JavaScript

</p>



Plan du cours :

III. Introduction au Web

IV. HTML 5

V. CSS 3

VI. JavaScript

VII.

Les listes

Paramètres d'une liste ordonnée (numérotée)

Exemple

<p>

 Plan du cours :

<ol type="I" reversed="5">

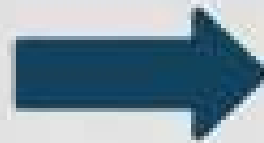
 Introduction au Web

 HTML 5

 CSS 3

 JavaScript

</p>



Plan du cours :

- V. Introduction au Web
- IV. HTML 5
- III CSS 3
- II JavaScript
- I

LISTES IMBRIQUÉES

Texte 1

 texte Niveau 1

 Texte Niveau 1.1

 Texte Niveau 1.2

 texte Niveau 2

Texte 3

Texte 1

1. texte Niveau 1

1. Texte Niveau 1.1

2. Texte Niveau 1.2

2. texte Niveau 2

Texte 3



```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>L'univers merveilleux des listes</title>
5     <meta charset="utf-8"/>
6   </head>
7   <body>
8     <h1>Les listes imbriquées</h1>
9     <ol>
10      <li>Natation
11        <ul>
12          <li>Maillot</li>
13          <li>Bonnet</li>
14          <li>Lunettes</li>
15        </ul>
16      </li>
17      <li>Vélo</li>
18      <li>Course à pied</li>
19    </ol>
20  </body>
21 </html>

```

EXEMPLE 2

Les listes imbriquées

1. Natation
 - Maillot
 - Bonnet
 - Lunettes
2. Vélo
3. Course à pied



REMARQUES IMPORTANTES

- On ne peut pas mettre une liste à puce à l'intérieur d'un paragraphe <P>.
- La balise OL ou UL ne peut avoir comme enfant direct que la balise Li,

Par exemple: on ne peut pas mettre la balise P comme enfant direct de la balise OL ou UL.

Par contre, on peut mettre P à l'intérieur du Li.





LE CORPS DE LA PAGE

Les listes de définition

Les listes

Listes de définitions : Éléments HTML **dl** + **dt** + **dd**

La liste de définition permet de présenter un glossaire, un lexique ou un dictionnaire.

Ces listes comportent deux parties , un Terme et une Description.

- **dl** : la liste de définitions (conteneur)
- **dt** : Le terme à définir (terme de définition)
- **dd** : La définition du terme (description de définition)

Les listes

Listes de définitions : Éléments HTML *dl + dt + dd*

- Exemple :

```
<p>  
  <b> Quelques définitions </b>  
  <br>  
  <dl>  
    <dt> HTML </dt>  
    <dd> Langage de création de pages Web </dd>  
    <dt> CSS </dt>  
    <dd> Langage de décoration de pages Web </dd>  
  </dl>  
</p>
```


Les listes

Listes de définitions : Éléments HTML **dl** + **dt** + **dd**

- Exemple :

```
<p>  
  <b> Quelques définitions </b>  
  <br>  
  <dl>  
    <dt> HTML </dt>  
    <dd> Langage de création de HTML  
    <dt> CSS </dt>  
    <dd> Langage de décoration CSS  
  </dl>  
</p>
```

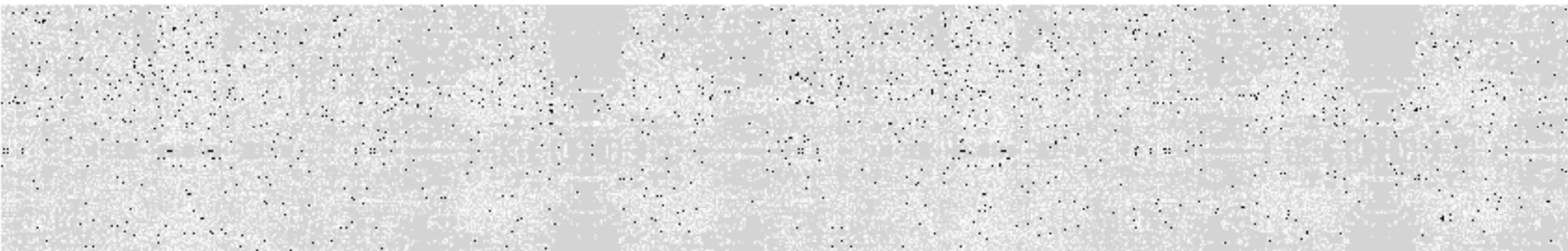
Quelques définitions

HTML	Langage de création de pages Web
CSS	Langage de décoration de pages Web



LE CORPS DE LA PAGE

Les liens



Les liens

<a>: Définit un hyperlien dont l'adresse de destination est définie dans l'attribut **href**.

Exemple :

```
<a href = "http://www.usthb.dz/"> USTHB </a>  
USTHB
```



Les liens peuvent être :

✓ relatifs ou absolus



Lien absolu, lien relatif

- Le lien peut être *absolu*, c'est-à-dire la destination du lien est formulée sous forme d'une **adresse complète**, notamment dans le cas d'un lien menant vers un autre site hébergé sur un autre nom de domaine.
- Il peut être aussi *relatif*, si l'on souhaite se baser sur **l'adresse courante – affichée dans la barre d'adresses** – et faire référence à un document stocké au même niveau, ou dans un niveau différent (sous-répertoire, répertoire parent, répertoire racine).

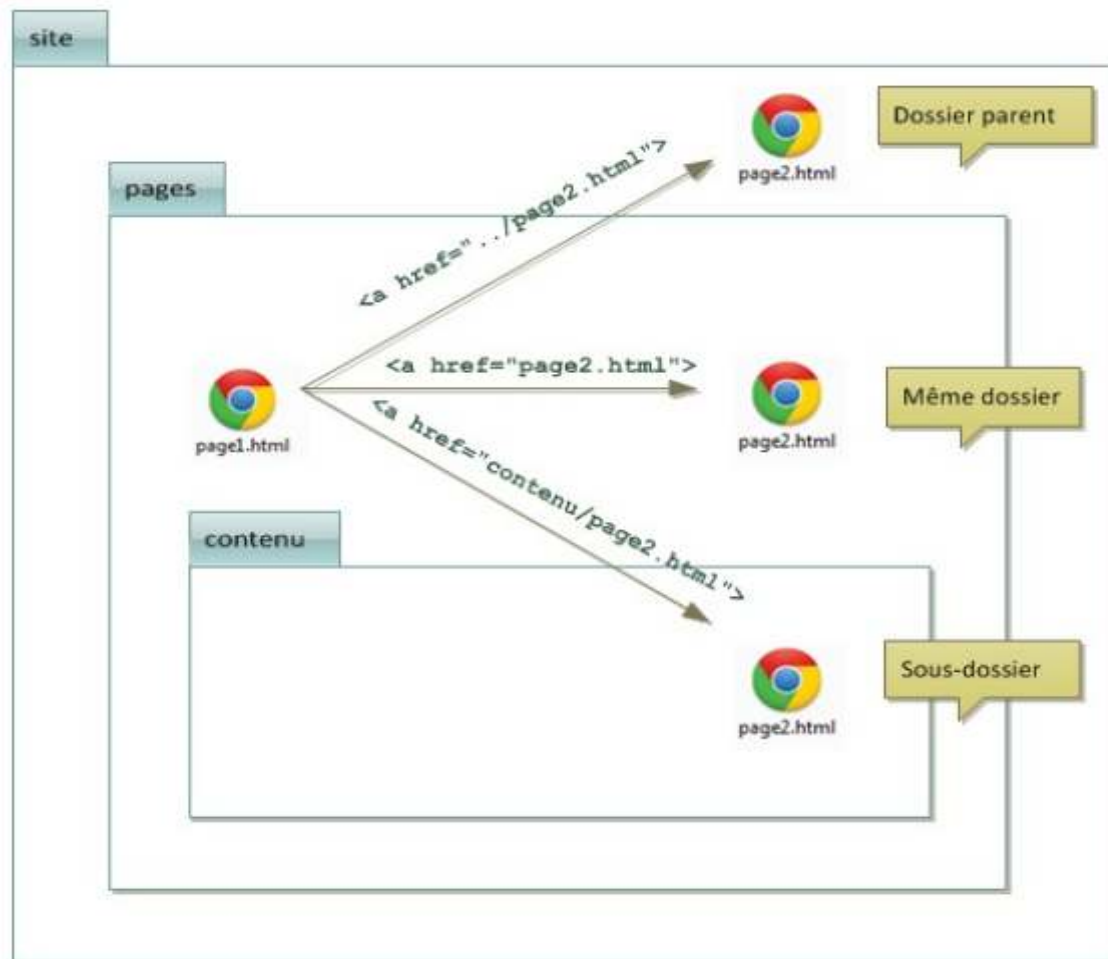


PRIVILÉGIER L'ADRESSE RELATIVE

- Lors de la création d'un site destiné à un seul serveur, tous les liens internes doivent être sous forme relative
- Si les liens internes sont sous forme absolue, ils seront perdus après le transfert : les noms de répertoires ne seront peut-être pas les mêmes sur le serveur !
- Utiliser une adresse relative permet au routeur de ne pas avoir à résoudre à chaque requête le chemin vers le serveur. Cette opération prend du temps, rendant ainsi la page plus longue à charger.



Résumé



Les différents liens



MEILLEURE SOLUTION

- L'idéal est de donner le chemin du fichier par rapport à la racine du site et non pas par rapport au document sur lequel on se trouve actuellement.
- Pour le faire, on commence le chemin par une barre oblique
/Dossier1/Dossier2/..../Fichier.ext



Les Liens

Il est facile de reconnaître un lien sur une page:

Exemple de lien



On peut faire des liens :

- d'une page vers une autre page.
- vers un autre site.



→ liens externes → le même fonctionnement

- vers une ancre (au sein de la même page) → liens internes

Les Liens

Liens externes

➤ Lien vers un autre site

` Aller à Google ` ➡ [Aller à Google](http://www.google.dz)

➤ Lien vers une autre page

- Avec un lien relatif : `Exemple de lien`

- Avec un lien absolu :

`Exemple de lien`

Les Liens

Lien vers une ancre

Une ancre est une sorte de point de repère qu'on peut mettre dans les pages HTML lorsqu'elles sont très longues.

Pour créer une ancre, il suffit de rajouter l'attribut identifiant « **id** » à une balise qui va servir de repère.

Utilisons l'attribut « **id** » pour donner un nom à l'ancre .

**LES SIGNETS
" LIEN VERS UNE
ANCHE "**



Les Liens

Exemple:

```
<p id="p_1">  
<a href="#p_10">Aller au paragraphe 10</a>  
</p>  
...  
...  
...  
...  
...  
...  
<p id="p_10">  
<a href="#p_1">Aller au premier paragraphe</a>  
</p>
```

Les Liens

Lien vers une ancre située dans une autre page :

Comment créer cette ancre ?

L'idée est de faire un lien qui ouvre une autre page ET qui amène directement à une ancre sur cette page .

`` Titre du paragraphe **``**

Les Liens

Exemple : soit la page 1 html suivante :

```
<!DOCTYPE html>
<html>
  <head>
    <title>  les liens </title>
  </head>
  <body>

    <p> Listes des différents LOISIRS <br><br>
    <a href="page2.html#Natation"> Natation </a><br>
    <a href="page2.html#Voyage"> Voyage  </a><br>
    <a href="page2.html#Musique"> Musique </a><br>
    <a href="page2.html#Dessin"> Dessin </a><br>
    </p>

  </body>
</html>
```

Les Liens

Exemple : soit la page 1 html suivante :

```
<!DOCTYPE html>
<html>
  <head>
    <title> les liens </title>
  </head>
  <body>

    <p> Listes des différents LOISIRS
    <a href="page2.html#Natation"> Na
    <a href="page2.html#Voyage"> Voya
    <a href="page2.html#Musique"> Mus
    <a href="page2.html#Dessin"> Dess
    </p>

  </body>
</html>
```

Listes des différents LOISIRS

Natation

Voyage

Musique

Dessin

Les Liens

Et soit la 2^{ème} page HTML suivante :

```
<!DOCTYPE html>
<html>
  <head>
    <title>   page 2 </title>
  </head>
  <body>
    <p id="Natation"> Natation : <br>
    .....( beaucoup de texte).....<br>
    .....<br>
  </p>
  <p id="Voyage"> Voyage : <br>
  .....( beaucoup de texte).....<br>
  .....<br>
  <br>
  </p>
  <p id="Musique"> Musique : <br>
  .....( beaucoup de texte).....<br>
  .....<br>
  </p>
  <p id="Dessin"> Dessin : <br>
  .....( beaucoup de texte).....<br>
  .....<br>
  </p>
</body>
</HTML>
```


Les Liens

Listes des différents LOISIRS

[Natation](#)

[Voyage](#)

[Musique](#)

[Dessin](#)



Natation :

.....(beaucoup de texte).....

.....

Voyage :

.....(beaucoup de texte).....

.....

Musique :

.....(beaucoup de texte).....

.....

Dessin :

.....(beaucoup de texte).....

.....

Les Liens

Lien vers le début de la page :

```
<p>  
<a href="#"> Aller au début de la page </a>  
</p>
```

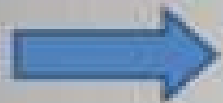
Ouvrir un lien dans un nouvel onglet :

```
<p>  
<a href="index.html" target="_blank">Lien</a>  
</p>
```

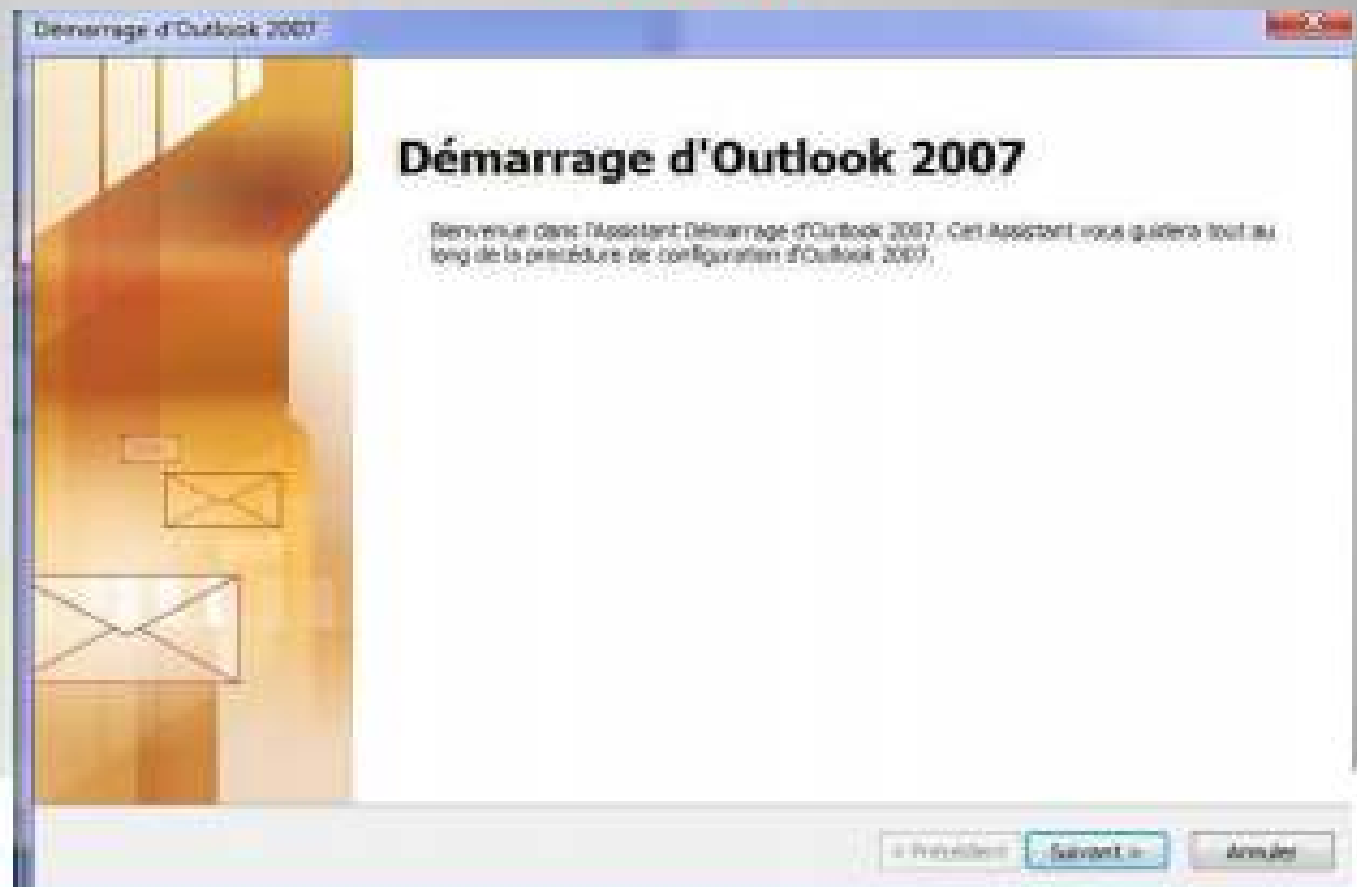
Les Liens

Lien vers une adresse e-mail :

`<p> Contactez-nous </p>`



Contactez-nous



Questions?

Comment créer un lien vers un fichier PDF?

Comment créer un lien vers un fichier PDF
dans un **nouvel onglet**?

Comment créer un lien vers **une page** en
cliquant sur une **image**?



Les Liens

Lien de téléchargement :

```
<p> <a href="C:/Users/digitec/Desktop/DAW/cours2.pdf">Télécharger</a>  
</p>
```

➡ Le fichier « cours2.pdf » sera ouvert dans le même onglet que la page appelante

```
<p> <a href="C:/Users/digitec/Desktop/DAW/cours2.pdf" target="_blank" >  
    Télécharger </a>  
</p>
```

➡ Le fichier « cours2.pdf » sera ouvert dans un autre onglet

Lien sous forme d'image :

```
<p>  
<a href="page1.html">  </a>  
</p>
```

LES IMAGES



- La balise **** permet d'intégrer une image dans une page web.
- Les images sont **liées** à des pages web.
- La balise **** est vide, elle contient uniquement des attributs et n'a pas de balise de fermeture.
- Elle a deux attributs obligatoires:
 - - **src**: spécifie le chemin d'accès à l'image (URL)
 - - **alt**: spécifie un texte alternatif pour l'image

```

```



Remarque:

- Il faut s'assurer que l'image reste au même endroit par rapport à la page web
- si l'utilisateur pour une raison quelconque ne peut pas la visualiser (en raison d'une connexion lente, d'une erreur dans l'attribut src ... le navigateur affichera la valeur de l'attribut **alt** avec l'icone du lien brisé..
- l'idéal est de modifier l'image sur Photoshop et l'utiliser directement avec sa taille naturelle sans width et height.



- Image en tant que lien
- Pour le faire placer la balise à l'intérieur de la balise <a>
- ```

 <img src=« photo.gif" alt=« photos de vacances"
 style="width:42px;height:42px;">

```



# FORMATS D'IMAGE PRIS EN CHARGE

Les types de fichiers image les plus courants, qui sont pris en charge dans tous les navigateurs (chrome, edge, firefox, safari, opera)

|      |                                        |                               |
|------|----------------------------------------|-------------------------------|
| APNG | Animated Portable Network Graphics     | .apng                         |
| GIF  | Graphics Interchange Format            | .gif                          |
| ICO  | Microsoft Icon                         | .ico, .cur                    |
| JPEG | Joint Photographic Experts Group Image | .jpg, .jpeg, .jif, .jpe, .jfi |
| PNG  | Portable Network Graphics              | .png                          |
| SVG  | Scalable Vector Graphics               | .svg                          |



# Les figures

- Les figures peuvent être de différents types : des illustrations, des diagrammes, des photos, des listes de codes, des citations, etc.
- Si une image illustre le texte (et n'est pas seulement décorative), il est conseillé de la placer au sein d'une balise **<Figure>**. La balise **<figurecaption>** permet d'écrire la légende de l'image.
- Exemple

```
<figure>

 <figcaption>Logos des différents navigateurs</figcaption>
</figure>
```



- `<figure>`  
    ``  
    `<figcaption>Fig.1 - Trulli, Puglia, Italy.</figcaption>`  
`</figure>`



# LES ÉLÉMENTS MULTIMÉDIAS



# AUDIO

**<audio controls>**

**<source** src=" \chemin par rapport à la racine du site\musique.mp3 " >

**<source** src=« test.mp3" type="audio/mpeg">

**<p>** votre navigateur ne supporte pas l'audio **</P>**

**</audio>**



On peut aussi:

**<audio controls src**=" \chemin par rapport à la racine du site\musique.mp3 " >

**<p>** votre navigateur ne supporte pas l'audio **</P>**

**</audio>**



# Audio

- Le texte entre les balises `<audio>` et `</audio>` ne sera affiché que dans les navigateurs qui ne prennent pas en charge l' `<audio>`élément.
- L'élément `<source>` permet de spécifier des fichiers audio alternatifs parmi lesquels le navigateur peut choisir. Le navigateur utilisera le premier format reconnu.
- Trois formats d'audio sont pris en charge: MP3, WAV, Ogg

## Quelques attributs:

- **Controls**: attribut booléen, le fait de le mettre seulement invite le navigateur à afficher les boutons de contrôle de lecteur audio.
- **Autoplay**: démarre le fichier audio automatiquement.
- **Loop**: lecture en boucle.



# vidéo

```
<video width="320" height="240" controls>
```

```
<source src="movie.mp4" type="video/mp4">
```

```
<source src="movie.ogv" type="video/ogg">
```

```
<P> votre navigateur ne supporte pas l'élément Video. </P>
```

```
</video>
```

- L'élément Video fonctionne de la même façon que l'élément Audio
- Trois formats de vidéos sont pris en charge: MP4, WebM, Ogg
- Vous pouvez choisir une image représentative de la vidéo qui sera affichée à l'aide de l'attribut poster:

```
<video src="film.mp4" controls width="360" height="240" poster="aperçu.jpg">
</video>
```



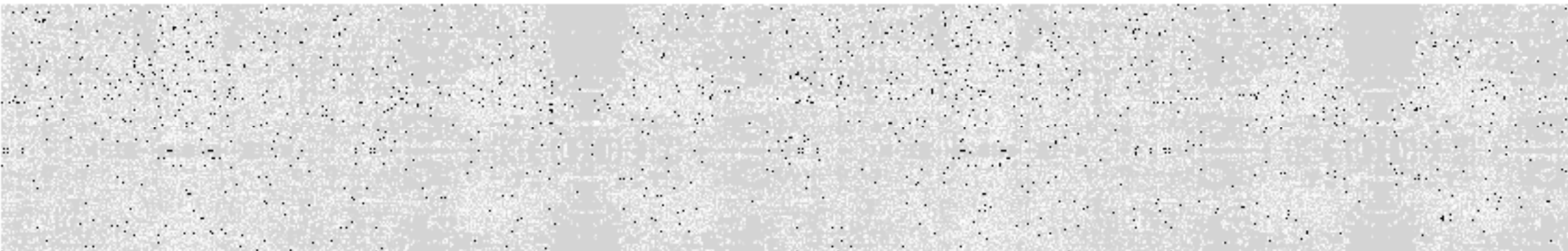


- Pour insérer une vidéo depuis Youtube on utilise l'élément **iframe**.
- Procéder comme suit:
- aller sur la vidéo youtube -> cliquer sur **partager** -> cliquer sur **Intégrer** -> copier et coller le code avec la balise iframe -> supprimer les attributs en plus





# **LES ÉLÉMENTS BLOC ET EN LIGNE**



- Les différentes balises que l'on peut mettre dans le corps de la page peuvent être classées en deux groupes : balises de type block et balises de types inline.



# ÉLÉMENTS DE TYPE BLOCK

- Un retour à la ligne est inséré avant et après chaque élément créant ainsi un bloc de contenu (commence toujours sur une nouvelle ligne)
- Ils prennent la largeur de leurs conteneurs
- a une marge supérieure et inférieure.
- Les balises suivantes sont de type block :
- H1..H6, p, li, ul, ol, dl, dt, div, article, form,....etc



# ÉLÉMENTS DE TYPE INLINE

- ne commence pas sur une nouvelle ligne.
- n'occupe que la largeur nécessaire.
- Les principales balises de type inline sont: br, em, strong, img et a.



# REMARQUE

- Un élément en ligne ne peut pas contenir un élément de niveau bloc !



# Les éléments de groupement DIV et Span

Les balises de groupement sont avant tout destinées à en grouper d'autres:

- **L'élément <Div>:** Souvent utilisé comme conteneur en bloc pour d'autres éléments HTML
- **<span>:** À la différence de <div>, span est un conteneur en ligne spécifiquement dédié au contenu phrasé. (une partie d'un texte ou une partie d'un document)



# SÉMANTIQUE HTML



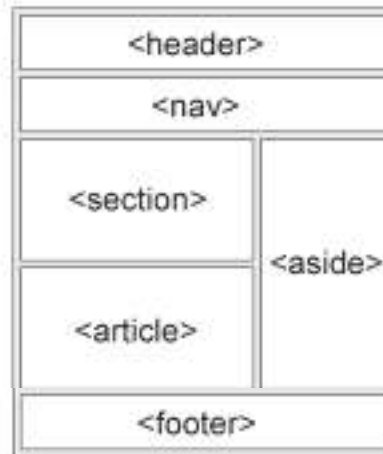


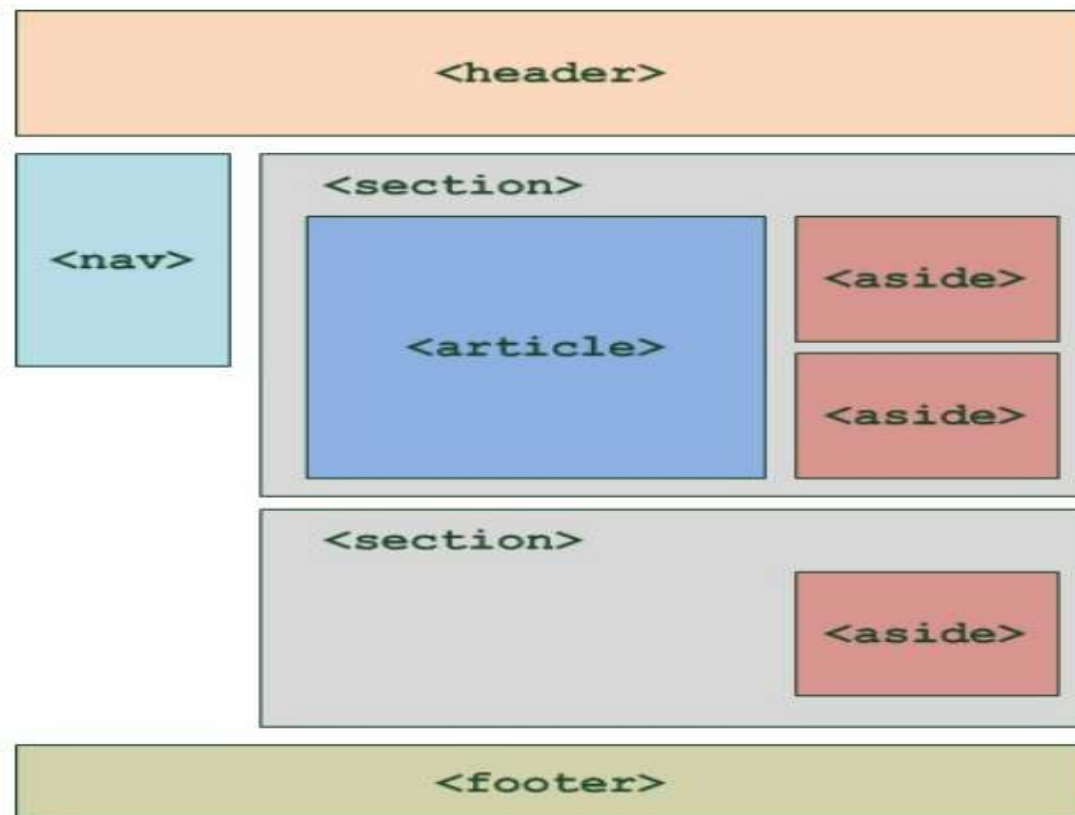
- De nombreux sites Web contiennent du code HTML tel que :
- `<div id="nav">`
- `<div class="header">`
- `<div id="footer">` pour indiquer la navigation, l'en-tête et le pied de page
- Un élément sémantique décrit clairement sa signification à la fois pour le navigateur et le développeur.
- Exemples d'éléments non sémantiques : `<div>` et `<span>`- Ne dit rien sur son contenu.
- Exemples d'éléments sémantiques : `<form>`, `<table>`, et `<article>`- Définit clairement son contenu.



En HTML, certains éléments sémantiques peuvent être utilisés pour définir différentes parties d'une page Web :

- <article>
- <aside>
- <details>
- <figcaption>
- <figure>
- <footer>
- <header>
- <main>
- <mark>
- <nav>
- <section>
- <summary>
- <time>





# ÉLÈMENT SECTION

- Selon la documentation HTML du W3C : « Une section est un regroupement thématique de contenu, généralement avec un titre.
- Exemples d'utilisation de l'élément `<section>` :
  - Chapitres
  - introduction
  - Informations
  - Informations de contact
- Une page Web peut normalement être divisée en sections pour l'introduction, le contenu et les informations de contact.



# EXAMPLE:

- Deux sections dans un document
- `<section>`
- `<h1>Lorem</h1>`
- `<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Natus officiis,`
  - `inventore similique tempore earum quasi ipsa iste iusto eius labore blanditiis`
- `.</p>`
- `</section>`
- `<section>`
- `<h1>Ipsum</h1>`
- `<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Natus officiis,`
  - `inventore similique tempore earum quasi ipsa iste iusto eius labore blanditiis`
- `.</p>`
- `</section>`



# ELÉMENT ARTICLE

- Un article doit avoir un sens en soi et il doit être possible de le diffuser indépendamment du reste du site Web.
- C'est un contenu indépendant et autonome
- Par exemple, on peut utiliser l'élément Article dans:
  - Messages du forum
  - Articles de blog
  - Commentaires des utilisateurs
  - Fiches produits
  - Articles de journaux



# EXAMPLE

- Trois articles au contenu independant et autonome
- `<article>`  
`<h2>Google Chrome</h2>`  
`<p>Google Chrome is a web browser developed by Google, released in 2008. Chrome is the world's most popular web browser today!</p>`  
`</article>`  
  
`<article>`  
`<h2>Mozilla Firefox</h2>`  
`<p>Mozilla Firefox is an open-source web browser developed by Mozilla. Firefox has been the second most popular web browser since January, 2018.</p>`  
`</article>`  
  
`<article>`  
`<h2>Microsoft Edge</h2>`  
`<p>Microsoft Edge is a web browser developed by Microsoft, released in 2015. Microsoft Edge replaced Internet Explorer.</p>`  
`</article>`



# IMBRIQUER ARTICLE DANS SECTION ET L'INVERSE

`<article>` spécifie un contenu indépendant et autonome.

`<section>` définit la section d'un document.

- vous trouverez des pages HTML avec des `<section>` contenant des `<article>` et des `<article>` contenant des `<section>`.





# ELÉMENT HEADER

- `<header>` représente un conteneur pour le contenu d'introduction ou un ensemble de liens de navigation.
- Un `<header>` contient généralement :
  - un ou plusieurs éléments de titre (`<h1>` - `<h6>`)
  - logo ou icône
  - informations sur la paternité
- Remarque : Vous pouvez avoir plusieurs `<header>` dans un même document HTML. Cependant, `<header>` ne peut pas être placé dans un `<footer>`, `<address>` ou un autre `<header>`



# ELÉMENT FOOTER

- `<footer>` définit un pied de page pour un document ou une section.
- Un `<footer>` contient généralement :
  - Informations sur le droit d'auteur
  - Informations de contact
  - plan du site
  - liens de retour en haut
  - documents connexes
- Vous pouvez avoir plusieurs `<footer>` dans un même document.



# EXAMPLE

- `<footer>`  
    `<p>Author: Hege Refsnes</p>`  
    `<p><a href="mailto:hege@example.com">hege@example.com</a></p>`  
    `</footer>`



# ÉLÉMENT <NAV>

- <nav> définit un ensemble de liens de navigation.
- Il est destiné uniquement au **bloc majeur** de liens de navigation.

- Exemple

- <nav>  
    <a href="/html/">HTML</a> |  
    <a href="/css/">CSS</a> |  
    <a href="/js/">JavaScript</a> |  
    <a href="/jquery/">jQuery</a>  
  </nav>



# ÉLÉMENT <ASIDE>

- définit un contenu en dehors du contenu dans lequel il est placé (comme une barre latérale).
- Le <aside>contenu doit être indirectement lié au contenu environnant.
- Exemple: Affichez du contenu en plus du contenu dans lequel il est placé :
- <p>My family and I visited The Epcot center this summer. The weather was nice, and Epcot was amazing! I had a great summer together with my family!</p>

<aside>

<h4>Epcot Center</h4>

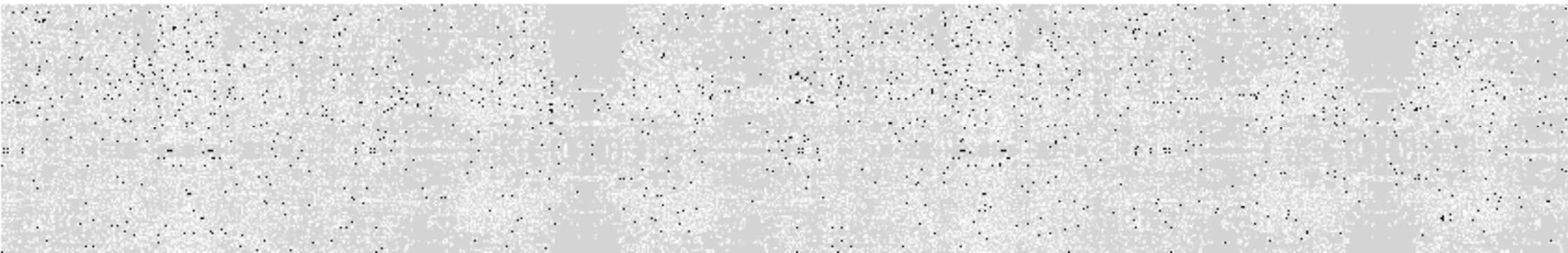
<p>Epcot is a theme park at Walt Disney World Resort featuring exciting attractions, international pavilions, award-winning fireworks and seasonal special events.</p>

</aside>





# COMMENT STRUCTURER MON SITE?



**<header>**

**<nav>**

**<section>**

**<article>**

**<article>**

**<article>**

**<aside>**

**<footer>**



`<header>`

`<nav>`

`<section>`

`<article>`

`<aside>`

`<aside>`

`<section>`

`<aside>`

`<footer>`



# Structurer la page

*En-tête de la page*

*zone de Navigation ( Menu )*

*Partie latérale*

*Contenu de la page*

*Pied de page*

## Structurer la page – Élément HTML 5 à utiliser

