

Exercice d'application : CV en HTML

Réaliser votre CV avec une photo et un lien vers un site

En entête de la page CV (balise **head**)

En titre vos prénom et nom de famille

En dessous une adresse de domicile

Un Code Postal

Un numéro de téléphone

Une adresse email

Un titre (**H3**) : Formation

En dessous : une liste non ordonnée mentionnant vos formations

Un titre (**H3**) : expérience professionnelle

En dessous : une liste non ordonnée mentionnant vos formations

2.7 Les formulaires

Un formulaire HTML permet d'envoyer des données au serveur sous la forme de variables (paire nom / valeur) appelés champs. Un formulaire est un ensemble de champs, il peut donc y avoir plusieurs formulaires dans une même page Web. Chaque formulaire est indépendant et possède l'url du fichier de destination des données dans un attribut.

Il existe plusieurs méthodes d'envoi des données (GET et POST).

- Les données envoyées par la méthode GET sont visibles dans l'url dans la barre d'adresse du navigateur.
- Les données envoyées par la méthode POST, sont encapsulées dans l'entête HTTP de la requête (mais restent accessibles via l'inspecteur dans la partie réseau).

L'envoi des données via un formulaire peut être sécurisé en utilisant le protocole HTTPS, à configurer sur le serveur.

Exemple de formulaire :

Le formulaire est divisé en deux sections principales, chacune encadrée par une bordure rouge. La première section, intitulée "A propos des CSS :", contient une question "Savez-vous ce que veut dire CSS ? :" avec deux boutons radio "oui" (sélectionné) et "non". En dessous, une question "Si oui, les utilisez-vous plutôt :" est suivie d'un menu déroulant affichant "toujours". La deuxième section, intitulée "Vos coordonnées :", contient un champ de texte "Votre email :" avec la valeur "email" et un grand champ de texte "Vos commentaires :". En bas du formulaire, il y a deux boutons "Envoyer" et "Annuler".

2.7.1 La balise form

Elle doit contenir l'ensemble des champs du formulaire et à minima, l'url du fichier de destination des données dans l'attribut action ainsi que la méthode utilisée.

```
<form action="contact.php" method="post">  
  ...  
</form>
```

Dans le cas où l'on souhaite associer des fichiers, on peut préciser l'encodage de ces derniers.

```
<form action="contact.php" method="post" enctype="multipart/form-data">  
  ...  
</form>
```

2.7.2 Les libellés

Pour afficher correctement un label d'un champ, on utilise la balise `label`. L'association se fait via l'attribut `for` du label qui doit correspondre à la valeur de l'*ID* du champ.

```
<label for="city-id">Votre ville</label>
<input type="text" name="city" id="city-id">
```

Dans l'exemple, on a un *ID* qui est différent du *name* mais ils peuvent être identique.

L'association permet par exemple de cocher une case à cocher directement en cliquant sur le libellé, ce qui améliore l'accessibilité et l'ergonomie du formulaire.

Dans le cas des cases à cocher ou des boutons radio, il est possible de placer directement la balise `input` dans la balise et d'omettre le couple `for` / *ID*:

```
<label>
<input type="checkbox" name="conditions">
J'ai pris connaissance des conditions d'utilisation.
</label>
```








Les regroupements de champs

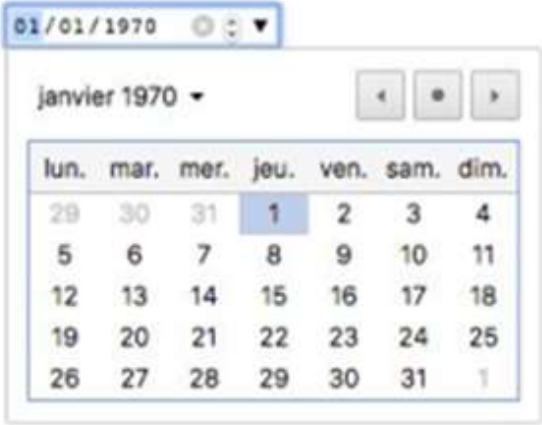
Pour organiser les champs par groupes, on peut utiliser le couple de balises `fieldset` / `legend`. La balise `legend` n'est pas obligatoire mais si elle est présente, elle doit être le premier élément du `fieldset`.

```
<fieldset>
  <legend>Titre du groupe de champs</legend>
  ...
</fieldset>
```

2.7.3 La balise input

C'est une balise auto-fermante. Elle doit contenir, à minima, le type ainsi que le l'attribut name qui correspond au nom de la variable côté serveur.

text	champ texte simple (par défaut) 
password	champ dont la valeur est un mot de passe, elle est cachée, remplacée par des points pour éviter d'être lue par d'autres au moment de la saisie 
radio	choix unique parmi plusieurs (OU) <input checked="" type="radio"/> Option 1 <input type="radio"/> Option 2 <input type="radio"/> Option 3
checkbox	choix multiple parmi plusieurs (ET) <input checked="" type="checkbox"/> Choice A <input type="checkbox"/> Choice B <input type="checkbox"/> Choice C
file	champ permettant de sélectionner un ou plusieurs fichiers sur son ordinateur 
hidden	champ caché, très utilisé pour transmettre au serveur des informations liées à la sécurité ou à la session en cours
search	champ dédié à la recherche 
number	champ permettant uniquement la saisie d'un nombre 
range	champ de choisir une valeur dans une plage de valeurs (avec une réglette coulissante) 
color	champ permettant uniquement la saisie d'une couleur 

date	champ permettant de choisir une date 
email	champ permettant uniquement la saisie d'une adresse email <input type="email" value="name@email.com"/>
url	champ permettant uniquement la saisie d'une URL <input type="url" value="http://127.0.0.1"/>
tel	champ dédié à la saisie d'un numéro de téléphone <input type="tel" value="(999) 999-9999"/>

Pour afficher correctement un label d'un champ, on utilise la balise label. L'association se fait via l'attribut `for` du label qui doit correspondre à la valeur de l'`ID` du champ.

```
<label for="city-id">Votre ville</label>
<input type="text" name="city" id="city-id">
```

Dans l'exemple, on a un `ID` qui est différent du `name` mais ils peuvent être identique.

L'association permet par exemple de cocher une case à cocher directement en cliquant sur le libellé, ce qui améliore l'accessibilité et l'ergonomie du formulaire.

Dans le cas des cases à cocher ou des boutons radio, il est possible de placer directement la balise `input` dans la balise et d'omettre le couple `for / ID`:

```
<label>
<input type="checkbox" name="conditions">
J'ai pris connaissance des conditions d'utilisation.
</label>
```

input type="text"

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

Nom de famille

input type="email"

```
<p>  
  <label for="email">Email</label>  
  <input type="email" id="email" name="email" />  
</p>
```

Email

input type="password"

```
<p>  
  <label for="mdp">Mot de passe</label>  
  <input type="password" id="mdp" name="mot_de_passe" />  
</p>
```

Mot de passe

Input type="date"

```
<p>  
  <label for="naissance">Date de naissance</label>  
  <input type="date" id="naissance" name="date_de_naissance" />  
</p>
```

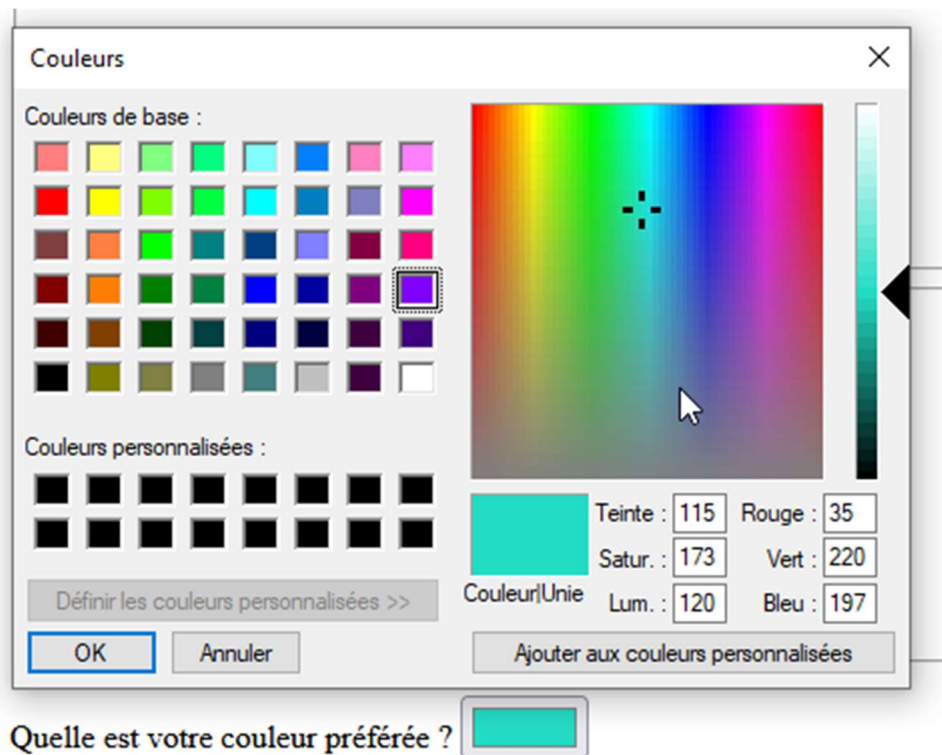
Date de naissance

< octobre 2021 >

lun.	mar.	mer.	jeu.	ven.	sam.	dim.
27	28	29	30	1	2	3

input type="color"

```
<p>
  <label for="couleur">Quelle est votre couleur préférée ?</label>
  <input type="color" id="couleur" name="couleur_preferee" />
</p>
```



input type="radio"

```
<p>
  Sexe :
</p>
<p>
  <input type="radio" id="homme" name="sexe" value="homme" />
  <label for="homme">Homme</label>
  <input type="radio" id="femme" name="sexe" value="femme" />
  <label for="femme">Femme</label>
  <input type="radio" id="autre" name="sexe" value="autre" />
  <label for="homme">Autre</label>
</p>
```

Sexe :
☐ Homme ☐ Femme ☐ Autre

input type="checkbox"

Les cases à cocher offrent la possibilité d'effectuer un choix multiple, aussi il est préférable de les regrouper dans un tableau. En ajoutant des crochets dans l'attribut name, cela enregistre les choix dans un tableau :

name="nom_variable[]"

Ensuite, il est nécessaire d'indiquer la valeur que l'on souhaite enregistrer :

value="valeur_choisie"

```
<div>Quels sont vos 2 mangas préférés ?:</div>
<p>
  <input type="checkbox" id="naruto" name="choix_mangas[]" value="naruto">
  <label for="naruto">Naruto</label>
</p>
<p>
  <input type="checkbox" id="dragon_ball" name="choix_mangas[]" value="dragon_ball">
  <label for="dragon_ball">Dragon Ball</label>
</p>
<p>
  <input type="checkbox" id="kaiju_n_8" name="choix_mangas[]" value="kaiju_n_8">
  <label for="kaiju_n_8">kaiju_n_8</label>
</p>
<p>
  <input type="checkbox" id="one_piece" name="choix_mangas[]" value="one_piece">
  <label for="one_piece">One Piece</label>
</p>
```

Quels sont vos 2 mangas préférés ?:

- ☐ Naruto
- ☐ Dragon Ball
- ☐ kaiju_n_8
- ☐ One Piece

Le serveur recevra les choix de l'utilisateur sous forme de tableau :

```
[choix_mangas] => Array
(
    [0] => naruto
    [1] => dragon_ball
)
```

2.7.4 La balise select

Si l'on souhaite afficher un choix unique (comme les boutons radio) ou multiple (comme les cases à cocher) sous la forme d'un menu, on peut utiliser la balise *select*.



Un exemple simple

Ici les carottes sont présélectionnées.

```
<select name="panier">
  <option value="fraises">Fraises</option>
  <option value="bananes">Bananes</option>
  <option value="carottes" selected>Carottes</option>
  <option value="navets">Navets</option>
</select>
```

Présélection et affichage

Ici les bananes et les carottes sont présélectionnées. L'attribut `size` permet d'indiquer le nombre de lignes affichées simultanément.

```
<select name="panier" multiple size="3">
  <option value="fraises">Fraises</option>
  <option value="bananes" selected>Bananes</option>
  <option value="carottes" selected>Carottes</option>
  <option value="navets">Navets</option>
</select>
```

Regroupement par catégorie

Pour organiser les options par groupes, on peut les regrouper dans des balises *optgroup*.

```
<select name="panier" multiple size="3">
  <optgroup label="Fruits">
    <option value="fraises">Fraises</option>
    <option value="bananes" selected>Bananes</option>
  </optgroup>
  <optgroup label="Légumes">
    <option value="carottes" selected>Carottes</option>
    <option value="navets">Navets</option>
  </optgroup>
</select>
```

2.7.5 la balise datalist

Pour créer la balise datalist 4 étapes :

```
<!-- 1ERE ETAPE : ON CREER UNE BALISE IMPUT CLASSIQUE SANS SE SOUCIER  
DE LA BALISE DATALISTE -->
```

```
<label for="choix_serie">Choisissez votre série</label>  
<input type="text" id="choix_serie" name="choix_serie" />
```

```
<!-- 2EME ETAPE : ON AJOUTE A LA BALISE DATALISTE L'ATTRIBUT list="nom_liste"> -->
```

```
<label for="choix_serie">Choisissez votre série</label>  
<input type="text" id="choix_serie" name="choix_serie" list="liste_series" />
```

```
<!-- 3EME ETAPE : ON CREER LA BALISE DATALISTE EN LUI DONNANT UN id="nom_liste"  
CET ID DOIT CORRESPONDRE A LA VALEUR DE L ATTRIBUT list="nom_liste" -->
```

```
<datalist id="liste_series">
```

```
</datalist>
```

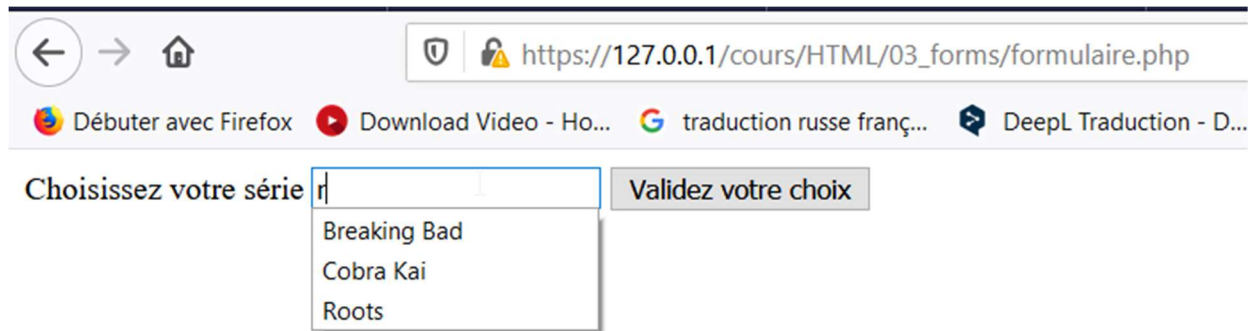
```
<!-- 4EME ETAPE : ON CREER LA LISTE DE DONNEES COMME UNE BALISE SELECT AVEC OPTION  
MAIS SANS LE SELECT (LA BALISE DATALISTE REMPLACE LE SELECT) -->
```

```
<datalist id="liste_series">  
  <!--<select name="" id="">-->  
    <option value="breaking_bad">Breaking Bad</option>  
    <option value="cobra_kai">Cobra Kai</option>  
    <option value="roots">roots</option>  
    <option value="the_it_crowd">The IT Crowd</option>  
  <!--</select> -->  
</datalist>
```

Synthèse :

```
<label for="choix_serie">Choisissez votre série</label>  
<input list="liste_serie" type="text" id="choix_serie" name="choix_serie">  
<datalist id="liste_serie">  
  <option value="Breaking_Bad">Breaking Bad</option>  
  <option value="Cobra_Kai">Cobra Kai</option>  
  <option value="roots">Roots</option>  
</datalist>
```

Rendu dans le navigateur :



The screenshot shows a web browser window with the address bar displaying `https://127.0.0.1/cours/HTML/03_forms/formulaire.php`. The browser's search bar contains the text "Choisissez votre série". A dropdown menu is open, showing three options: "Breaking Bad", "Cobra Kai", and "Roots". To the right of the input field is a button labeled "Validez votre choix".

Si le navigateur ne prend pas en charge la balise `datalist`, il est possible d'ajouter une balise `select`, celle-ci sera prise en compte le cas échéant :

```
<!-- DATALIST NAVIGATEURS INCOMPATIBLES -->
<label for="choix_serie">Choisissez votre série</label>
<input list="serie" type="text" id="choix_serie" name="choix_serie">
<datalist id="serie">
  <label for="ma_serie">Liste</label>
  <select id="ma_serie" name="ma_serie">
    <option value="Breaking_Bad">Breaking Bad</option>
    <option value="Cobra_Kai">Cobra Kai</option>
    <option value="The_Expanse">The Expanse</option>
    <option value="The_It_Crowd">The IT Crowd</option>
    <option value="The_Walking_Dead">The Walking Dead</option>
  </select>
</datalist>

<button type="submit">Validez votre choix</button>
</form>
```


2.7.6 La balise textarea

Si l'on souhaite un champ de saisie multilignes, il faut utiliser la balise textarea.

Par exemple:

```
<textarea name="name" placeholder="Veuillez saisir votre  
adresse"></textarea>
```



Attention le contenu initial doit se placer entre la balise ouvrante et la balise fermante. Si le contenu est vide, les balises ouvrante et fermante doivent être collées (sinon, il se peut que vous envoyez des espaces comme valeur au serveur).

2.7.7 Les boutons

Pour envoyer les données au serveur, il faut valider le formulaire via un bouton.

On peut utiliser la balise `input` avec le type `"submit"` mais il est préférable d'utiliser la balise `button`. Elle est composée d'une balise ouvrante et d'une balise fermante. Le libellé se place entre les deux et peut contenir d'autres éléments comme une image.

Elle possède également un attribut `type` qui peut prendre 3 valeurs:

submit	les données sont envoyées au serveur à l'adresse spécifiée dans l'attribut <code>action</code> de la balise <code>form</code>
button	aucune donnée n'est envoyée, très souvent utilisé conjointement avec du JavaScript pour effectuer des actions personnalisées sur le formulaire
reset	très peu utilisé, réinitialise tous les champs du formulaire en leur donnant leur valeur par défaut

Un exemple de bouton d'envoi des données avec l'attribut `type="submit"`:

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

Barre de progression :

Représentée sous forme de jauge horizontale (par défaut), elle requiert 2 Valeurs, l'une pour déterminer le maximum (`max`) et l'autre le niveau (`value`).

```
<label for="barre_chargement">Téléchargement en cours</label>  
<progress id="barre_chargement" max="100" value="75"></progress>
```

Résultat :

Téléchargement en cours 

Couplée avec un script JavaScript elle peut se modifier dynamiquement.

2.7.8 Regroupement de champs : Fieldset

Pour organiser les champs par groupes, on peut utiliser le couple de balises *fieldset* / *legend*. La balise *legend* n'est pas obligatoire mais si elle est présente, elle doit être le premier élément du *fieldset*.

```
<fieldset>
  <legend>Titre du groupe de champs</legend>
  ...
</fieldset>
```

2.8 Audio et vidéo

Le HTML5 a ajouté le support natif de la lecture des fichiers audio et vidéo directement par le navigateur. Avant cela, il était obligatoire d'ajouter des plugins au navigateur tels que Flash ou Silverlight. Ces plugins ont posé beaucoup de problème de compatibilité et de sécurité.

La spécification HTML5 permet de lire des fichiers audio et vidéo directement dans le navigateur en utilisant deux balises dédiées : **<audio>** et **<video>**. Elle permet aussi de manipuler les médias à l'aide d'une API JavaScript associée.

Mise en place

Dans la plupart des cas, il est préférable de déléguer l'affichage de la vidéo à un service spécialisé comme les plateformes de vidéos telles que YouTube ou Vimeo. Ces plateformes ont des infrastructures adaptées aux vidéos, à savoir des serveurs performants avec de grosses capacités de stockage et des gros débits de données.

Dans ce cas, le code source à insérer est disponible le plus généralement depuis le menu "partager" et ressemblera à cet exemple (vimeo).

```
<iframe src="//player.vimeo.com/video/11386048?portrait=0&color=8BA0FF" width="500" height="281" frameborder="0" webkitallowfullscreen mozallowfullscreen allowfullscreen></iframe>
```

Si vous préférez héberger par vous-même vos fichiers vidéo, il faudra opter pour un hébergement adapté avec à minima un espace disque conséquent. Dans ce cas, il existe aussi des players complets permettant de configurer l'affichage des éléments de contrôle de la vidéo ainsi que de gérer automatiquement l'affiche de la vidéo pour tous les navigateurs à l'aide de technologies de remplacement si besoin.

Quelques players :

- <http://blog.jilion.com/2010/01/25/introducing-sublimevideo>
- <http://mediaelementjs.com/>
- <http://jplayer.org/>

2.8.1 La balise video

Dans certains cas, vous aurez besoins d'utiliser directement la balise video. La balise **video** comporte, à l'inverse de la balise **image**, une balise ouvrante et une balise fermante. Le texte contenu entre les deux balises ne sera lu que par les navigateurs non compatibles. On y place généralement un texte incitant à changer de navigateur ou une source à télécharger sous la forme d'un lien.

Le minimum :

Pour afficher une vidéo avec le minimum de paramètres

```
<video src="media/intro.webm" controls>
```

Votre navigateur ne prend pas en charge les vidéos HTML5, merci d'utiliser un navigateur plus récent.

```
</video>
```

Dans ce cas la source de la vidéo est définie via l'attribut **src**

Source alternatives

Il est possible d'ajouter des sources alternatives pour améliorer la compatibilité avec les différentes plateformes.

```
<video controls><source src="media/intro.webm" type="video/webm"><source  
src="media/intro.mp4" type="video/mp4"> Votre navigateur ne prend pas  
en charge les vidéos HTML5, merci d'utiliser un navigateur plus  
récent.
```

```
</video>
```

Dans ce cas, on ajoute autant de sources que l'on possède de versions alternatives de la même vidéo, en précisant le type MIME correspondant (l'attribut type). Ces informations sont en générales fournies par le logiciel d'encodage de la vidéo au moment de l'export.

Les paramètres

Plusieurs paramètres sont disponibles sous la forme d'attributs.

Voici une déclaration plus complète:

```
<video width="480" height="320" poster="poster.jpg" preload="auto"autoplay
controls loop muted><source src="media/intro.webm" type="video/webm"><source
src="media/intro.mp4" type="video/mp4">Votre navigateur ne prend pas en charge les
vidéos HTML5, merci d'utiliser un navigateur plus récent. </video>
```

Voici les paramètres disponibles :

Attribut	Valeurs possibles	Description / exemple
width	Une valeur (en pixels) ou en pourcentage (mettre %)	La largeur de la vidéo
height	Une valeur (en pixels) ou en pourcentage (mettre %)	La hauteur de la vidéo
poster	Un chemin vers un fichier image	Permet de spécifier l'image visible avant le chargement de la video <code>poster="poster.jpg"</code>
autoplay	aucune (attribut booléen)	Si présent, permet de lancer la vidéo en lecture automatiquement dès que possible.
controls	aucune (attribut booléen)	Si présent, permet d'afficher les boutons de contrôle de lecture et du volume.
loop	aucune (attribut booléen)	Si présent, joue la vidéo en boucle.
muted	aucune (attribut booléen)	Si présent, le son de la vidéo sera coupé lors du lancement.
preload	<i>none</i> : la vidéo n'est pas préchargée <i>metadata</i> : seules les métadonnées de la vidéo (durée, taille) sont chargées <i>auto</i> : la vidéo est préchargée	Contrôle le préchargement de la vidéo



Attention Les attributs booléens doivent être présent pour signifier "vrai" et absent pour signifier "faux".

2.8.2 Balise Audio

Il existe également une balise audio fonctionnant de la même manière.

Exemple complet :

```
<audio preload="auto" autoplay controls loop muted><source  
src="media/musique.ogg" type="audio/ogg"> <source  
src="media/musique.mp3" type="audio/mp3">Votre navigateur ne  
prend pas en charge l'audio HTML5, merci d'utiliser un navigateur plus  
récent.</video>
```

Les attributs de taille width et height ainsi que l'attribut poster ne sont pas valide pour l'élément audio.

2.8.3 Les éléments embarqués (frameset, iframe, etc...).

Les premières années d'Internet étaient déjà marquées par le désir d'intégrer des éléments d'autres sites au sein d'un site. De cette manière, l'écran du navigateur était divisé en plusieurs parties (zones). Chaque zone permettait d'intégrer une page HTML. L'ensemble de ces parties formait un jeu de cadres (frameset).

Les propriétés rows et cols permettent de segmenter la page en proportion (%) ou en pixels.

```
<frameset rows="33%,33%,33%" cols="50%,50%">  
  <frame>#intégration 1</frame>  
  <frame>#intégration 2</frame>  
  <frame>#intégration 3</frame>  
  <frame>#intégration 4</frame>  
  <frame>#intégration 5</frame>  
  <frame>#intégration 6</frame>  
</frameset>
```

Une page pouvait par exemple être segmenté horizontalement (rows) en 3 parties et verticalement (cols) en 3.

Rendu visuel :

Cependant, l'usage des frameset posait plusieurs problèmes notamment : failles de sécurité, référencement, lecture ou sélection des contenus (ergonomie). Cette balise est désormais obsolète.

D'autres techniques sont apparues pour intégrer notamment des vidéos, des animations, des plugins, dont l'intégration était impossible en HTML : les applet Java ou Flash. Ces intégrations ont été rendues possibles grâce aux éléments <object> et <embed>.

La balise object permettait notamment d'intégrer les plugins flash (abandonné en janvier 2021).

La balise embed permet également l'intégration d'applications externes, de vidéos, de plugins :

```
<embed type="video/webm" src="/assets/videos/flower.mp4" width="250" height="200">
```

Ces 2 balises sont désormais moins utilisées, il faut désormais privilégier la balise <iframe>

La balise Iframe autorise l'intégration de nombreux éléments : documents pdf, vidéos, carte google maps.... Il s'agit d'un élément imbriqué qui permet également d'intégrer une page HTML.

Il s'agit d'une balise paire qui a comme principales propriété src, width, height.

Pdf interne :

```
<a href="assets/pdf/manga.pdf">Le manga</a>
<iframe src="assets/pdf/manga.pdf" width="100%" height="800px" frameborder="0"></iframe>
```

Vidéo externe

```
<iframe width="560" height="315" src="https://www.youtube.com/embed/cv3jasBSNr4" title="YouTube video player"
frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture"
allowfullscreen>
</iframe>
```


3. Les éléments blocks et inline

3.1 Généralités

De manière schématique, on peut considérer qu'il existe deux grands types d'affichage principaux pour les éléments HTML : **un élément HTML va pouvoir être soit de niveau (ou de type) block, soit de niveau (ou de type) inline.**

Ces types d'affichage vont définir la façon dont les éléments vont se comporter dans une page par rapport aux autres et la place qu'ils vont prendre dans la page.



3.2 Notion de flux et principe des boîtes

3.2.1 Les éléments de type « block »

**LES BLOCS S'ÉTENDENT NATURELLEMENT SUR
TOUTE LA LARGEUR DE LEUR CONTENANT**



**ILS SE METTENT AUTOMATIQUEMENT LES UNS SOUS LES
AUTRES**



Un élément block a les caractéristiques suivantes, non limitatives :

- Si aucune largeur n'est définie, il prendra toute la largeur de son élément parent.
- Il peut avoir des marges et des paddings.
- Si aucune hauteur n'est définie, il prendra la hauteur de ses éléments enfants (en supposant qu'il n'y a pas de "float" ou de positionnement sur des éléments environnants).

Il est donc inutile pour un élément block de définir une largeur ou de lui donner une `width: 100%` si vous voulez qu'il s'étende sur la largeur de son élément parent. Cela pourrait même avoir des effets indésirables.

Exemples : `<p>` `<h1>` `<h6>` `` `<form>`

La balise universelle neutre de type bloc est : `<div>`

3.2.2 Les éléments de type « inline »

INLINE ELEMENTS FLOW WITH TEXT

PELLENTESQUE HABITANT MORBI TRISTIQUE SENECTUS
ET NETUS ET MALESUADA FAMES AC TURPIS EGESTAS.
VESTIBULUM **INLINE ELEMENT** VITAE, ULTRICIES
EGET, TEMPOR SIT AMET, ANTE. DONEC EULIBERO SIT
AMET QUAM EGESTAS SEMPER. AENEAN ULTRICIES MI
VITAE EST. MAURIS PLACERAT ELEIFEND LEO.

Un élément "inline" ne commence pas sur une nouvelle ligne et ne prend que la largeur nécessaire.

- Il s'inscrit dans le flux du texte et donc ne 'saute' pas à la ligne comme le ferait un élément block
- Il ignore les marges top et bottom mais applique les marges left et right, ainsi que tout padding.
- Il ignore les propriétés width et height.

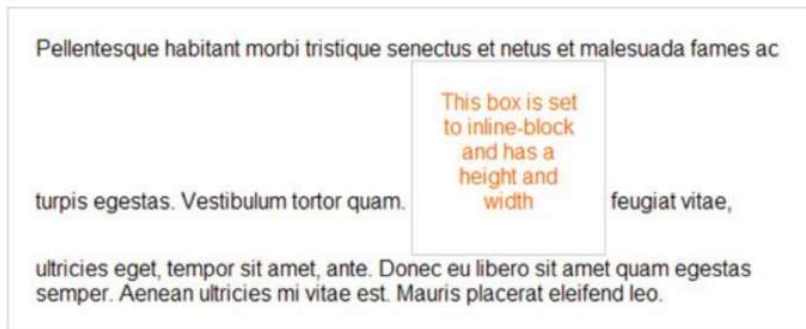
Pour se faire une bonne idée de l'élément inline, on peut l'imaginer comme une boîte qui se comporte comme du texte. Un texte qui n'est pas interrompu par d'autres éléments s'écoule, lettre après lettre, et si vous ajoutez un élément inline dans le texte, il suivra le flux comme tout autre élément du texte.

Exemples : `<a>` `` `<input>` `<select>`

La balise universelle neutre de type bloc est : ``

3.2.3 Les éléments de type « inline-block »

En gros, c'est une manière de mettre les éléments inline tout en préservant leurs capacités d'éléments block, tels que la possibilité de définir une largeur et une hauteur, des marges et paddings top et bottom, etc.



Il n'existe pas d'élément naturellement de type **inline-block**, si nous souhaitons qu'un élément soit de ce type il faudra le définir en CSS.