



Développement Web - HTML- Formulaires, Tableaux et Médias

Groupe des étudiants : CIR1

Nadine Abdallah Saab

E-Mail : nadine.abdallah-saab@isen-ouest.yncrea.fr

Formulaires

Formulaires

Objectif

Interagir avec l'utilisateur

Permet d'obtenir des réponses de l'utilisateur

Syntaxe générale :

```
<form action="..." method="..."> Eléments de formulaire </form>
```

Formulaires

- **Syntaxe générale :** `<form action="..." method="...">` *Eléments de formulaire* `</form>`
 - **L'attribut action :** le contenu de cet attribut est l'URL du programme qui traitera les informations soumises par le formulaire
 - **L'attribut method :** le contenu de cet attribut est la méthode HTTP qui sera utilisée pour envoyer les données au serveur

Formulaires

- **Syntaxe générale :** `<form action="..." method="..."> Eléments de formulaire </form>`
 - **L'attribut action :** le contenu de cet attribut est l'URL du programme qui traitera les informations soumises par le formulaire
 - **L'attribut method :** le contenu de cet attribut est la méthode HTTP qui sera utilisée pour envoyer les données au serveur
 - **GET :** les données du formulaires sont ajoutées à l'URL de l'attribut action avec '?' comme séparateur

- **Exemple :**

```
<form action="page1.html" method="get">
```

```
.....
```

```
</form>
```

Prénom:

Nom:

page1.html/?fname=pr1&lname=n1

Formulaires

- **Syntaxe générale :** `<form action="..." method="..."> Eléments de formulaire </form>`
 - **L'attribut action :** le contenu de cet attribut est l'URL du programme qui traitera les informations soumises par le formulaire
 - **L'attribut method :** le contenu de cet attribut est la méthode HTTP qui sera utilisée pour envoyer les données au serveur
 - **GET :** les données du formulaires sont ajoutées à l'URI de l'attribut action avec '?' comme séparateur
 - **POST :** les données du formulaires sont incluses dans le corps du formulaire et envoyées vers le server

- **Exemple :**

```
<form action="page1.html" method="post">
```

```
.....  
</form>
```

Prénom:

Nom:

page1.html/



Formulaires - zone de texte -

Syntaxe : `<input type="text">` permet à l'utilisateur de saisir une ligne de texte.

Elle a les attributs suivants :

- *name* : Définit le nom de la donnée (obligatoire)
- *size* : Définit la taille visible du champ en nombre de caractères
- *maxlength* : Taille maximale de caractères autorisés
- *minlength* : Taille minimale de caractères autorisés
- *value* : Définit le texte par défaut à l'intérieur du champ

Exemple :

```
<form action="page1.html" method="get">  
  Prénom:<br>  
  <input type="text" name="fname"><br>  
  Nom:<br>  
  <input type="text" name="lname"><br><br>  
  <input type="submit" value="Valider">  
</form>
```

page1.html/?fname=pr1&lname=n1

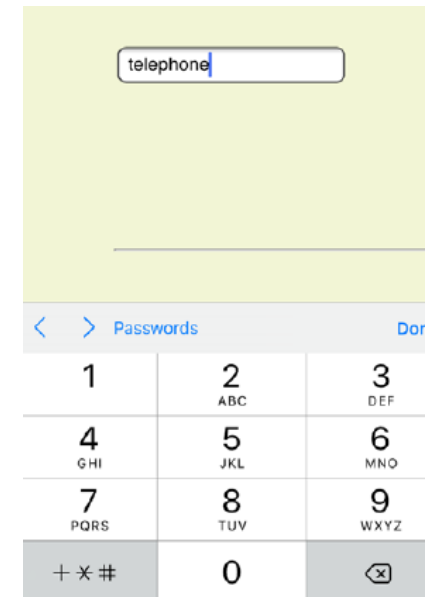
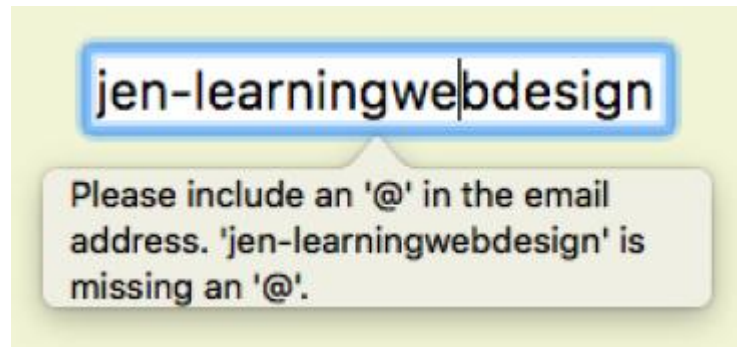
Prénom:

Nom:

Formulaires - zones de texte spécialisées -

- `type="password" | "email" | "url" | "date" | "file" | "search" | "tel" | "number" | ...`

- **Exemple :** Password:



Formulaires – Chargement d'un fichier

- `type="file"`

Obligatoire : Les attributs `method="post"` et `enctype="multipart/form-data"` sont utilisés conjointement pour permettre l'envoi de fichiers à partir d'un formulaire web vers un serveur.

- Exemple :

```
<form method="post" enctype="multipart/form-data">
  <div>
    <label for="file">Choose file to upload</label>
    <input type="file" id="file" name="file" multiple />
  </div>
  <div>
    <button>Submit</button>
  </div>
</form>
```



Choose file to upload No file chosen

Formulaires - zone de texte multiligne -

- **Syntaxe :**

```
<textarea name="comments" cols="20" rows="4" placeholder= "Veuillez  
remplir cette zone de texte..."> </textarea>
```

- **Exemple :**

```
<p>Official contest entry:<br>  
<em>Tell us why you love the band. Five winners will get backstage  
passes!</em><br>  
<textarea name="contest_entry" placeholder="50 words or less" rows="5"  
cols="50"></textarea>  
</p>
```

Official contest entry:

Tell us why you love the band. Five winners will get backstage passes!

50 words or less

Formulaires - Menu déroulant -

- Syntaxe :

`<select>...</select>`

Menu control

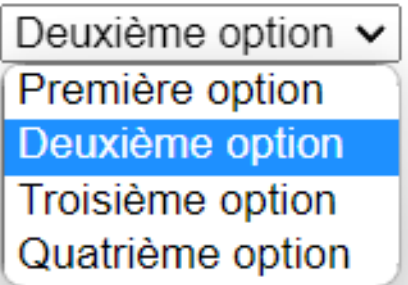
`<option>...</option>`

An option within a menu

- Exemple :

```
<label for="idliste">ma liste de choix :</label>
<SELECT NAME="maliste" id="idliste">
  <OPTION VALUE="1">Première option</OPTION>
  <OPTION VALUE="2" SELECTED>Deuxième option</OPTION>
  <OPTION VALUE="3">Troisième option</OPTION>
  <OPTION VALUE="4">Quatrième option</OPTION>
</SELECT>
```

ma liste de choix :



Formulaires - Boutons radio et checkbox -

- **Syntaxe :** la Balise `<input>` permet aussi de représenter une liste de cases à cocher ou une liste dite "bouton radio".
Spécifier l'attribut `type="checkbox"` pour une case à cocher ou `type="radio"` pour un bouton radio.
Les autres attributs :
 - `name` : Définit le nom de la donnée (obligatoire)
 - `value` : valeur de l'entrée si elle est sélectionnée
 - `checked` : permet de sélectionner le choix
- **Exemples :**

```
<u>Genre de musique favori </u><br>  
<input type="radio" name="rb_genre" value="rock"> Rock<br>  
<input type="radio" name="rb_genre" value="pop" checked> Pop<br>  
<input type="radio" name="rb_genre" value="Jazz"> Jazz<br><br>
```



Genre de musique favori

☐ Rock

☒ Pop

☐ Jazz

```
<u>Langue(s) parlée(s)</u><br>  
<input type="checkbox" name="ck_en" value="en"> Anglais<br>  
<input type="checkbox" name="ck_fr" value="fr" checked> Français<br>  
<input type="checkbox" name="ck_es" value="es"> Espagnol<br>
```



Langue(s) parlée(s)

☐ Anglais

☒ Français

☐ Espagnol

Attribut **required**

L'attribut booléen **required**, s'il est présent, indique que l'utilisateur doit spécifier une valeur pour l'entrée avant que le formulaire propriétaire puisse être soumis.

L'attribut required est pris en charge pour les éléments text, search, url, tel, email, password, date, month, week, time, datetime-local, number, checkbox, radio, les éléments file, <input>, <select> et <textarea> de contrôle de formulaire.

Formulaires - étiquette -

- **Syntaxe :**

L'élément HTML **<label>** représente une légende pour un objet d'une interface utilisateur. Rattacher un libellé à un élément de saisie ([<input>](#)) permet une association visuelle et *technique* entre eux.

Utilité :

- Vous pouvez cliquer sur le libellé pour passer le focus voire activer le champ
- Lorsque l'utilisateur a le focus sur le champ, un lecteur d'écran pourra énoncer le contenu du libellé et permettre à l'utilisateur de disposer d'un meilleur contexte.

- **Exemple :**

HTML		CSS	OUTPUT
1	<code><div class="preference"></code>		Do you like cheese? ☐
2	<code> <label for="cheese">Do you like cheese?</label></code>		
3	<code> <input type="checkbox" name="cheese" id="cheese"</code>		
4	<code></div></code>		

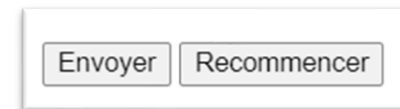
Formulaires - Boutons Submit et Reset -

- **Syntaxe :** `<input type="submit" name="btn" value="Envoyer le formulaire" >`
Bouton de validation du formulaire déclenchant l'envoi des données du formulaire vers le serveur.

`<input type="reset" name="btn" value="Recommencer" >`
Bouton permettant de réinitialiser l'ensemble des champs du formulaire avec leurs valeurs initiales.

- **Exemple :**

```
<input type="submit" name="btn_action" value="Envoyer">  
<input type="reset" name="btn_action" value="Recommencer">
```



<fieldset> : l'élément pour les ensembles de champs

Syntaxe :

```
<fieldset>
  <legend>Titre de l'ensemble </legend>
  ensemble de contrôles...
</fieldset>
```

permet de regrouper plusieurs contrôles interactifs ainsi que des étiquettes dans un formulaire HTML.
Exemple :

```
<fieldset>
  <legend>Genre de musique favori</legend>
  <input type="radio" name="rb_genre" value="rock"> Rock<br>
  <input type="radio" name="rb_genre" value="pop"> Pop<br>
  <input type="radio" name="rb_genre" value="Jazz"> Jazz<br>
</fieldset>
```

Genre de musique favori

- ☐ Rock
- ☐ Pop
- ☐ Jazz

Tableaux

<https://developer.mozilla.org/fr/docs/Web/HTML/Element/table>

Balise	Description
<code><table> ... </table></code>	Début et fin du tableau
<code><tr> ...</tr></code>	Déclaration du ligne (table row)
<code><th> ...</th></code>	Déclaration d'un en-tête (table header) : texte de la cellule mis en gras et centré
<code><td> ...</td></code>	Déclaration d'une cellule (table cell data)
<code><caption> ... </caption></code>	Titre du tableau

Tableaux

Attributs	Description
<code><td colspan="n"> ... </td></code> <code><th colspan="n"> ... </th></code>	Fusion de " n " colonnes
<code><td rowspan="m"> ...</tr></code> <code><th rowspan="m"> ... </th></code>	Fusion de " m " lignes

Tableaux

Attributs	Description
<code><td colspan="n"> ... </td></code> <code><th colspan="n"> ... </th></code>	Fusion de " n " colonnes
<code><td rowspan="m"> ...</tr></code> <code><th rowspan="m"> ... </th></code>	Fusion de " m " lignes

- Il est pratique de regrouper :
 - plusieurs `<th>` dans `<thead>`
 - plusieurs `<tr>` dans `<tbody>`
- La balise `<tfoot>` sert à insérer un *footer* au tableau (comme un résumé par exemple)

Tableaux

- Avant la version HTML5, un attribut border dans la balise table permet de générer les bordures : `<table border="5">`

 L'attribut border ainsi que d'autres attributs (bgcolor, align, ...) ne sont plus autorisés en HTML5

Tableaux

- Avant la version HTML5, un attribut border dans la balise table permet de générer les bordures : `<table border="5">`



L'attribut border ainsi que d'autres attributs (bgcolor, align, ...) ne sont plus autorisés en HTML5

- Solution :** utilisation d'un style CSS (Cascading Style Sheets) pour les tableaux (à insérer dans le `<head>`)

```
<style>
    table, td, th {border: 1px solid}
</style>
```

Exemple

```
<!DOCTYPE html>
<html>
<head>
  <title>Tableau HTML</title>
  <style> table, td, th {border: 1px solid}</style>
</head>
<body>
  <table>
    <tbody>
      <tr>
        <td>Ligne 1, Colonne 1</td>
        <td rowspan="2">Ligne 1 et 2 fusionnées, Colonne 2</td>
      </tr>
      <tr>
        <td>Ligne 2, Colonne 1</td>
      </tr>
    </tbody>
    <thead>
      <tr>
        <th>Colonne 1</th>
        <th>Colonne 2</th>
      </tr>
    </thead>
  </table>
</body>
</html>
```



Colonne 1	Colonne 2
Ligne 1, Colonne 1	Ligne 1 et 2 fusionnées, Colonne 2
Ligne 2, Colonne 1	

Médias

Médias

- Un fichier multimédia stocké dans le serveur web possède :
 - **Un type MIME** (Multipurpose Internet Mail Extensions) ¹
 - **Un suffixe** (une extension) ou plusieurs

Exemple : **fichier vidéo MPEG-4**

- **Type MIME** : video/mp4
- **Suffixes** : .mp4

¹ <https://htmlstrip.com/mime-file-type-checker>

Médias

- Un fichier multimédia stocké dans le serveur web possède :
 - **Un type MIME** (Multipurpose Internet Mail Extensions) ¹
 - **Un suffixe** (une extension) ou plusieurs

Exemple : **fichier vidéo MPEG-4**

- **Type MIME** : video/mp4
 - **Suffixes** : .mp4
- L'emplacement des informations MIME diffère d'un serveur à un autre.
Par exemple dans le cas du serveur Apache, ces informations sont stockées dans un fichier localisé dans un dossier racine appelé htaccess.

¹ <https://htmlstrip.com/mime-file-type-checker>

Médias : images

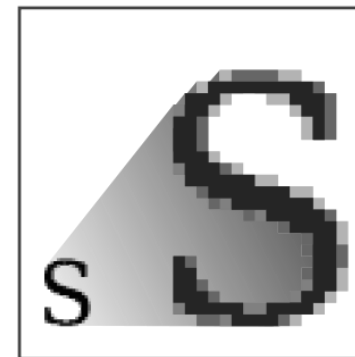
- Syntaxe : ``
 - l'`url` peut être soit absolue ou relative
 - `alt` (alternative text) : utile au cas de non chargement de l'image
- **Bonne pratique** : regrouper toutes les images dans un répertoire (nommé **images** par exemple)
- Le navigateur, au chargement de l'image, il l'enregistre dans la mémoire cache (mémoire temporaire)

Médias : images

- On peut définir une largeur et une hauteur d'une image en pixel en utilisant les attributs **width** et **height** :
 - Exemple : ``

Médias : images

- On peut définir une largeur et une hauteur d'une image en pixel en utilisant les attributs **width** et **height** :
 - Exemple : ``
- Formats des images souvent utilisés sur le web :
 - Images matricielles (bitmap)
 - PNG (Portable Network Graphics)
 - JPEG (Joint Photographic Experts Group)
 - GIF (Graphics Interchange Format)
 - Images vectorielles
 - SVG (Scalable Vector Graphics)
 - ...



Matriciel
.jpeg .gif .png



Vectoriel
.svg

Médias : images

- .svg (.svg) est le format d'image recommandé en web qui peut s'insérer :
 - **avec la balise **
 - Exemple :

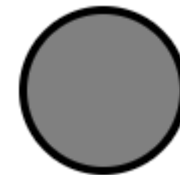
```

```

Médias : images

- .svg (.svg) est le format d'image recommandé en web qui peut s'insérer :
 - avec la balise `<img>`
 - avec la balise `<svg> ... </svg>`
 - Exemple :

```
<svg width="100" height="100">  
  <circle cx="50" cy="50" r="40" stroke="black" stroke-width="4" fill="gray" />  
</svg>
```



Médias : images

- .svg (.svg) est le format d'image recommandé en web qui peut s'insérer :
 - avec la balise `<img>`
 - avec la balise `<svg> ... </svg>`
 - **avec la balise `<object> ... </object>`**
 - Exemple :

```
<object type="image/svg+xml" data="logo-isen.svg">  
    
</object>
```

Liens dans les images

■ Principes

- Association d'une carte à une image

```
<IMG SRC="UrlImage" USEMAP="[URL]#nom de la carte">
```

■ Image-map

- Carte qui référence des zones d'une image, ainsi que les liens déclenchés sur chacune de ces parties d'images.
- Définition d'une carte

```
<MAP NAME="nom de la carte">
```

```
    <AREA ...>
```

```
    [<AREA ...>]
```

```
</MAP>
```

< AREA ... >

```
<AREA SHAPE=' rect | circle | poly | default '  
      COORDS="x1,y1,x2,y2  
              | x,y,r  
              | x1,y1,x2,y2,x3,y3 ..."  
      HREF="Url cible" >
```

SHAPE=" ... "

- rect → zone rectangulaire
- circle → disque
- poly → zone polygonale
- default → zone par défaut,
= toute la partie de l'image non recouverte par une zone définie.

< AREA ... >

```
<AREA SHAPE=" rect | circle | poly | default "  
      COORDS="x1,y1,x2,y2  
              | x,y,r  
              | x1,y1,x2,y2,x3,y3 ..."   
      HREF="Url cible" >
```



COORDS="x₁,y₁,x₂,y₂"

- Utilisé si SHAPE="rect"
- (x₁, y₁) définit le coin supérieur gauche de la zone rectangulaire,
- (x₂, y₂) définit le coin inférieur droit.

COORDS="x,y,r"

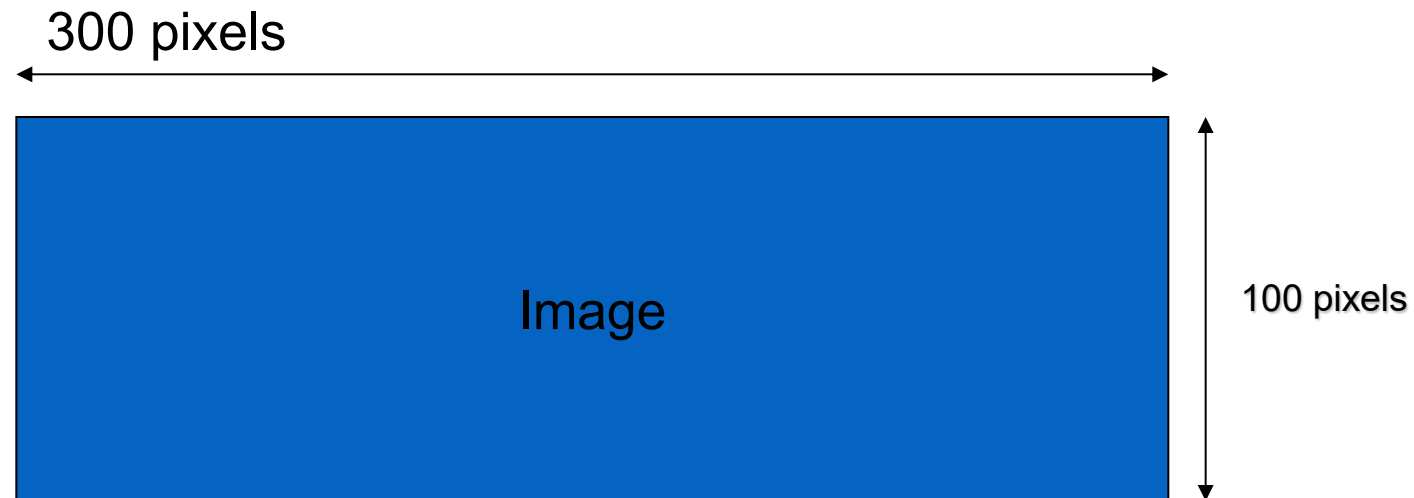
- Utilisé si SHAPE="circle"
- (x, y) désigne le centre du cercle,
- r désigne le rayon.

COORDS="x₁,y₁,x₂,y₂, ... x_n,y_n "

- Utilisé si SHAPE="poly"
- (x_i, y_i) définissent les points successifs du polygone.
- Remarque : Le polygone est fermé : (x₁, y₁) est automatiquement relié à (x_n, y_n)

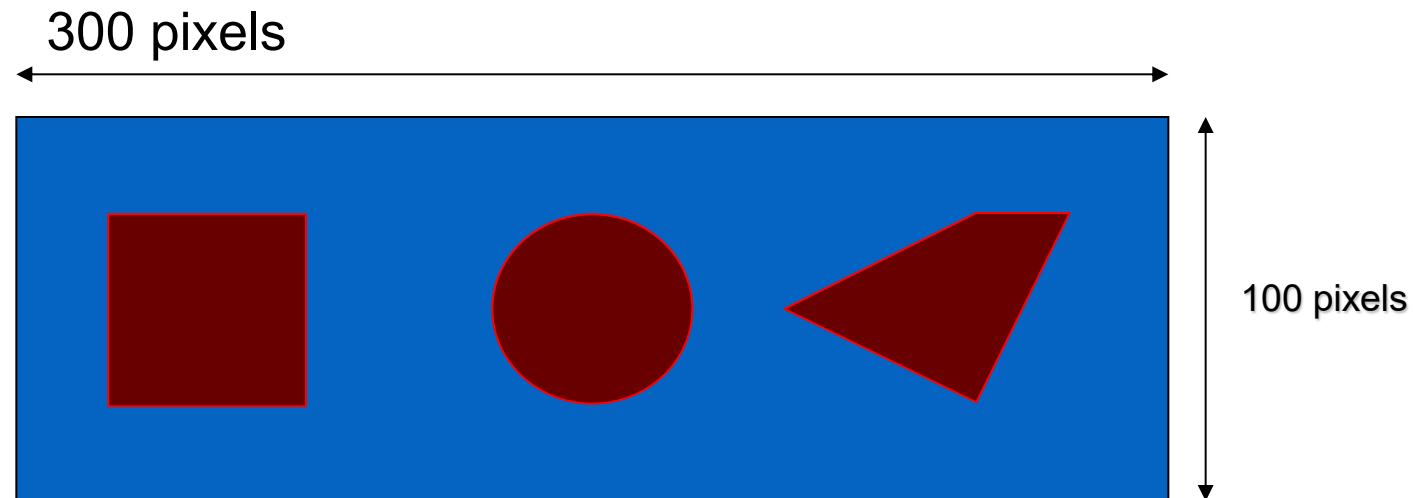
Exemple

```
<IMG SRC="path/to/image.jpg" ALT="ma carte" USEMAP="#carte_image">
<MAP NAME="carte_image">
  <AREA SHAPE="rect"    COORDS="25,25,75,75"
        HREF="rectangle.html">
  <AREA SHAPE="circle"  COORDS="150,50,25"
        HREF="cercle.html">
  <AREA SHAPE="poly"    COORDS="200,50,250,75,275,25,250,25"
        HREF="polygone.html">
</MAP>
```



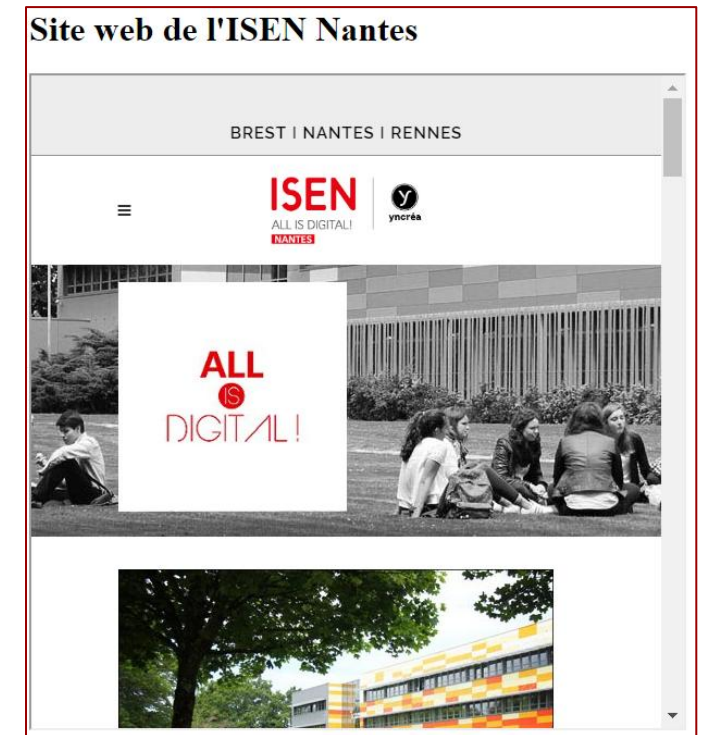
Exemple

```
<MAP NAME="carte_image">  
  <AREA SHAPE="rect"    COORDS="25,25,75,75"  
    HREF="rectangle.html">  
  <AREA SHAPE="circle"  COORDS="150,50,25"  
    HREF="cercle.html">  
  <AREA SHAPE="poly"    COORDS="200,50,250,75,275,25,250,25"  
    HREF="polygone.html">  
</MAP>
```



Médias : iframe

- La balise **iframe** permet l'affichage d'une page web à l'intérieur d'une autre page web
- Syntaxe** : `<iframe src="URL" width="200" height="200" name="nom"></iframe>`
 - URL : lien absolu ou relatif vers le contenu
 - width : largeur
 - height : hauteur
 - name : nom de l'élément (utile pour les liens avec l'attribut **target**)



Médias : Vidéo

- Aucun consensus existe entre les navigateurs sur l'extension à utiliser pour les fichiers vidéos (MP4, WebM, Ogg, ...)

- Syntaxe :

```
<video width="320" height="240" controls>  
  <source src="movie.mp4" type="video/mp4">  
  <source src="movie.ogg" type="video/ogg">  
</video>
```

- **controls** : affichage de play/pause ..
- **src** : URL relative ou absolue
- **width** : largeur
- **height** : hauteur
- **type** : MIME

Médias : Audio

- Pareil que les vidéos, aucun consensus existe entre les navigateurs sur l'extension à utiliser pour les fichiers audio (MP3, WAV, Ogg, ...)
- Syntaxe :

```
<audio controls>  
  <source src="audio.mp3" type="audio/mp3">  
  <source src="audio.ogg" type="audio/ogg">  
</audio>
```

- **controls** : affichage de play/pause ..
- **src** : URL relative ou absolue
- **type** : MIME

Médias



**Vous devez avoir le droit d'utilisation pour
chaque média (image, vidéo, audio, ...)
utilisé dans votre site web**