

# Cours

## Programmation avancée pour le web

**Credit : 4**

**Coefficient : 2**

**Mode d'évaluation :**

**- Examen (60%) - contrôle continu (40%)**

**Author:**

# Chap II HTML Rappel (suite)

L' HTML :

## Blocks et inline

les éléments ont des effets dits niveau block (**block-level** elements) ou ils ont des effets dits **inline**.

un élément à effet niveau block commence à une nouvelle ligne et occupe la totalité d'une ligne, et donc l'élément suivant débute aussi à une nouvelle ligne.

un élément inline ne commence pas à une nouvelle ligne et occupe juste l'espace nécessaire à son affichage. Il est dit aussi **<span>** element.

# Chap II HTML Rappel (suite)

L' HTML :

Blocks et inline

les éléments ont des effets dits niveau block (**block-level** elements) ou ils ont des effets dits **inline**.

**Les éléments niveau block** comme :

<div>, <section>, <ul>, <ol>, <li>, <figure>, <h1>...<h6>, <table>, <form>, <p>, <video>,...

**Les éléments inline** comme :

<a>, <br>, <i>, <img>, <input>, <label>, <time>, <button>, <sub>, <sup>, <map>, <acronym>,...

# Chap II HTML Rappel (suite)

L' HTML :

## Blocks et inline

l'élément **<div>** utilisé comme conteneur (container) pour ensemble d'éléments pour faire un block gérable du point de vue propriétés.

```
<div style="background-color:green;color:white;padding:20px;">  
  <h2>London</h2>  
  <p>London is the capital city of England. It is the most populous city in  
the United Kingdom.  
</p>  
</div>
```

## Chap II HTML Rappel (suite)

L' HTML :

Blocks et inline

l'élément `<div>` utilisé comme conteneur (container) pour ensemble d'éléments pour faire un block gérable du point de vue propriétés.

### London

London is the capital city of England. It is the most populous city in the United Kingdom,

## Chap II HTML Rappel (suite)

L' HTML :

Blocks et inline

l'élément **<span>** utilisé pour un mark up juste d'une partie de document.

```
<p>My mother has <span style="color:blue;font-weight:bold">blue</span> eyes and my father has <span style="color:darkolivegreen;font-weight:bold">dark green</span> eyes.</p>
```

## Chap II HTML Rappel (suite)

L' HTML :

Blocks et inline

l'élément **<span>** utilisé pour un mark up juste d'une partie de document.

My mother has **blue** eyes and my father has **dark green** eyes.

# Chap II HTML Rappel (suite)

L' HTML :

L'attribut **class** utilisé pour qualifier un élément.

plusieurs élément peuvent être qualifiés comme étant d'une même class.

exemple :

```
<!DOCTYPE html>
<html>
<head>
<style>
.city {
  background-color: tomato;
  color: white;
  border: 2px solid black;
  margin: 20px;
  padding: 20px;
}
</style>
</head>
<!-- à suivre ..... -->
```



# Chap II HTML Rappel (suite)

L' HTML :

L'attribut **class** utilisé pour qualifier un élément.

Si on définit un style pour cette classe, tout les éléments appartenant à cette s'attribuerons ce style .

exemple :

```
<!-- la suite -->
<body>
<div class="city">
  <h2>London</h2>
  <p>London is the capital of England.</p>
</div>
<div class="city">
  <h2>Paris</h2>
  <p>Paris is the capital of France.</p>
</div>
</body>
</html>
```

## Chap II HTML Rappel (suite)

L' HTML :

L'attribut **class** utilisé pour qualifier un élément.

Si on définit un style pour cette classe, tout les éléments appartenant à cette s'attribuerons ce style .

exemple :

London

London is the capital of England

Paris

Paris is the capital of France

## Chap II HTML Rappel (suite)

L' HTML :

L'attribut **id** utilisé pour identifier et qualifier un seul élément.

on ne peut attribuer un id à plus qu'un élément.

Exemple :

```
<!DOCTYPE html>
<html>
<head>
<style>
#myHeader {
  background-color: lightblue;
  color: black;
  padding: 40px;
  text-align: center;
}
</style>
</head>
<!-- à suivre ..... -->
```

## Chap II HTML Rappel (suite)

L' HTML :

L'attribut **id** utilisé pour identifier et qualifier un seul élément.

on ne peut attribuer un id à plus qu'un élément.

Exemple :

```
<!-- la suite -->
```

```
<body>
```

```
<h1 id="myHeader">My Header</h1>
```

```
</body>
```

```
</html>
```

## Chap II HTML Rappel (suite)

L' HTML :

L'attribut **id** utilisé pour identifier et qualifier un seul élément.

on ne peut attribuer un id à plus qu'un élément.

Exemple :



My Header

# Chap II HTML Rappel (suite)

L' HTML :

L'élément **<iframe>** pour inline frame permet d'afficher une page au sein d'une autre.

```
<!DOCTYPE html>
<html>
<body>

<h2>HTML Iframes</h2>
<p>You can use the height and width attributes to specify the size of the
iframe:</p>

<iframe src="demo_iframe.htm" height="200" width="300" title="Iframe
Example"></iframe>

</body>
</html>
```

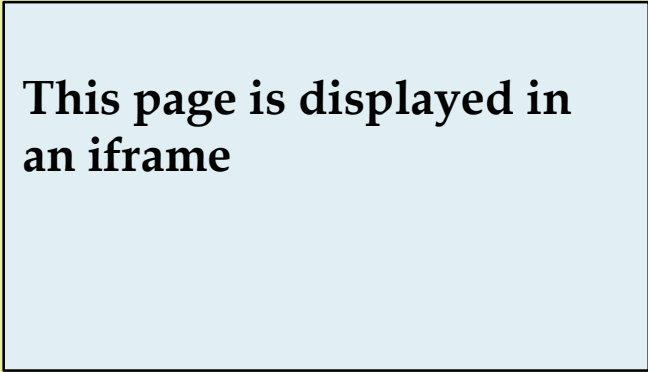
## Chap II HTML Rappel (suite)

L' HTML :

L'élément **<iframe>** pour inline frame permet d'afficher une page au sein d'une autre.

### HTML Iframes

You can use the height and width attributes to specify the size of the iframe:



**This page is displayed in  
an iframe**

# Chap II HTML Rappel (suite)

L' HTML :

## Les formulaires avec <form>

utilisés pour collecter des données saisies par l'utilisateur et envoyés le plus souvent au serveur.

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



# Chap II HTML Rappel (suite)

L' HTML :

## Les formulaires avec <form>

utilisés pour collecter des données saisies par l'utilisateur et envoyés le plus souvent au serveur.

exemple

```
<h2>HTML Forms</h2>
<form action="/action_page.php">
<label for="fname">First name:</label><br>
<input type="text" id="fname" name="fname"
value="John"> <br>
<label for="lname">Last name:</label><br>
<input type="text" id="lname" name="lname"
value="Doe"><br><br>
<input type="submit" value="Submit">
</form>
<p> click the "Submit" button. Data will be
sent to a page called "/action_page.php".</p>
```

# Chap II HTML Rappel (suite)

L' HTML :

## Les formulaires avec <form>

utilisés pour collecter des données saisies par l'utilisateur et envoyés le plus souvent au serveur.

exemple

### HTML Forms

First name:

Last name:

click the "Submit" button. Data will be sent to a page called "/action\_page.php".

ou bien [exécute](#)

# Chap II HTML Rappel (suite)

L' HTML :

**Les formulaires avec <form>**

les types pour l'élément **<input>**

- `<input type="text">` champs de saisie pour texte.
- `<input type="radio">` radio bouton.
- `<input type="checkbox">` checkbox.
- `<input type="submit">` bouton d'envoi.
- `<input type="button">` bouton cliquable.

# Chap II HTML Rappel (suite)

L' HTML :

## Les formulaires avec <form>

- <input type="radio"> radio bouton.

Exemple

```
<h2>Radio Buttons</h2>
<p>Choose your favorite Web language:</p>
<form>
<input type="radio" id="html" name="fav_language" value="HTML">
<label for="html">HTML</label><br>
<input type="radio" id="css" name="fav_language" value="CSS">
<label for="css">CSS</label><br>
<input type="radio" id="javascript" name="fav_language"
value="JavaScript">
<label for="javascript">JavaScript</label>
</form>
```

# Chap II HTML Rappel (suite)

L' HTML :

## Les formulaires avec <form>

- <input type="radio"> radio bouton.

Exemple

### Radio Buttons

Choose your favorite Web language:

- HTML
- CSS
- JavaScript

ou bien [exécute](#)

# Chap II HTML Rappel (suite)

L' HTML :

## Les formulaires avec <form>

- <input type="checkbox"> checkbox bouton.

Exemple

```
<h2>Checkboxes</h2>
```

```
<p>The <strong>input type="checkbox"</strong> defines a  
checkbox:</p>
```

```
<form action="/action_page.php">
```

```
  <input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">
```

```
  <label for="vehicle1"> I have a bike</label><br>
```

```
  <input type="checkbox" id="vehicle2" name="vehicle2" value="Car">
```

```
  <label for="vehicle2"> I have a car</label><br>
```

```
  <input type="checkbox" id="vehicle3" name="vehicle3" value="Boat">
```

```
  <label for="vehicle3"> I have a boat</label><br><br>
```

```
  <input type="submit" value="Submit">
```

```
</form>
```

# Chap II HTML Rappel (suite)

L' HTML :

## Les formulaires avec <form>

- <input type="radio"> radio bouton.

Exemple

## Checkboxes

The `input type="checkbox"` defines a checkbox:

- I have a bike
- I have a car
- I have a boat

ou bien [exécute](#)

# Chap II HTML Rappel (suite)

L' HTML :

## Les formulaires avec <form>

Les attributs :

- L'attribut **name** du tag <input> .  
Si l'attribut **name** du tag <input> est omis (non déclarée) le champs ne sera envoyé.
- L'attribut **action** de l'élément <form> spécifie l'action à réaliser suite au click. Le plus souvent les données sont envoyées à un fichier script coté serveur.
- L'attribut **target** de l'élément <form> déjà vue spécifie si l'affichage cible tiendra dans une nouvelle fenêtre, la même ou au top.



# Chap II HTML Rappel (suite)

L' HTML :

## Les formulaires avec <form>

Les attributs :

- L'attribut **method** de l'élément <form> spécifie la méthode HTTP à utiliser. Couramment **method="get"** ou **method="post"**.
- L'attribut **target** de l'élément <form> déjà vue spécifie si l'affichage cible tiendra dans une nouvelle fenêtre, la même ou au top.

# Chap II HTML Rappel (suite)

L' HTML :

## Les formulaires avec <form>

Les attributs :

- L'attribut **method="get"**
  - Les données envoyées sont ajoutées à l'url sous forme de paires (name/value).
  - Ces données sont visibles sur l'url, donc non pas utiliser cette méthode avec des données confidentielles .
  - L'url est de longueur limitée à 2kcaractères.
  - Elle est préférée pour les données non confidentielles.

# Chap II HTML Rappel (suite)

L' HTML :

## Les formulaires avec <form>

Les attributs :

- L'attribut **methode=" post"**
  - Les données envoyées sont ajoutées au niveau de body de la requête.
  - La longueur n'est pas à priori limitée et donc on peut envoyer un volume de données assez important.
  - Elle est préférée pour les données personnelles confidentielles.

# Chap II HTML Rappel (suite)

L' HTML :

**Les formulaires avec <form>**

Les attributs :

- L'attribut **autocomplete="on/off"**
  - Le browser fait rappeler les champs déjà saisie s'il y avait déjà tôt une première saisie.

# Chap II HTML Rappel (suite)

L' HTML :

**Les formulaires avec <form>**

Les éléments :

- **<input>**
- **<label>**
- **<select>**
- **<textarea>**
- **<button>**
- **<fieldset>**
- **<legend>**
- **<datalist>**
- **<output>**
- **<option>**

# Chap II HTML Rappel (suite)

L' HTML :

**Les formulaires avec <form>**

L'élément **<select>** : liste déroulante.

```
<h2>The select Element</h2>
<p>The select element defines a drop-down list:</p>
<form action="/action_page.php">
  <label for="cars">Choose a car:</label>
  <select id="cars" name="cars">
    <option value="volvo">Volvo</option>
    <option value="saab">Saab</option>
    <option value="fiat">Fiat</option>
    <option value="audi">Audi</option>
  </select>
  <input type="submit">
</form>
```

# Chap II HTML Rappel (suite)

L' HTML :

**Les formulaires avec <form>**

- L'élément `<select>` : liste déroulante.

## The select Element

The select element defines a drop-down list:

Choose a car:

ou bien [exécute](#)

# Chap II HTML Rappel (suite)

L' HTML :

## Les formulaires avec <form>

L'élément **<textarea>** : champs de saisie pour texte multi-ligne.

```
<h2>Textarea</h2>
<p>The textarea element defines a multi-line input
field.</p>

<form action="/action_page.php">
<textarea name="message" rows="10" cols="30">The cat
was playing in the garden.</textarea>
<br><br>
<input type="submit">
</form>
```



# Chap II HTML Rappel (suite)

L' HTML :

**Les formulaires avec <form>**

- L'élément `<textarea>` :

## Textarea

The textarea element defines a multi-line input field.

The cat was playing in the  
garden.

submit

# Chap II HTML Rappel (suite)

L' HTML :

## Les formulaires avec <form>

L'élément **<fieldset>** et **<legend>**: fieldset pour grouper ensemble des éléments d'un form.

```
<h2>Grouping Form Data with Fieldset</h2>
<p>The fieldset element is used to group related data in a form,
and the legend element defines a caption for the fieldset
element.</p>
<form action="/action_page.php">
  <fieldset style="width:40%">
    <legend>Personalia:</legend>
    <label for="fname">First name:</label><br>
    <input type="text" id="fname" name="fname" value="John"><br>
    <label for="lname">Last name:</label><br>
    <input type="text" id="lname" name="lname" value="Doe"> <br>
    <input type="submit" value="Submit">
  </fieldset>
</form>
```

# Chap II HTML Rappel (suite)

## Grouping Form Data with Fieldset

The fieldset to group related data, and the legend defines a caption for the fieldset element.

— Personalia —

First name

Last name

Ou bien [exécute](#)

# Chap II HTML Rappel (suite)

L' HTML :

## Les formulaires avec <form>

L'élément **<datalist>** : spécifie une liste d'options (choix) prédéfinis pour un input.

```
<h2>The datalist Element</h2>
<p>datalist element specifies a list of pre-defined options
for an input.</p>
<form action="/action_page.php">
  <input list="browsers" name="browser">
    <datalist id="browsers">
      <option value="Firefox">
      <option value="Chrome">
      <option value="Opera">
      <option value="Safari">
    </datalist>
  <input type="submit">
</form>
```

## Chap II HTML Rappel (suite)

L' HTML :

### Les formulaires avec <form>

L'élément **<datalist>** : spécifie une liste d'options (choix) prédéfinis pour un input.

**Remarquer** que l'élément input a comme attribut list qui doit avoir la valeur de l'id de datalist.

# Chap II HTML Rappel (suite)

L' HTML :

**Les formulaires avec <form>**

L'élément **<datalist>** : spécifie une liste d'options (choix) prédéfinis pour un input.

## The datalist Element

datalist element specifies a list of pre-defined options for an input

Firefox	submit
Chrome	
Opera	
Safari	

ou bien [exécute](#)

# Chap II HTML Rappel (suite)

L' HTML :

## **HTML Media**

Le multimédia rassemble tout ce que l'utilisateur a besoin pour voir et écouter comme images, animation, sons, music, films et videos.

Les formats video:

MPEG, AVI, WMV, QuickTime, Flash, Ogg, WebM, MP4 et aussi d'autres.

Les formats audio:

MIDI, WMA, AAC, WAV, Ogg, MP3, MP4 et aussi d'autres.

# Chap II HTML Rappel (suite)

L' HTML :

## HTML Media

L'élément **<video>** (HTML5)

```
<!DOCTYPE html>
<html>
<body>
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  Your browser does not support the video tag.
</video>
</body>
</html>
```

[on a browser](#)



# Chap II HTML Rappel (suite)

L' HTML :

## HTML Media

L'élément **<video>** (HTML5)

l'attribut `controls` ajoute les controles : play, pause, stop,..

```
<!DOCTYPE html>
<html>
<body>
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogv" type="video/ogg">
  Your browser does not support the video tag.
</video>
</body>
</html>
```

# Chap II HTML Rappel (suite)

L' HTML :

## HTML Media

L'élément **<video>** (HTML5)

Et l'attribut `autoplay` lance la video automatiquement.

```
<!DOCTYPE html>
<html>
<body>
<video width="320" height="240" autoplay>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  Your browser does not support the video tag.
</video>
</body>
</html>
```

[on a browser](#)

# Chap II HTML Rappel (suite)

L' HTML :

## HTML Media

L'élément **<audio>** jouer les fichiers audio.

```
<!DOCTYPE html>
<html>
<body>

<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
  Your browser does not support the audio element.
</audio>

</body>
</html>
```

# Chap II HTML Rappel (suite)

L' HTML :

## HTML Media

Les plug-ins et l'élément **<embed>**

Programmes qui étendent les fonctionnalités des browsers et font tourner par exemple des appelets java, des ActiveX, des movies flash ou des scan-virus.

Ils ont tendance à ne pas être supporté par les navigateurs.

l'élément **<object>** est supporté aussi que **<embed>**.

# Chap II HTML Rappel (suite)

L' HTML :

## HTML Graphics (HTML5)

L'élément **<canvas>** utilisé pour dessiner du graphique sur une page html moyennant javascript.

**<canvas>** est un conteneur par défaut sans contour et sans contenu.

```
<!DOCTYPE html>
<html>
<body>
<canvas id="myCanvas" width="200" height="100"
style="border:1px solid #000000;">
Your browser does not support the HTML canvas tag.
</canvas>
</body>
</html>
```

# Chap II HTML Rappel (suite)

L' HTML :

## HTML Graphics

L'élément **<canvas>** utilisé pour dessiner du graphique sur une page html moyennant javascript.

`<canavas>` est un conteneur par défaut sans contour et sans contenu.



# Chap II HTML Rappel (suite)

L' HTML :

## HTML Graphics

L'élément **<canvas>** (HTML5) utilisé pour dessiner du graphique sur une page html moyennant javascript.

exemple <script>

```
<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.moveTo(0, 0);
ctx.lineTo(200, 100);
ctx.stroke();
</script>
```

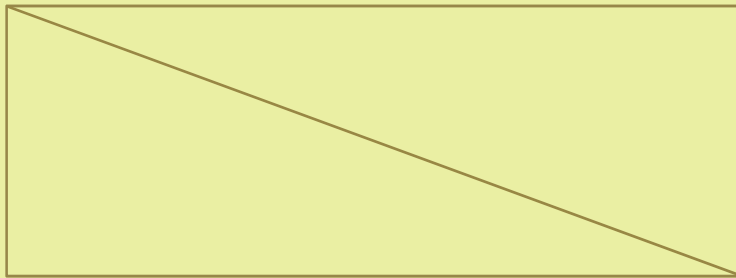
# Chap II HTML Rappel (suite)

L' HTML :

## HTML Graphics

L'élément **<canvas>** utilisé pour dessiner du graphique sur une page html moyennant javascript.

exemple `<script>`





# Chap II HTML Rappel (suite)

## L' HTML : HTML APIs

Géolocalisation Cette API permet de repérer la position d'un utilisateur. La méthode à utiliser est : **getCurrentPosition()**.

```
<script>
var x = document.getElementById("demo");
function getLocation() {
  if (navigator.geolocation) {
    navigator.geolocation.getCurrentPosition(showPosition);
  } else {
    x.innerHTML = "Geolocation is not supported by this
browser.";
  }
}
function showPosition(position) {
  x.innerHTML = "Latitude: " + position.coords.latitude +
  "<br>Longitude: " + position.coords.longitude;
}
</script>
```

# Chap II HTML Rappel (suite)

## L' HTML : HTML APIs

Géolocalisation Cette API permet de repérer la position d'un utilisateur. La méthode à utiliser est :

getCurrentPosition().

Elle renvoie un objet « position » avec les propriétés :

**coords.latitude** : latitude (décimale renvoyée toujours)

**coords.longitude** : longitude (décimale renvoyée toujours)

**coords.accuracy**: accuracy (renvoyée toujours)

**coords.altitude**: altitude en metres au dessus de la mer (if available)

**coords.altitudeAccuracy** : altitude accuracy (if available)

**coords.heading** : en degrés par rapport au nord (if available)

**coords.speed** : speed (metres/second returned if available)

**timestamp** :date et heure de la réponse ( if available)

# Chap II HTML Rappel (suite)

## L' HTML : HTML APIs

Drag and drop API glisser-déposer : Cette API permet de déplacer des éléments. La méthode à utiliser est :

```
<!DOCTYPE HTML>
<html>
<head>
<script>
function allowDrop(ev) {
    ev.preventDefault();
}
function drag(ev) {
    ev.dataTransfer.setData("text", ev.target.id);
}
function drop(ev) {
    ev.preventDefault();
    var data = ev.dataTransfer.getData("text");
    ev.target.appendChild(document.getElementById(data));
}
</script>
</head>
<!-- à suivre .... -->
```

# Chap II HTML Rappel (suite)

## L' HTML : HTML APIs

Drag and drop API glisser-déposer : Cette API permet de déplacer des éléments. La méthode à utiliser est :

```
<!-- la suite ... -->
<body>

<div id="div1" ondrop="drop(event)" ondragover="allowDrop(event)"></div>



</body>
</html>
```