

Image



- Format vectoriel

- **SVG** : *Scalable Vector Graphics*

- Image décrite en XML (éditable) constituée d'éléments simples : formes, coordonnées, dimensions, couleurs, effets
 - 16 millions de couleurs disponibles
 - Réduction et agrandissement sans perte de qualité
 - Supporte l'animation et la transparence ; Taille de fichier réduite

```
<svg height="80" width="80">  
  <ellipse cx="80" cy="80" rx="80" ry="70" style="fill:blue" />  
  <ellipse cx="80" cy="80" rx="60" ry="50" style="fill:white" />  
  <ellipse cx="80" cy="80" rx="40" ry="25" style="fill:red" />
```

Desole, votre navigateur ne supporte pas le format svg

```
</svg>
```



Image

- ``
 - balise **auto-fermante** -en ligne-
 - attributs **indispensables** à l'insertion d'une image :
 - **alt="..."** (**obligatoire**) définit un texte alternatif quand l'image ne s'affiche pas (malvoyant, pb d'affichage).
Renseigne l'internaute sur le contenu et/ou la fonction de l'image (exemple : retour à l'accueil).
Peut ne pas être renseigné pour une image décorative.
 - **title="..."** (**obligatoire**) définit un titre. Le texte s'affiche lors du survol de l'image avec la souris.
 - **width="..."** (**conseillé**) définit la largeur en pixel
 - **height="..."** (**conseillé**) définit la hauteur en pixel

Image

– attribut optionnel :

- ***usemap="#nommap"*** carte qui peut lui être associée, définie avec ***<map>***

– règles de bon usage :

- *à l'aide d'un logiciel de retouche d'image :*
 - *définir les dimensions de l'image*
 - *choisir un taux de compression*
 - ⇒ *réduire le poids du fichier image*
 - ⇒ *optimiser le rapport qualité/poids*
- *éviter le téléchargement d'images trop volumineuses*
Ex : logo : 10 Ko ; photo : 100 Ko
- *respecter les proportions (ratio hauteur/largeur) pour éviter les déformations*
- *regrouper les images dans un même répertoire*

Image

- `<figure> ... </figure>` -bloc-
 - utilisée pour inclure une illustration, une photographie ou encore un graphique accompagné de sa légende.
- `<figcaption> ... </figcaption>` -bloc-
 - définit si besoin la légende d'une illustration, d'une photographie ou d'un graphique.
 - doit être insérée entre `<figure>` et `</figure>`

Image

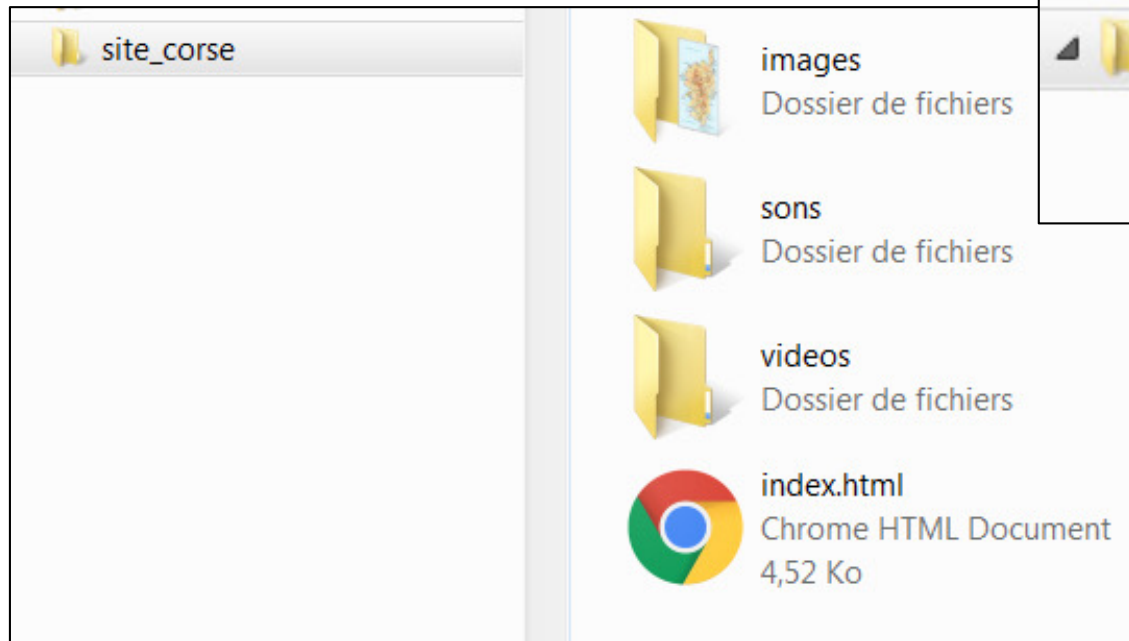
- *Exemple :*

<figure>

**

<figcaption>Carte de la région Corse</figcaption>

</figure>



Image

- *Exemple :*

```
<figure>
```

```
   Carte de la
```

```
</figure>
```

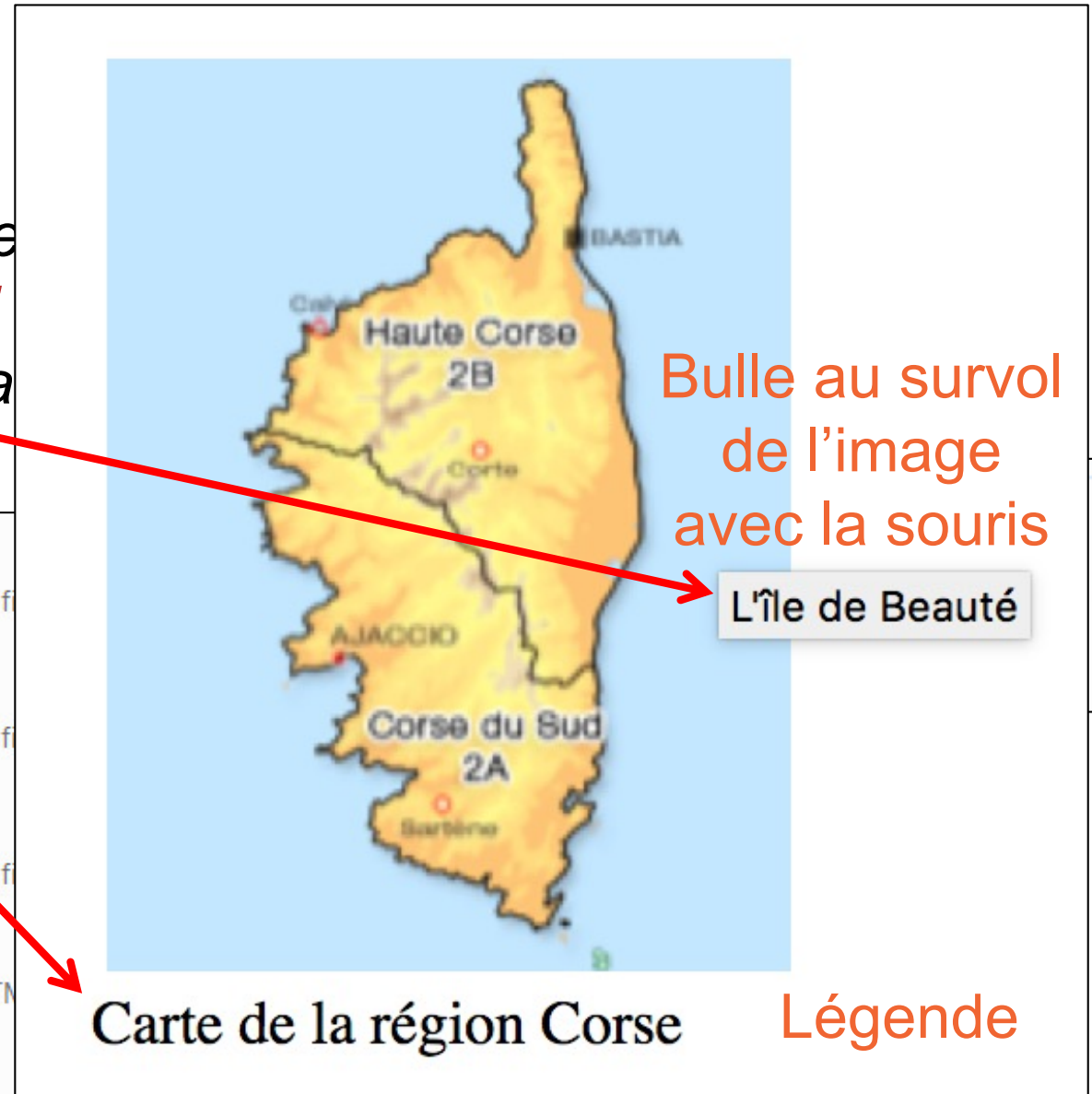
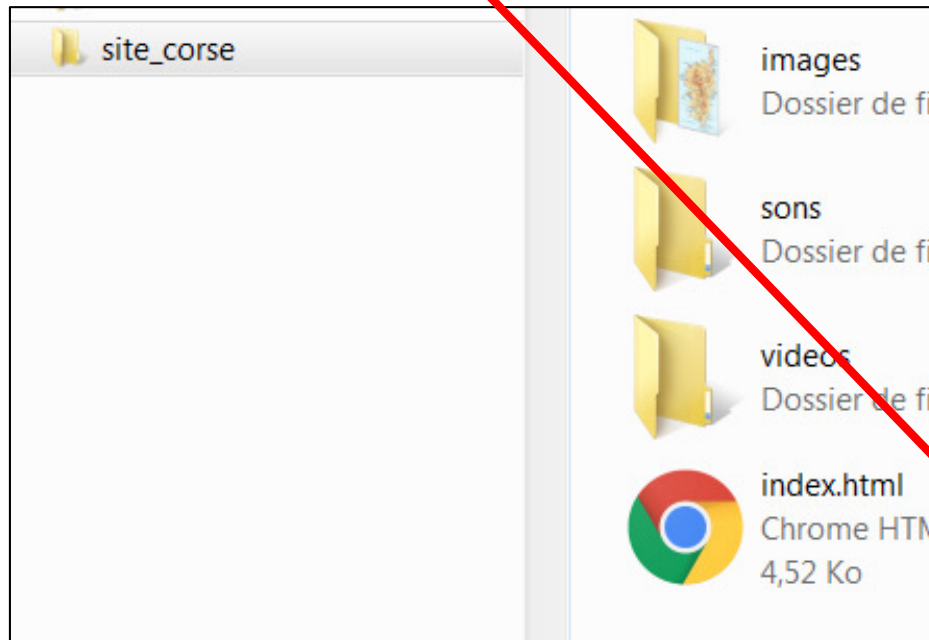


Image & lien

- Image utilisée comme pointeur d'un lien*



**Destination
du lien**

corse.html

Ouverture du lien

``

``

``

Insertion de l'image

Fermeture du lien

Image & zones réactives

- *Zones réactives utilisées comme pointeurs de liens*

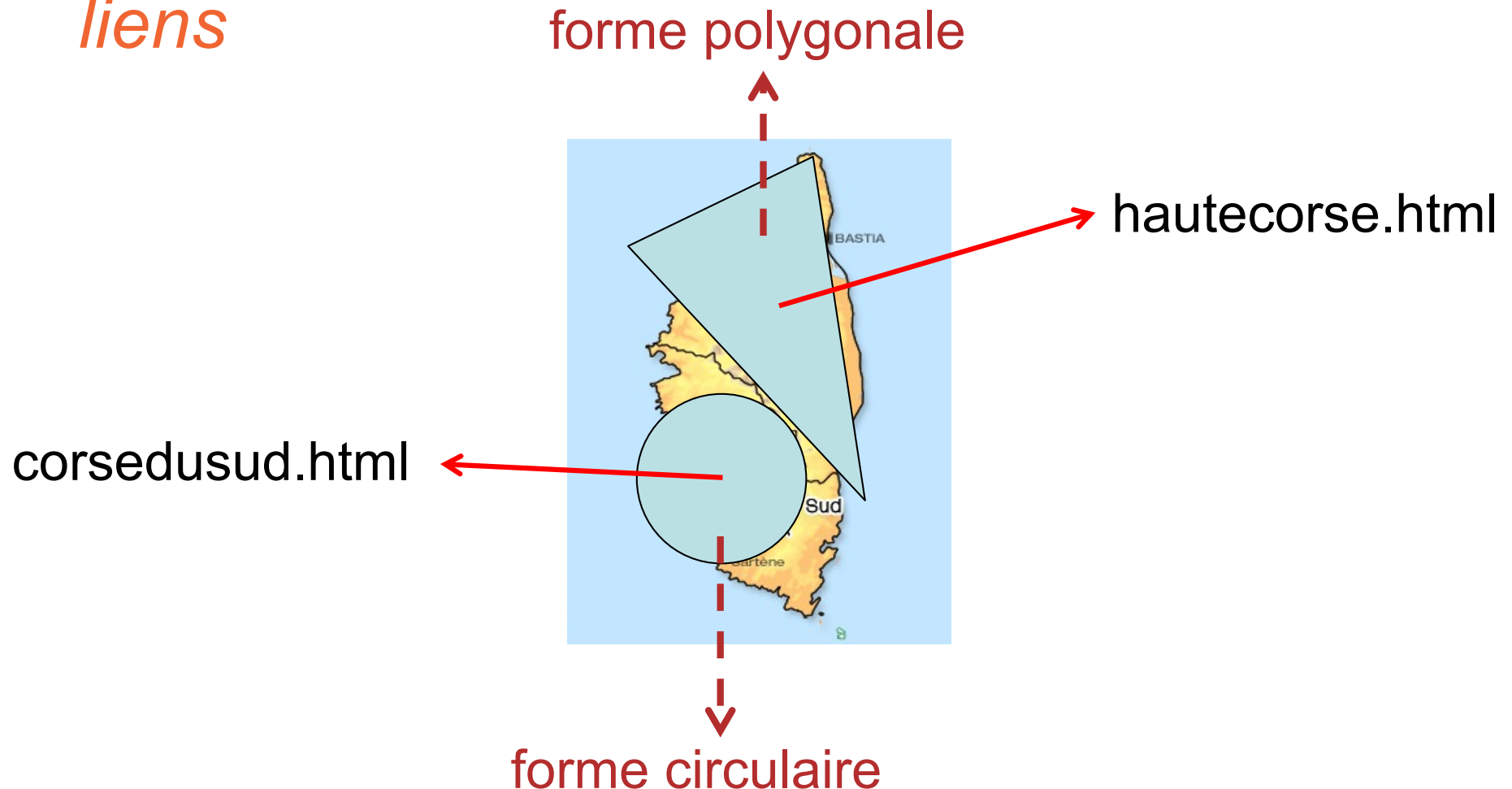


Image - carte

- `<map name="..." id="..."> ... </map>` -en ligne-
 - regroupe les zones réactives d'une même image définies à l'aide des balises `<area />`.
 - **name** et **id** sont utilisées pour nommer la carte. Ce nom est utilisé pour faire le lien avec l'image.
- **Ex :**

```

<map name="corse" id="corse" > ← lien doublé
    <area shape="poly" coords="28,46,187,8,130,166"
        href="hautecorse.html" alt="Nord" >
    <area shape="circle" coords="68,148,36"
        href="corsedusud.html" alt="Sud">
</map>
```

Image – Zone réactive

- `<area alt="..." shape="..." coords="..." >`
 - balise **auto-fermante**
 - définit une zone sensible dans une image réactive.
 - **alt="..."** est **obligatoire**.
 - **shape="..."** décrit les formes disponibles :
 - "rect" : rectangle, "circle" : cercle et "poly" : polygone
 - **coords="..."** définit les coordonnées :
 - "x1,y1,x2,y2" pour un rectangle
 - "x,y,rayon" pour un cercle
 - "x1,y1,x2,y2,x3,y3,... xn,yn" pour un polygone

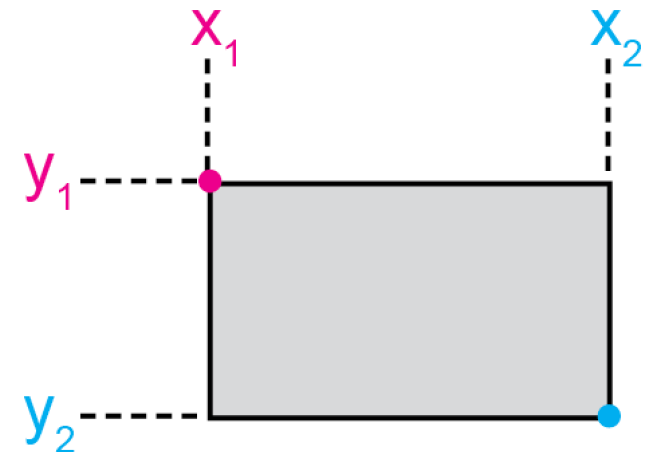
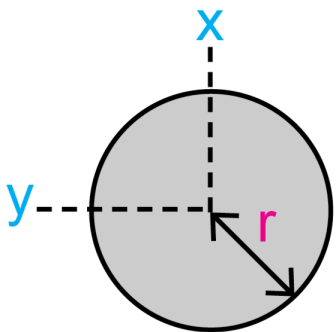
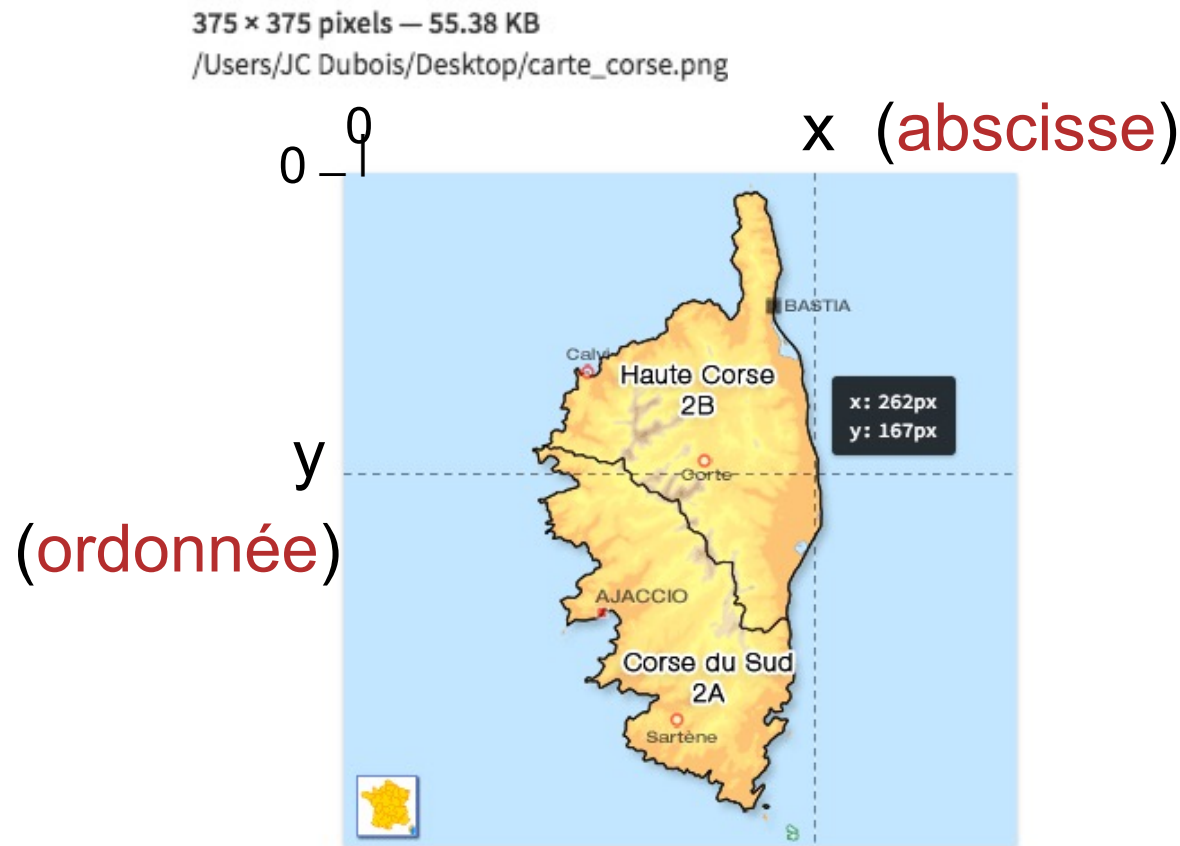


Image – Zone réactive

- Définition des coordonnées



- Survol de l'image
à l'aide de la souris

- Site dédié : <https://www.image-map.net>

Image adaptive

- `<picture>` propose plusieurs images à l'aide de 0 ou plusieurs balises `<source>` et d'une balise ``. L'image la plus appropriée est sélectionnée.

- *Exemple :*

```
<picture>  
  <source srcset="petite_img.jpg"  
           media="(max-width: 480px)" >  
  <source srcset="moy_img.jpg"  
           media="(max-width: 760px)" >  
    
</picture>
```

Tableau

- `<table> ... </table>` -bloc-
 - début et fin du tableau
- `<caption> ... </caption>` -bloc-
 - légende du tableau placée, par défaut, immédiatement après `<table>`
- `<tr> ... </tr>` -bloc-
 - définition d'une ligne (**table row**)

Tableau

- `<th> ... </th>` -bloc-
 - cellule contenant un **intitulé (table header)** placé en **tête de ligne** ou de **colonne**.
- `<td> ... </td>` -bloc-
 - cellule contenant une donnée (**table data**)
- `<th>` ou `<td>` peuvent avoir pour attribut :
 - `colspan="x"` fusion de cellules sur **x** colonnes
 - `rowspan="x"` fusion de cellules sur **x** lignes

Exemple de tableau

Code HTML

```
<table>
  <caption>Comparatif</caption>
  <tr>
    <th>Marque</th>
    <th>Modèle</th>
    <th>Prix</th>
  </tr>
  <tr>
    <td rowspan="2">Peugeot</td>
    <td>3008 HDI</td>
    <td>31.000€</td>
  </tr>
  <tr>
    <td>5008 HDI</td>
    <td>33.000€</td>
  </tr>
  <tr>
    <td colspan="2">Prix moyen</td>
    <td>32.000€</td>
  </tr>
</table>
```

Comparatif

Marque	Modèle	Prix
Peugeot	3008 HDI	31.000€
	5008 HDI	33.000€
Prix moyen		32.000€

Affichage

Options de tableau

- `<thead>... </thead>` -bloc-
 - spécifie l'**en-tête** du tableau et regroupe toutes les lignes constituant l'en-tête
- `<tbody> ... </tbody>` -bloc-
 - spécifie le **corps** du tableau
- `<tfoot> ... </tfoot>` -bloc-
 - spécifie le **pied** du tableau

Exemple de tableau

Code HTML

```
<table>
  <caption>Comparatif</caption>
  <thead>
    <tr>
      <th>Marque</th>
      <th>Modèle</th>
      <th>Prix</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td rowspan="2">Peugeot</td>
      <td>3008 HDI</td>
      <td>31.000€</td>
    </tr>
    <tr>
      <td>5008 HDI</td>
      <td>33.000€</td>
    </tr>
    <tr>
      <td colspan="2">Prix moyen</td>
      <td>32.000€</td>
    </tr>
  </tbody>
</table>
```

Comparatif

Marque	Modèle	Prix
Peugeot	3008 HDI	31.000€
	5008 HDI	33.000€
Prix moyen		32.000€

Affichage

Encodage des caractères

- Plusieurs encodages possibles :
 - Jeu de caractères ASCII standard : définit sur 7 bits et ne permet pas de coder les accents.
 - Nécessité de remplacer certains caractères :

<i>é</i> remplacé par <i>&eacute;</i> ;	<i>É</i> remplacé par <i>&Eacute;</i> ;
<i>è</i> remplacé par <i>&egrave;</i> ;	<i>ï</i> remplacé par <i>&iuml;</i> ;
<i>à</i> remplacé par <i>&agrave;</i> ;	<i>ç</i> remplacé par <i>&ccedil;</i> ;
<i>ê</i> remplacé par <i>&ecirc;</i> ;	<i>æ</i> remplacé par <i>&aelig;</i> ;
<i>ô</i> remplacé par <i>&ocirc;</i> ;	<i>œ</i> remplacé par <i>&oelig;</i> ;
 - *Jeu de caractères UTF-8* : définit les caractères grâce à une suite de 1 à 4 octets.
Il permet de coder toutes les langues.

Encodage des caractères spéciaux

- Caractères `<` , `>` , `&` et `"`
 - utilisés dans la syntaxe du code HTML
 - nécessité de les remplacer :
 - `>` remplacé par `>`; `"` remplacé par `"`;
 - `<` remplacé par `<`; `&` remplacé par `&`;
- Caractère espace
 - non pris en compte par les navigateurs
 - ` ` (**non breaking space**) permet de forcer l'insertion d'un espace dans un texte

Styles sémantiques

- `<q> ... </q>` -en ligne-
 - **citation** courte insérée dans un texte (encadrée par des guillemets).
- `<cite> ... </cite>` -en ligne-
 - encadre le **titre** d'une œuvre (livre, film...).
- `<time datetime="..."> ... </time>` -en ligne-
 - encode une **date** ou une **heure**.
 - *`datetime="2019-04-01T14:00"` indique la date ou l'heure dans le format choisi*

Styles sémantiques

- ` ... ` -en ligne-
 - met en valeur du texte
(affiché par défaut en *italique*) ;
- ` ... ` -en ligne-
 - met un texte très fortement en valeur
(affiché par défaut en **gras**) ;

Styles sémantiques

- `<mark> ... </mark>` -en ligne-
 - met en surbrillance du texte
(affiché par défaut en **surligné en jaune**) ;
- ` ... ` -en ligne-
 - délimite une partie d'un texte présentée avec une mise en page spécifique.
 - conteneur générique en-ligne.

Enrichissement du texte

- `_{...}` -en ligne-
 - Affiche le texte sous la forme d'un **indice**
 - Ex : `<p>H₂O</p>` pour H₂O
- `^{...}` -en ligne-
 - Affiche le texte sous la forme d'un **exposant**
 - Ex : `<p>x² + 3y</p>` pour x² + 3y

Enrichissement du texte

- **<ins> ... </ins>** -en ligne-
 - texte **ajouté** (généralement souligné)
- ** ... ** -en ligne-
 - texte **supprimé** (généralement ~~barré~~)
- **Attributs pour apporter de la précision :**
 - ***datetime="2019-01-01T12:30:20TZD"*** précise la date du changement (YYYY-MM-DDThh:mm:ssTZD).
 - ***cite="explication.html"*** insère une adresse externe expliquant les raisons du changement.

Insertion d'un document

- `<iframe src="..."> ... </iframe>` -en ligne-
 - insertion d'un document HTML dans un cadre réservé (*inline frame*) au sein de la page HTML
 - **`src="..."`** est **obligatoire**. Il indique l'URL du fichier source à insérer.
 - **`width="..."`** et **`height="..."`** précisent les dimensions du cadre en pixel
- **Exemple : Insertion d'une carte Google maps**
`<iframe
src="https://www.google.com/maps/embed?pb=!1m14!1m12!1m3!1d42094.55
8294754155!2d-
3.4697595!3d48.7454179999999994!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!
5e0!3m2!1sfr!2sfr!4v1504644665423" title="Carte interactive" width="600"
height="450"></iframe>`

Média - Audio

- `<audio src="..." > ... </audio>` -en ligne-
 - lecture d'un fichier **son**
 - formats acceptés par les navigateurs :

<i>Chrome : .wav, .mp3, .ogg, .flac</i>	<i>Internet Explorer : .mp3</i>
<i>Firefox : .wav, .mp3, .ogg, .flac</i>	<i>Safari : .wav, .mp3, .flac</i>
 - ***src="..."*** est **obligatoire**. Il indique l'URL du fichier source.
 - ***autoplay="autoplay"*** permet la lecture automatique du fichier.
 - ***controls="controls"*** gère l'affichage des boutons de contrôle
 - ***preload="auto"*** télécharge le son en anticipant la décision de l'internaute de l'écouter.
 - ***loop="loop"*** définit le mode de répétition en boucle.

Média - Audio

- *Exemple :*

```
<audio src="sons/musique.mp3"  
      controls="controls"  
      preload = "auto">
```

*<p>Le navigateur ne supporte pas la balise audio
Téléchargez le fichier en cliquant*

```
<a href="sons/musique.mp3">ici</a> (3 Mo)</p>  
</audio>
```

Média - Vidéo

- `<video src="..." > ... </video>` -en ligne-

- Lecture d'un fichier **vidéo**
- Formats acceptés par les navigateurs :

Chrome : .mp4, .ogg, .webM

Internet Explorer : .mp4

Firefox : .mp4, .ogg, .webM

Safari : .mp4, .mov

- **src="..."** est obligatoire. Il indique l'URL du fichier source.
- **autoplay="autoplay"** permet la lecture automatique du fichier.
- **controls="controls"** gère l'affichage des boutons de contrôle
- **preload="auto"** anticipe le téléchargement de la vidéo
- **poster="..."** affiche une image par défaut avant le lancement effectif de la vidéo.
- **width="..."** et **height = "..."** précisent les dimensions en pixel

Média - Video

- *Exemple :*

```
<video src="videos/videosuperdrole.mp4"  
      controls="controls"  
      preload="auto"  
      width="640"  
      height="480"  
      poster="videos/videosuperdrole.png">
```

```
<p>
```

Le navigateur ne supporte pas la balise video

```
</p>
```

```
</video>
```

Média - Source

- *<source src="..." type="..." >*
 - indique le chemin de plusieurs fichiers sons/vidéos pour palier les problèmes de compatibilité.
 - le navigateur joue le premier fichier compatible.
 - **Exemple :**

```
<audio controls="controls" >  
  <source src="sons/musique.wav" type="audio/wav" >  
  <source src="sons/musique.mp3" type="audio/mp3" >  
  <source src="sons/musique.ogg" type="audio/ogg" >  
  <p>Le navigateur ne supporte pas la balise audio  
    Téléchargez le fichier en cliquant  
    <a href="sons/musique.wav">ici</a> (5 Mo)</p>  
</audio>
```

Exercice

- Structure à coder
Fichier carhaix.html

- Code à vérifier

Site :

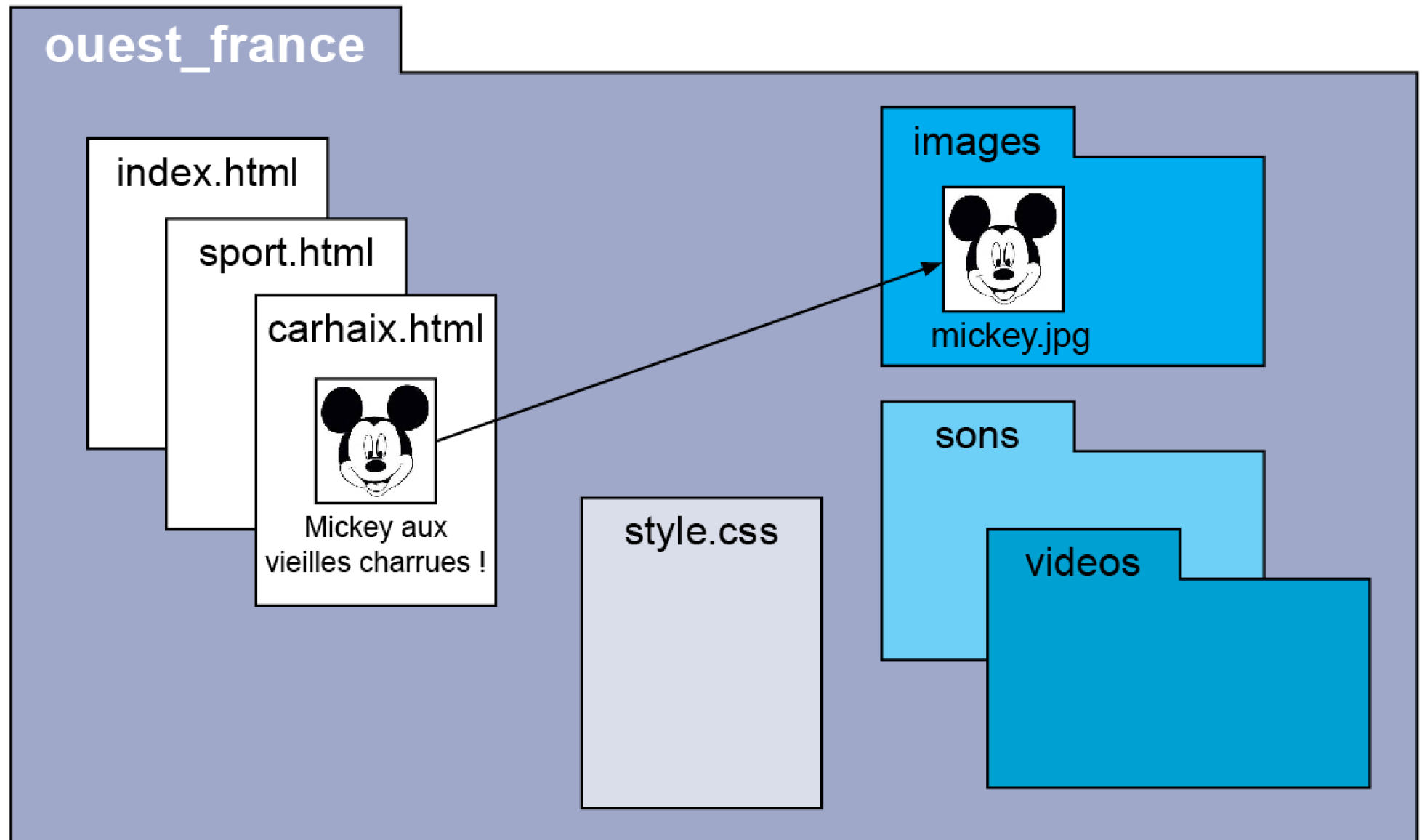
<https://validator.w3.org/>

Onglet :

Validate by upload



Organisation d'un site



Exercice

- Structure codée en HTML et visualisée via un navigateur



Exercice

- Code HTML rédigé via Visual Studio Code

```
1  <!DOCTYPE html>
2  ▼ <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
3  ▼ <head>
4      <meta charset="utf-8" />
5      <title>Structure</title>
6      <meta name="description" content="Exemple de structure HTML" />
7      <meta name="keywords" content="HTML, head, body, main, header, nav,
      section, article, aside, footer" />
8  </head>
```

Exercice

- Code HTML rédigé via Visual Studio Code (suite...)

```
9  ▼ <body>
10 ▼   <header>
11       <h1>Ouest France</h1>
12 ▼       <nav>
13           <p>Menu</p>
14       </nav>
15   </header>
16 ▼   <main>
17       <h2>Carhaix</h2>
18 ▼       <section>
19           <h3> Cinéma</h3>
20 ▼           <article>
21               <h4> Valerian</h4>
22           </article>
23 ▼           <article>
24               <h4> Seven sisters</h4>
25           </article>
26       </section>
27 ▼       <aside>
28           <p>Publicité</p>
29       </aside>
30 ▼       <section>
31           <h3> Tourisme</h3>
32           <header>Festivals</header>
33 ▼           <article>
34               <h4> Affluence record !</h4>
35           </article>
36 ▼       <footer>
37           <p> Correspondant</p>
38       </footer>
39   </section>
40 </main>
41 ▼ <footer>
42     <p>Equipe de rédaction</p>
43 </footer>
44 </body>
45 </html>
```

Les Formulaires

Formulaire

Un **formulaire** permet de recueillir les informations saisies par un utilisateur, puis de les transmettre

- **<form>** : contient l'ensemble des éléments constituant le formulaire
 - **name** : nom du formulaire
 - **method** : méthode de transfert des données
 - **"get"** : par défaut, les données suivent l'URL
 - **"post"** : les données sont envoyées dans le corps d'une requête
 - **action** :
 - URL du script ou programme qui va traiter les données
 - adresse mail du destinataire des informations (mailto:...)

Exemple :

```
<form name="questionnaire" method="post"  
      action="http://www.iut-lannion.fr/cgi-bin/script.cgi" >  
</form>
```

Formulaire

- `<label>` / libellé d'un champ de formulaire
 - `for` : correspond à la valeur de l' `ID` du champ auquel il se réfère
 - `accesskey` : définit un raccourci clavier pour un accès direct
ALT + Shift + touche (Firefox sur Linux)
CTRL + Option + touche (Firefox sur Mac)

Exemple :

```
<label for="utilisateur">Votre pseudo :</label>
```

```
<input type="text" id="utilisateur" name="pseudo"/><br />
```

Formulaire

- `<input ... />` : insère un des éléments suivant :
 - texte, mot de passe, champ masqué
 - case à cocher
 - bouton d'option
 - pièce jointe
 - bouton graphique
 - bouton de commande
 - bouton de soumission ...
- Attributs **généraux communs** :
 - **name** : (**obligatoire**) recueille la valeur du champ
 - **type** : type de champ (texte, case à cocher,...)
 - **tabindex** : position du champ dans la séquence de tabulations
- Les autres attributs varient selon le type

Formulaire

- `<input type="text" />` / champ de texte
 - `value` : valeur par défaut du champ
 - `placeholder` : indication sur le format du contenu attendu
 - `size` : longueur d'affichage du champ en nombre de caractères
 - `minlength` : nombre de caractères minimum à saisir
 - `maxlength` : nombre de caractères maximum à saisir
 - `required` : saisie obligatoire (`required="required"`)
 - mais aussi `readonly`, `disabled`, `pattern`, `inputmode`...

Exemple :

```
<label for="ville">Votre pseudo :</label>
```

```
<input type="text" name="maville" id="ville" size="30"  
maxlength="25">
```


Formulaire

- `<input type="password" />` / mot de passe
 - idem à "text" mais les caractères saisis sont masqués
 - `placeholder` n'est pas effectif
- `<input type="hidden" />` / élément masqué
 - `value` : valeur à transmettre au serveur sans que le champ ne soit visible à l'écran
- `<input type="search" />` / champ de recherche

S'utilise, plutôt seul, pour lancer une recherche dans un site. Certains navigateurs affichent une croix à droite du champ pour permettre la réinitialisation du contenu sans perdre le focus

Formulaire

- `<input type="checkbox" />` / case à cocher
 - **value** : valeur correspondante à envoyer au serveur (identique ou différente de la valeur affichée)
 - **checked** : coche la case lors du chargement de la page (**checked="checked"**)

Une ou plusieurs cases peuvent être cochées, un nom différent est donc attribué à chaque case

Exemple :

```
<input type="checkbox" name="choix1" value="1"
                                checked="checked"> Alsace
<input type="checkbox" name="choix2" value="2"> Bretagne
<input type="checkbox" name="choix3" value="3"> Corse
```

Formulaire

- `<input type="radio" />` / bouton radio
 - `value` : valeur à envoyer au serveur
 - `checked` : active l'option au chargement
 - `disabled` : désactive l'option

Un seul bouton ne peut être activé parmi un groupe de boutons

Le même nom est donné à chaque élément du groupe

Exemple :

```
<input type="radio" name="choix" value="alsace" checked="checked"> Alsace
```

☐ Bretagne

```
<input type="radio" name="choix" value="corse" disabled="disabled"> Corse
```

Formulaire

- `<input type="file" />` / sélection d'un fichier

- `accept` : définit les types MIME autorisés en précisant :
 - l'extension du fichier : .xls, .pdf, .gif...
 - ou son contenu : audio/*, video/* ou image/*

Avec le type "**file**", il faut impérativement passer l'attribut suivant au formulaire :

- `method="post"` (un fichier ne peut être envoyé dans l'URL)

- `<input type="color" />` / sélection d'une couleur

- `value` : prédéfinit une couleur en hexadécimal (#00ff00)

Le choix de la couleur peut se faire en cliquant, à l'aide d'une pipette, au sein d'une mire

Formulaire

- `<input type="url" />` / champ url
 - Une chaîne de type "http://", "https://", "ftp://" doit être renseignée, dans le cas contraire, l'internaute sera informé du manque et l'envoi non effectué.
- `<input type="email" />` / champ email
 - Si la chaîne de caractère saisie n'est pas de type mail (comportant par exemple le caractère @), l'utilisateur recevra un message d'avertissement et la soumission ne sera pas effectuée.