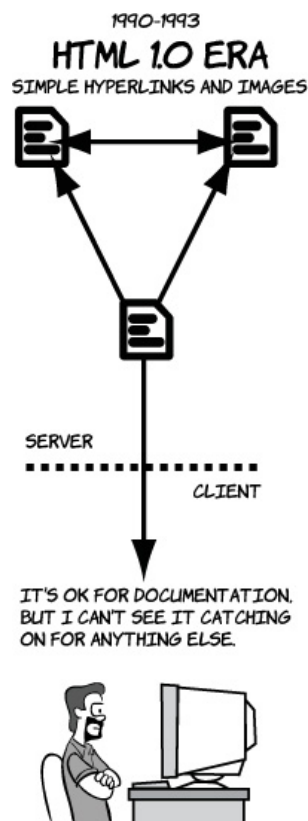


## HTML/CSS

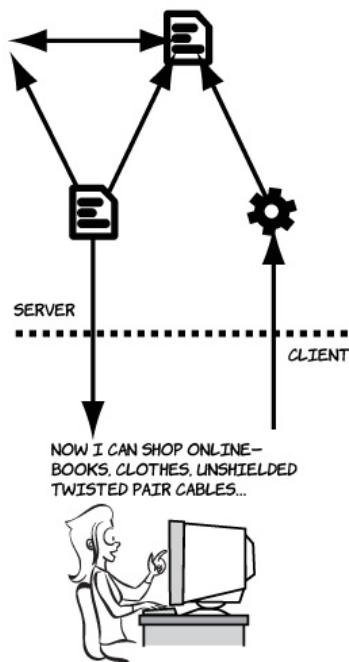
HTML est un langage de description de documents hypertextes. Les documents hypertextes sont constitués de titres, de paragraphes, de listes, d'images, etc... et, surtout, de liens vers d'autres documents hypertextes ; ce sont les liens qui constituent la partie hypertexte.

### Les différentes version HTML



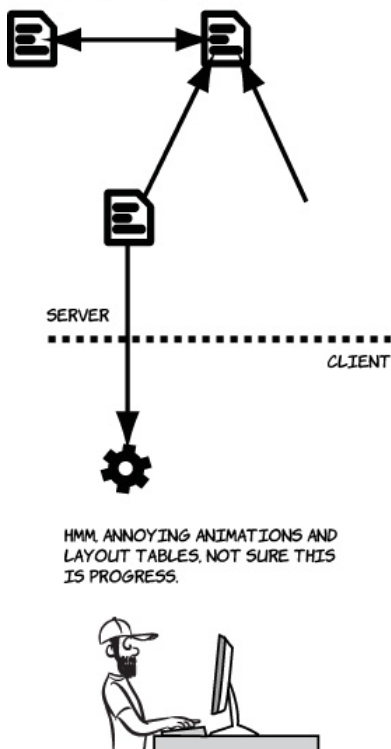
Lien sur des pages avec textes et images et liens vers d'autres pages. Aucune interaction à part les liens. HTML a été développé par Tim Berners-Lee au CERN de Genève pour faciliter la communication entre physiciens sur les accélérateurs de particules.

1993-1995  
**HTML 2.0 ERA**  
 INTRODUCING THE FORM ELEMENT



Possibilité d'utiliser la balise `<form>` pour entrer des données.

1995-1997  
**HTML 3.2 ERA**  
 CLIENT-SIDE INTERACTIVITY WITH JAVASCRIPT



Utilisation de javascript pour animer la page, ici des actions peuvent s'effectuer côté client mais problèmes de compatibilités dûs à la guerre des navigateurs et des différentes versions Javascript.

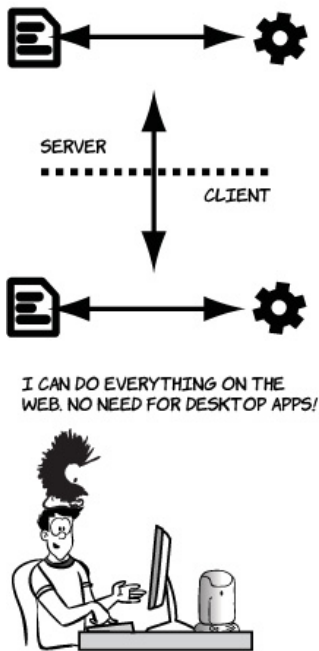
D'où la création du W3C<sup>1</sup>, consortium proposant de normaliser les besoins en publiant plusieurs « *drafts* » (brouillons), puis passent au stade « *Proposed Recommendations* » et un vote est proposé pour être repassé en *draft* ou passer en *Recommendation* sur lesquelles les éditeurs de logiciels devront se baser.

<sup>1</sup> World Wide Web Consortium - Tim Berners-Lee en est le dirigeant

1997-2010  
**HTML 4.0 ERA**  
 CLIENT-SIDE INTERACTIVITY  
 WITH JAVASCRIPT

## Normalisation Javascript ECMAScript 5,6,7

Les pages WEB sont de véritables application (SPA) grâce à la normalisation des moteurs Javascript et l'éco-système extrêmement foisonnant.



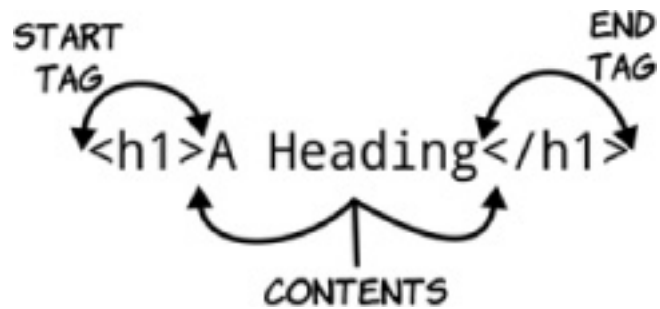
XHTML1.0 : HTML respecte les règles XML qui consiste à vérifier la cohérence de la page web à l'aide d'une « grammaire » DTD. ça permet de mieux détecter les erreurs mais ce régime est un peu draconien de plus IE ne gère pas correctement le XHTML.

L'élément `<!DOCTYPE>` indique que le document actuel appartient à un type spécifique de HTML

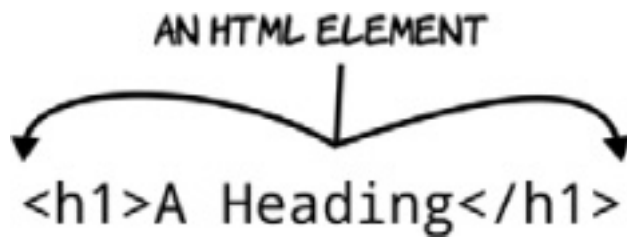
HTML5	<code>&lt;!DOCTYPE html&gt;</code>
HTML 4.01	<code>&lt;!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd";&gt;</code>
XHTML 1.0	<code>&lt;!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"&gt;</code>

## HTML4

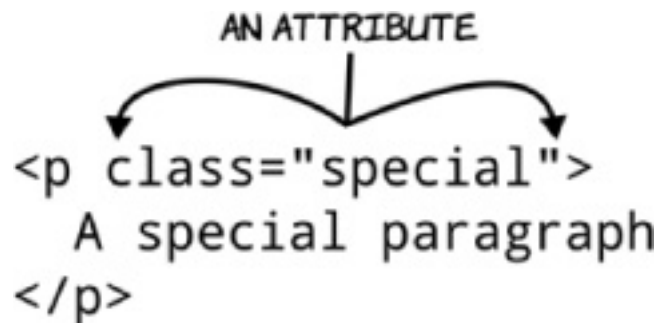
### LES BALISES (TAGs)



### HTML ELEMENT



### ATTRIBUT D'UN ELEMENT (ID , CLASS)



Les éléments peuvent contenir du texte mais aussi d'autres éléments . L'élément contenant étant le **parent**.

Testons nos scripts HTML avec l'éditeur ligne suivant

<https://fr.w3docs.com/tools/editor>

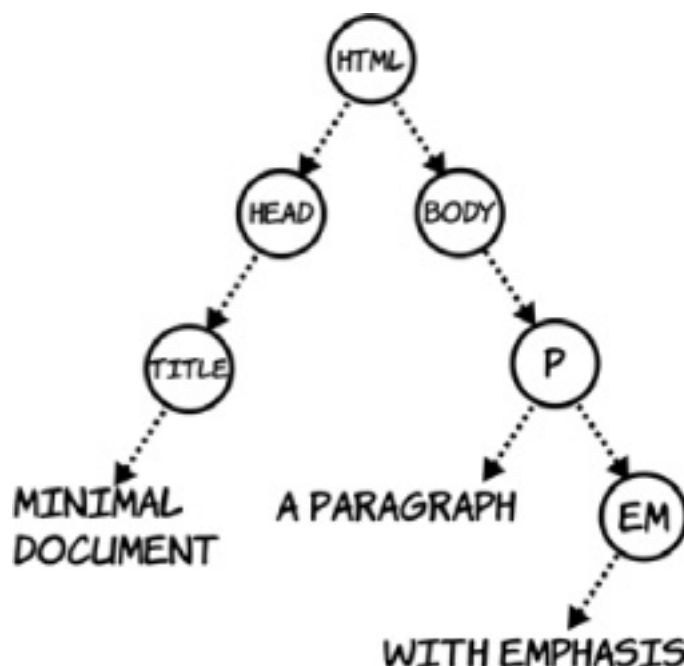
Un document HTML est un arbre partant du tag <html> avec 2 enfants <head> et <body>.

<head> pour les méta-données (metadata -> données à propos des données)  
exemple

<body> pour le contenu de la page

```
<html>
<head>
  <title>Minimal document</title>
</head>

<body>
<p>A paragraph
  <em>with emphasis</em> <-- italique
</p>
</body>
</html>
```



Une balise doit toujours être fermée par un slash <div>.....</div>

<HEAD>

<title> le titre qui apparait en haut du navigateur </title>

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<meta name="Author" content="J. Smith">
```

```
<meta name="keywords" lang="fr" content="AFPA, webdev, brest " >²
```

```
<meta name="description " lang="en" content="course, HTML" >
```

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

```
</head>
```

Les américains ont défini les caractères représentés par des nombres  
Ce sont les caractères ASCII

0 NUL	32 espace	64 @	96 `
1 SOH	33 !	65 A	97 a
2 STX	34 "	66 B	98 b
3 ETX	35 #	67 C	99 c
4 EOT	36 \$	68 D	100 d
5 ENQ	37 %	69 E	101 e
6 ACK	38 &	70 F	102 f
7 BEL	39 '	71 G	103 g
8 BS	40 (	72 H	104 h
9 HT	41 )	73 I	105 i
10 LF	42 *	74 J	106 j
11 VT	43 +	75 K	107 k
12 FF	44 ,	76 L	108 l
13 CR	45 -	77 M	109 m
14 SO	46 .	78 N	110 n
15 SI	47 /	79 O	111 o
16 SLE	48 0	80 P	112 p
17 CS1	49 1	81 Q	113 q
18 DC2	50 2	82 R	114 r
19 DC3	51 3	83 S	115 s
20 DC4	52 4	84 T	116 t
21 NAK	53 5	85 U	117 u
22 SYN	54 6	86 V	118 v
23 ETB	55 7	87 W	119 w
24 CAN	56 8	88 X	120 x
25 EM	57 9	89 Y	121 y
26 SIB	58 :	90 Z	122 z
27 ESC	59 ;	91 [	123 {
28 FS	60 <	92 \	124
29 GS	61 =	93 ]	125 }
30 RS	62 >	94 ^	126 ~
31 US	63 ?	95 _	127 ■

---

² Plus n  
AFPA 202

Le problème se pose pour les autres caractères dans le monde (accents français, n tilde pour l'espagnol ou le breton, et alphabets asiatiques

D'ou l'invention de Unicode.

Unicode prend 2 octets au lieu d 1 seul octet pour l'ASCII

UTF-8 permet d'utiliser l'ASCII quand c'est possible puis l'UNICODE si nécessaire.

Il existe des Entités HTML pour représenter certains caractères unicode mais c'est déconseillé car peut lisible, il vaut déclarer l'encodage de votre éditeur qui permet l'écriture Unicode (ou UTF-8).

&acute ;  
&nbsp ;  
&grave ;  
...etc...

## Base

créons un fichier index.html

```
<html>
  <body>
    mon texte
  </body>
</html>
```

Toujours après un tag ouverture => un tag de fermeture

Structure d'un page head

```
<html>
  <head>
    <title> Titre de notre page </title>
    <meta ... etc...
  </head>
  <body>
    mon texte
  </body>
</html>
```

Paragraphes et Titres dans <body>

```
<html>
  <head>
    <title> Titre de notre page </title>
  </head>
  <body>
    <h1>Titre de niveau 1</h1>
    <p>Paragraphe 1</p>
    <p>Paragraphe 2</p>
    <p>Paragraphe 3</p>
  </body>
</html>
```

Line break

```
<br/>
```



Lien

```
<a href="http://reddit.com" target="_blank">le site  
reddit.com</a>
```

## Entêtes et paragraphes

```
<html>
<head>
  <title>
    Headings and
    implicit structure
  </title>
</head>
<body>
  <h1>The main heading</h1>
  <p>Main introduction</p>
  <h2>First section</h2>
  <p>Section introduction</p>
  <h3>Subsection heading 1.1</h3>
  <p>Subsection 1.1</p>
  <h2>Second section</h2>
  <p>Section introduction</p>
  <h3>Subsection heading 2.1</h3>
  <p>Subsection 2.1</p>
  <h4>Sub-subsection</h4>
  <p>Subsection 2.1.1</p>
</body>
</html>
```

Listes:

non numérotées :

```
<ul>
  <li>List item</li>
  <li>List item</li>
  <li>List item</li>
</ul>
```

numérotées :

```
<ol>
  <li>List item</li>
  <li>List item</li>
  <li>List item</li>
</ol>
```

Balise Table :

```
<table>
  <caption>Passagers du vol 377</caption>

  <tr>
    <th>Nom</th>
    <th>Âge</th>
    <th>Pays</th>
  </tr>
  <tr>
    <td>Carmen</td>
    <td>33 ans</td>
    <td>Espagne</td>
  </tr>
  <tr>
    <td>Michelle</td>
    <td>26 ans</td>
    <td>États-Unis</td>
  </tr>
</table>
```

LES URL ( Universal Resource Locator)

Absolues

<http://www.google.fr>

Relatives :

Fichiers qui se trouvent soit à la racine de votre site Web , soit dans des répertoires et sous-répertoires à partir de la racine (répertoires descendants ou ascendants).

## Liens et ressources:

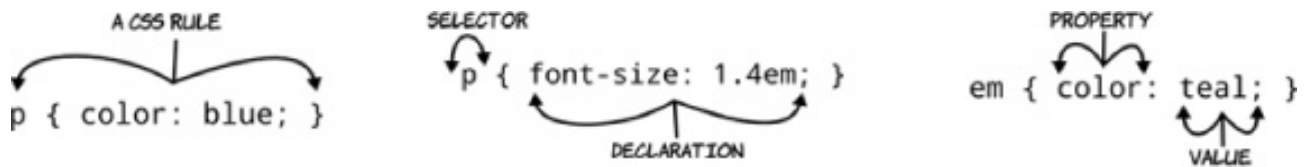
```
<p>Use
  <a href="http://www.google.com/">
    the Google
  </a>.
</p>
```

Permet de se connecter à un site web

```
<p>Go to
  <a href="pages/links-3.html">
    another page
  </a>.
</p>
```

Ou sur une page locale à partir de la racine du site.

## CSS



Une règles CSS (Cascade Style Sheet) est constituée d'un sélecteur, d'une liste de déclarations avec le séparateur « : » le tout englobé dans 2 accolades

Si un élément du fichier HTML correspond au sélecteur de cet élément les déclarations sont appliquées à cet élément

```
<!DOCTYPE html>
<html>
<head>
  <title>Simple CSS Example</title>
  <style>
    body {
      font-family: "Komika Hand";
      font-size: 250%;
    }
    p {
      color: blue;
      font-size: 1.4em;
    }
    em {
      color: teal;
    }
  </style>
</head>
<body>
  <p>A paragraph
    <em>with emphasis</em>
  </p>
</body>
</html>
```

`<p style="color: red;">Another paragraph</p>`

On peut directement appliquer un style dans un élément (déconseillé).

Dans cet exemple le style est inclus dans la page web mais il est préférable de l'extraire dans un fichier externe puis de le lier ainsi (dans l'entête html <head>)

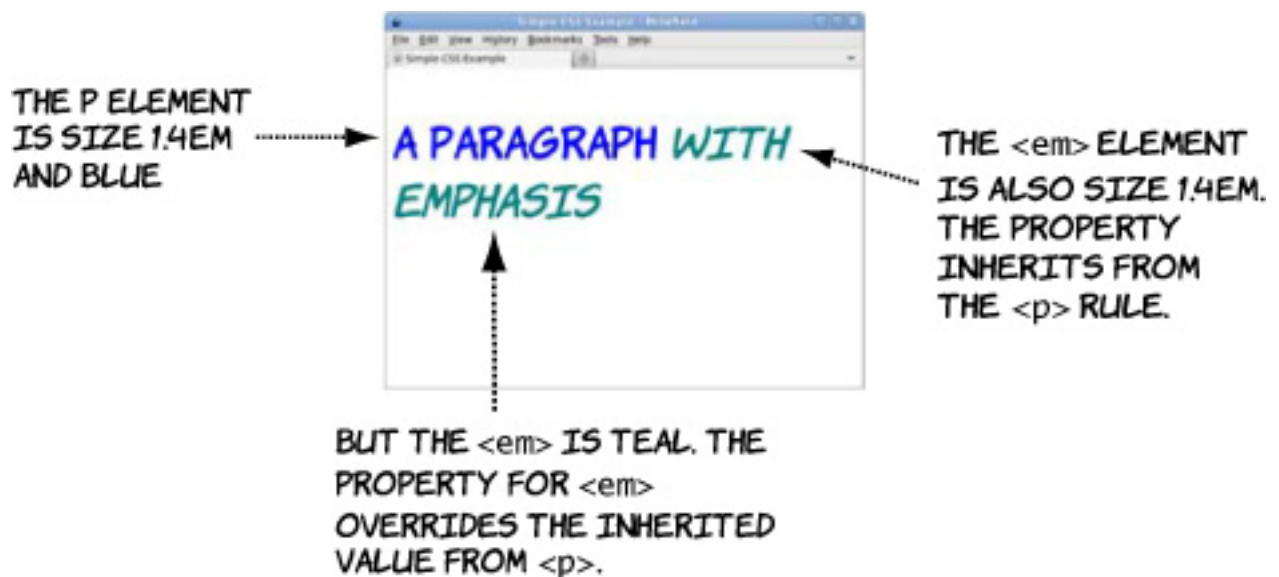
```
<link href="/bundles/common/css/master-30373b45.css" type="text/css" rel="stylesheet" >
```

Mais il peut y avoir combinaison des deux (CASCADING)

```
<head>
  <title>CSS Cascade 1</title>
  <link href="style-1.css" rel="stylesheet">
  <style>
    p {
      color: black;
      background-color: white;
    }
  </style>
</head>
```

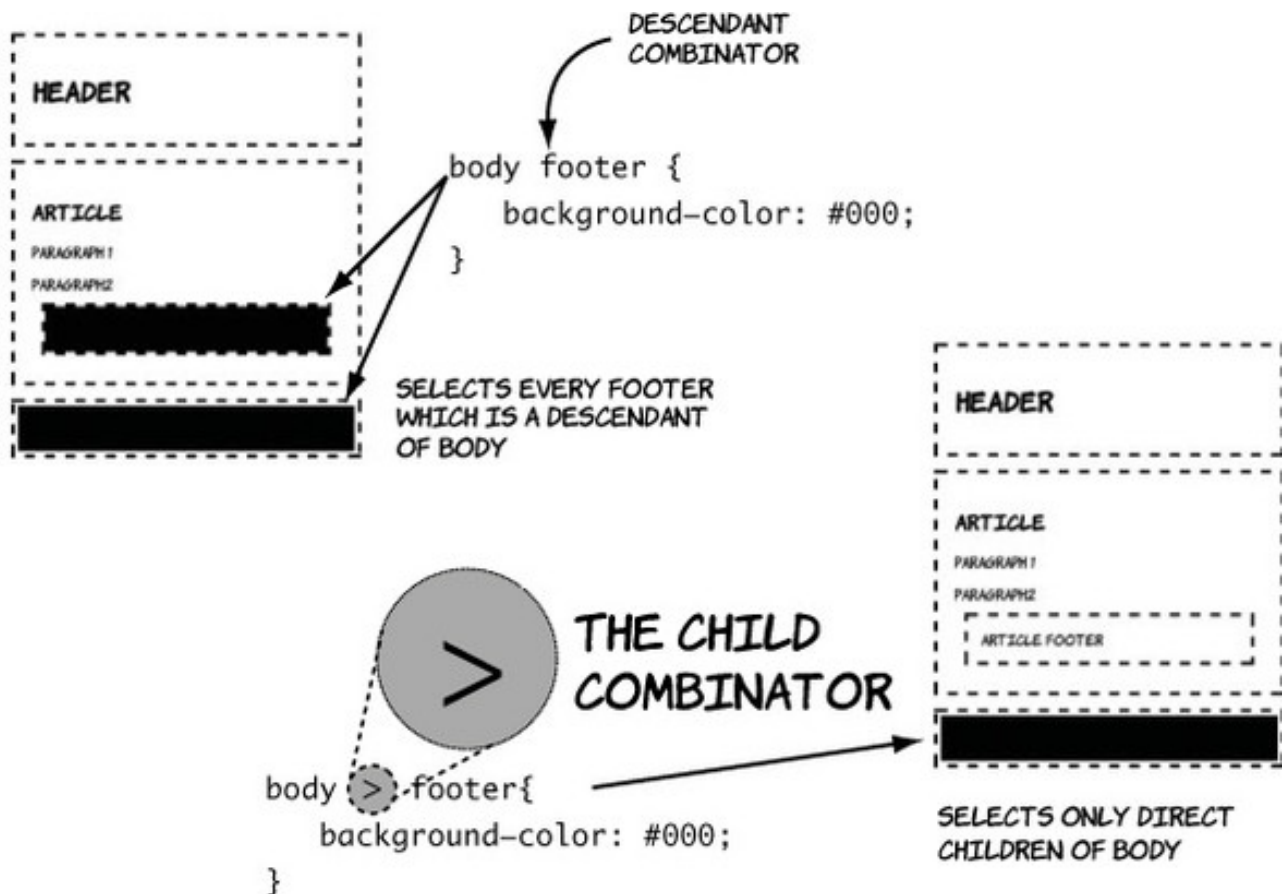
## Héritage

```
p { color: blue; font-size: 1.4em; }
em { color: teal; }
```



## Le combineur « > »

```
<header>
  <h1>Header</h1>
</header>
<article>
  <h1>Article</h1>
  <p>Paragraph 1</p>
  <p>Paragraph 2</p>
  <footer>Article footer</footer>
</article>
<footer>
  Body footer
</footer> »
```



## LES SELECTEURS

tags html  
classes  
id

```
<p>le paragraphe </p>
```

```
<p class="post">un post</p>
```

```
<p id="para-1">paragraphe1</p>
```

CSS:

```
p { color: red }
```

```
.post { color: green }
```

```
#para-1 { color: silver }
```

Il est possible de combiner les classes dans un seul élément, un élément peut avoir un seul id et une ou plusieurs classes

```
<p id="monid" class="ma-classe1 maclasse-2"> para.....</p>
```

Tester ceci

```
<p id="monid" class="ma-classe1 maclasse-2" style="color:red"> para.....</p>
```

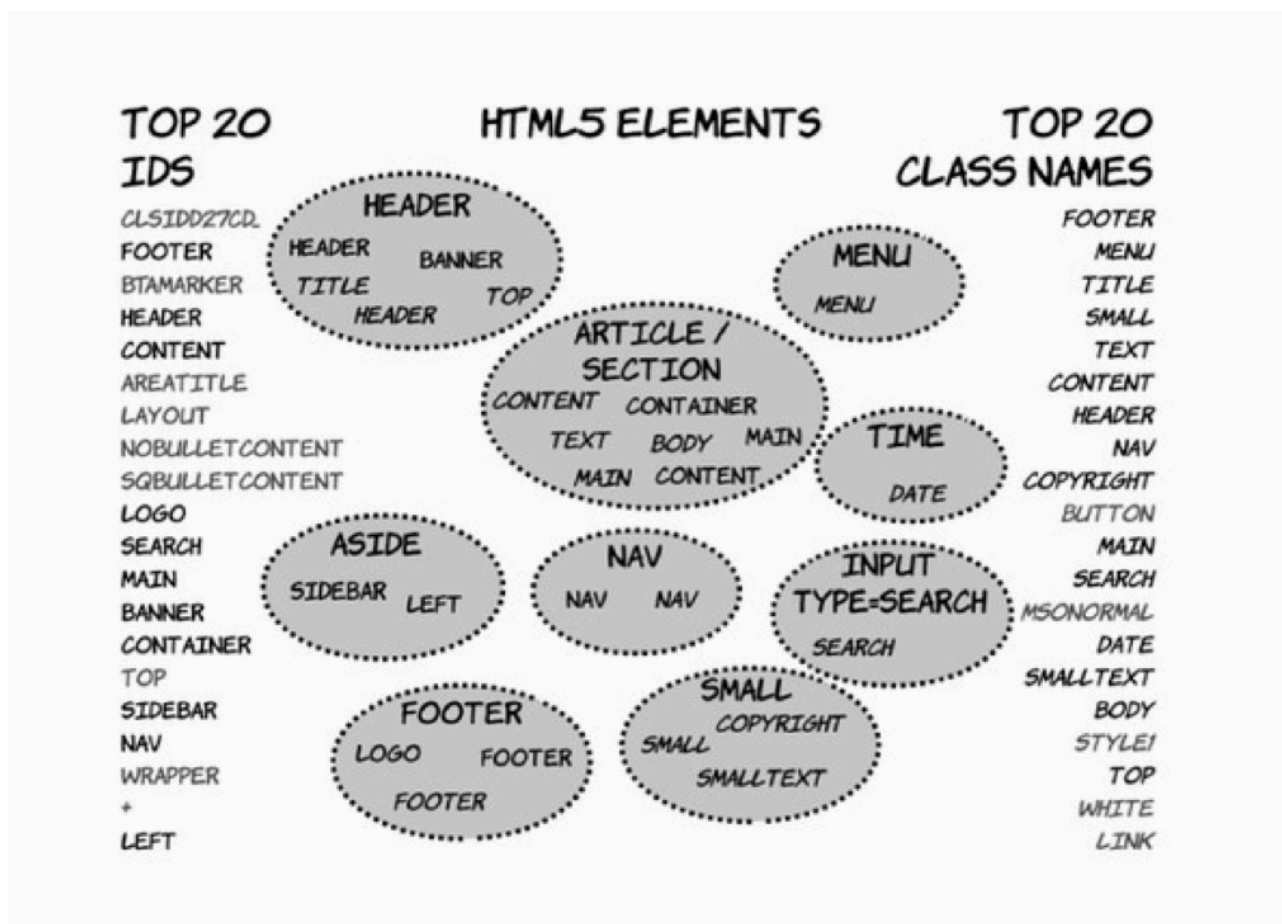
## HTML5 : Sémantique

Les éléments HTML sont censés avoir un sens, c'est ce à quoi on pense quand on parle de sémantique. Les éléments HTML4 n'ont pas vraiment de sens.

Par exemple la nouvelle balise `<address>` a plus de signification que `<div>`, ça ne change en rien l'affichage mais ça donne plus d'indications au système qui va « parser » cette page web, c'est utile par exemple aux convertisseurs text-to-speech ou à une application de mashup web.

C'est en analysant les attributs **class** et **id** de nombreux sites web que les concepteurs HTML5 ont défini de nouvelles balises voyant que certaines **class** ou **id** se répétaient très souvent `<ol id=nav>` `<div class='address'>` ...

En faisant des statistiques sur des millions de pages ils ont fait remonter ces informations (id et class) :





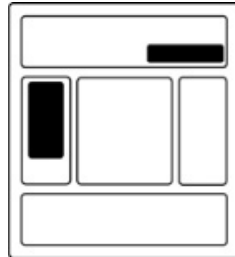
<header>

<hgroup>            regroupement de tags d'entête

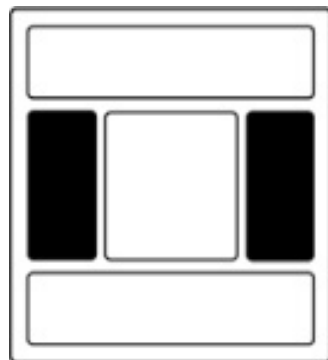
<section>

<hgroup>

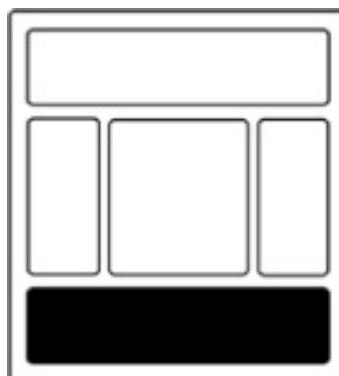
<nav>



<aside>



<footer>



<small>            souvent dans footer ou aside , on y trouve copyright, mentions légales ..etc..

<time>

exemple

```
<time datetime=" 2011-06-01">today</time>
```

```
<time datetime="2011-06-01T18:00:00+01:00">6 o'clock on 1/6/2011</time>
```

computers can read the ISO 6801 date and see the date and time is readable by human.

<place>

<figure> image avec légende.

<fieldset> séparateur de formulaires (border + légende)

exemple

```
<fieldset style="colour" title="Other form elements">
  <legend>Example</legend>
  <label>
    Upload file
    <input type="file" name="name">
  </label>
</fieldset>
<input type="submit">
```

## Retour sur la balise <input> en HTML5

### Type HTML5 rajoutés

Number, range, date, time, email, url

Chercher de la documentation et tester ces types.

### La validation

required, min, max, pattern (regexp),  
les types email et url font l'objet d'une validation

### nouveaux type (pas encore tout à fait validés sur tous les navigateurs) :

color  
meter  
progress

### Le placeholder:

indication dans le input

```
<input type="email" placeholder="email@example.com">
```



Les médias:

## images:

```

```

```

```

Indiquer la taille des images : svg OUI !!! jpg,png,gif

Les images peuvent être embarquées si elle sont en base64.

<https://fr.w3docs.com/tool/>

## Audio/ Videos:

Les sons et videos sont stockés en format compressé. Ils doivent pour cela être encodés dans ce format puis décodés. Le logiciel qui permet l'encodage et le décodage est appelé **codec**. Ces codecs sont généralement reconnus par l'extension des fichiers.

exemple: mp3 pour l'audio et mpeg2 pour la video (noté .mp4)

format ogg / ogv (firefox)

web (google video)

		WAV	OGG	MP3	AAC	WebM
Browser support quick check: audio codecs		8	5	5	5	8
		3.5	3.5	~	~	4
		~	~	9	9	[*]
		10.5	10.5	~	~	11.1
		4	[**]	4	4	[**]

Pour écouter sur IE,Firefox, chrome, safari

```
<audio id="myaudio" controls>
  <source src="myaudio.mp3" type="audio/mp3">
  <source src="myaudio.ogg" type="audio/ogg">
  No audio support!
</audio>
```

## HTML5 creation du tag video

Pour lire les vidéos auparavant on passait par flash

```
<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/
  swflash.cab#version=6,0,40,0"
width="320" height="240"
id="myvideoname">
  <param name="movie"
    value="myvideo.swf">
  <param name="quality" value="high">
  <param name="bgcolor" value="#ffffff">

  <embed href="myvideo.swf"
    quality="high" bgcolor="#ffffff"
    width="320" height="240"
    name="myvideoname"
    type="application/x-shockwave-flash"
    pluginspage="http://www.macromedia.com/
      go/getflashplayer">
  </embed>
</object>
```

Il fallait utiliser un **plug-in**

**plug-in**: extension qui permet navigateur de rendre un contenu par un programme externe.

Mais suite à une déclaration de Steve Jobs annonçant qu'il ne supporterait pas flash sur iPhone, flash fut petit à petit abandonné<sup>3</sup>.

---

<sup>3</sup><https://flowplayer.com/blog/the-rise-and-fall-of-flash>

Une video c'est un conteneur et un codec

Le **conteneur** est un format de fichier qui contient les flux audio et vidéo, l'information codec et les métadonnées.

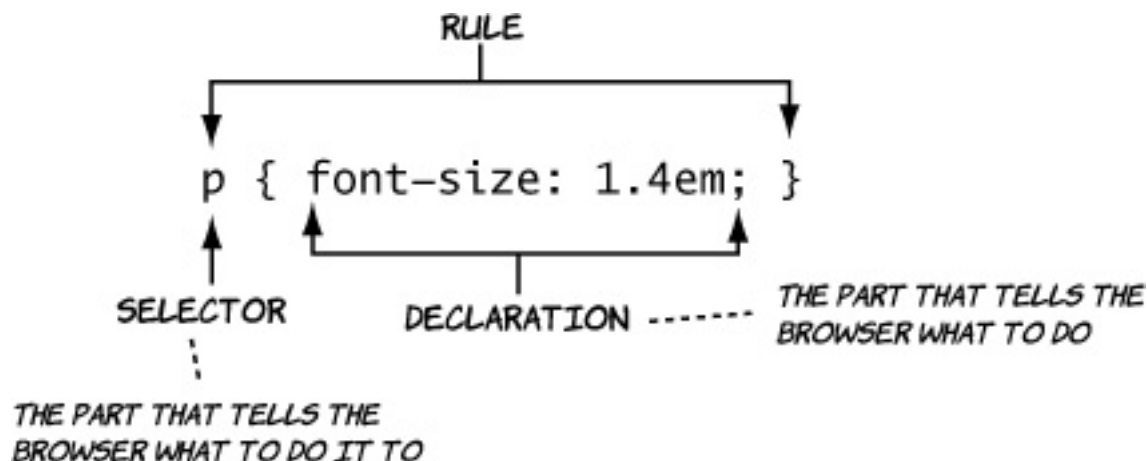
Un **codec** (acronyme de codage-décodage) est un algorithme de compression / décompression d'un signal audiovisuel numérique.

Le codec est un programme qui se trouve sur la machine (il y en a des centaines). Le codec le plus courant est le mpeg4 ou H.264 créé par Apple .

Actuellement la plupart du temps on trouve ces fichiers video avec le suffixe .mp4.

Aujourd'hui HTML5 permet de s'affranchir des plug-in et il est recommandé d'encoder en MP4/h.264, AAC (Advance Audio Encoding)

## CSS



Il y a 3 façons d'insérer une feuille de style

feuille de style externe nommée unnom.css et importée dans la section head de la page HTML ainsi

```
<link href='/chemin/de/monfichier/unnom.css' rel='stylesheet' type='text/css' />
```

Déclarer un tag <style> directement dans la page html (déconseillé)

```
<head>
<style>
body { color:red ;}
</style>
</head>
```

Applique l'attribut style directement dans un élément (encore plus déconseillé)

```
<div style="color:red;font-size:16px">lorem ipsum.....</div>
```

Nous verrons plus tard que des actions JQuery insèrent directement ces directives css dans les éléments.

### Typographie:

- font-family
- font-size
- font-style
- font-weight
- color

font-family: 'police voulue','web safe fonts','generic'

police voulue : cherche la police dans votre PC , ou sera inséré par CSS

web safe font : police théoriquement universelle pour tous les navigateurs (genre Verdana, Helvetica...etc) on peut en mettre plusieurs à la suite.

generic : dernier recours, cinq options : serif, sans serif, monospace , cursive, fantasy (en general : serif)

Polices google:

CSS permet d'insérer des polices google ([fonts.google.com](https://fonts.google.com))

```
<!DOCTYPE html>
<html>
  <head>
    <title>Cours HTML et CSS</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="cours.css">
    <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Open+Sans|Roboto">
  </head>

  <body>
    <h1>Un titre de niveau 1</h1>

    <p>Un premier paragraphe</p>
    <p id="p2">Un autre paragraphe</p>
  </body>
</html>
```

```
body{
  font-family: "Open sans", Georgia, serif;
}

#p2{
  font-family: Roboto, Verdana, sans-serif;
}
```

[fonts.google.com](https://fonts.google.com)

<https://www.dafont.com/fr/>



## Utiliser une police externe avec CSS

Comme pour les vidéos il faut plusieurs types de fichiers pour exploiter les polices sur les différents navigateurs

Pour ça il faut uploader une police choisie sur DAFONT par exemple puis générer le ou les fichiers nécessaires avec un site très utile : <https://www.fontsquirrel.com/tools/webfont-generator>

- Télécharger une police sur Dafont
- la passer dans Fontsquirrel
- télécharger le fichier et décompresser
- On aura des fichiers de type .woff reconnus aujourd'hui
- puis insérer l'exemple CSS fourni

/\*! Generated by Font Squirrel (<https://www.fontsquirrel.com>) on August 22, 2020 \*/

```
@font-face {  
  font-family: 'quadrantaregular';  
  src: url('quadranta-regular-webfont.woff2') format('woff2'),  
       url('quadranta-regular-webfont.woff') format('woff');  
  font-weight: normal;  
  font-style: normal;  
}
```

Puis utiliser cette font-family, exemple:

```
h1 { font-family: 'quadrantaregular' ;}
```

TP: Tester avec l'exemple fourni par Fontsquirrel

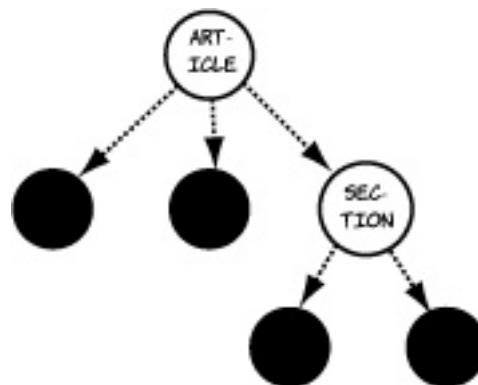
CSS3 ajoute l'utilisation de combineurs et de pseudo-classes

## Combinateur

Considérons

```
<article>  
  <p></p>  
  <p></p>  
  <section>  
    <p></p>  
    <p></p>  
  </section>  
</article>
```

Donne le DOM suivant



si on applique le style

```
article p { color : red; }
```

l'espace après article indique que le style s'appliquera aux sélecteur « P » descendants de « article ».

mais si on utilise le combineur greater-than ainsi

```
article > p {color : red }
```

le style sera appliqué aux enfants directs de « article »

### **Combinateur de voisin direct**

```
h1 + p { background-color: #000; }
```

Elément directement situé après et seulement celui-là

### **Général sibling**

```
h1 ~ p { background-color: #000; }
```

### **Pseudo-classes**

```
ul li:first-child  
ul li:last-child
```

```
li:last-child {  
  background-color: #000;  
}  
li:last-child a {  
  background-color: #fff;  
}
```

Le n-ième élément

```
nth-child(1), :nth-child(2), :nth-child(3)
```

Tous les second éléments (ou pairs) . Utilisation du pattern « 2n »

```
li:nth-child(2n) {  
  background-color: #000;
```

Pour les éléments impairs

```
li:nth-child(2n-1) {  
    background-color: red ;  
}
```

raccourcis

```
i:nth-child(odd) {  
    background-color: #f00;  
}  
li:nth-child(even) {  
    background-color: #006;  
}
```

## Selection des éléments par leur attributs

```
<span typeinfo="sport">page des sports</span>  
<span typeinfo="economie">page economie</span>
```

**style :**

```
[type info]          { font-size: 24px; }  
[type info=sport]    { colour:red ; }
```

## expression régulière

selection des élément possédant l'attribut itemprop dont la propriété commence par org

```
[itemprop^="org"] {  
    outline: 4px dashed black;  
    display: block;  
}
```

