



Technologies du WEB

HTML5

Tarek AIT BAHA

t.aitbaha@uiz.ac.ma

ESEF AGADIR - Université IBN ZOHR

Mode d'évaluation



- **Contrôles continus** : Tests, devoirs, Comptes-rendus des TP, Mini projet.
Présentent 40% de la note du module.
- **Examen final** : Présente 50% de la note du module.
- **Assiduité, présence et discipline** : Présente 10% de la note du module.

Le module est validé si sa moyenne est supérieure ou égale à 10/20.

L'étudiant ne peut dépasser 3 absences. Si un étudiant cumule 4 absences, il ne sera pas autorisé à passer l'examen.



Le **Web** (*Tim-Berners-lee* 1990) :

- Repose sur Internet.
- Système d'information réparti en **documents Web**.
- Basé sur la notion d'**hyperliens** permettant de naviguer entre les documents Web.
- S'appuie sur des protocoles de communication **HTTP** et **HTTPS**.
- Les adresses qui permet de repérer les documents: **URL**.
- Des **navigateurs** pour afficher et interpréter ces documents.
- Des langages pour créer les documents: **HTML**, **CSS** et **JavaScript**.

Plan du cours



- Introduction
- Syntaxe
- Élément HTML5
- Les formulaires
- Les frames

Plan du cours



- **Introduction**
- Syntaxe
- Éléments HTML5
- Les formulaires
- Les frames

Introduction



L'acronyme **HTML**

Hyper **T**ext **M**arkUp **L**angage

HTML est un langage de description qui va nous permettre de :

- Décrire l'aspect d'un document.
- Inclure des informations variées (*textes, images, sons, etc*).
- Établir des relations cohérente entre ces informations grâce au liens hypertexte.

C'est donc en HTML que vous écrirez ce qui doit être affiché sur la page : du texte, des liens, des images...

Introduction - Les versions d'HTML

Au fil du temps, le langage HTML a beaucoup évolué. Dans la toute première version de **HTML (HTML 1.0)** il n'était même **pas possible d'afficher des images** !

Les versions de HTML :

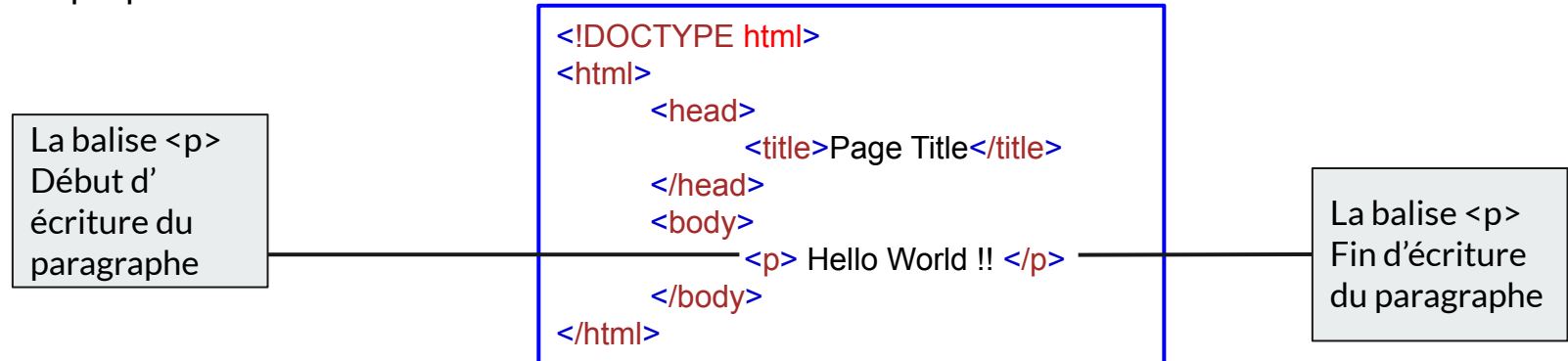
- **HTML 1** : C'est la toute première version créée par Tim Berners-Lee en 1991.
- **HTML 2** : La deuxième version du **HTML** apparaît en **1994** et prend fin en **1996**. C'est cette version qui posera en fait les bases des versions suivantes du HTML. Les règles et le fonctionnement de cette version sont donnés par le **W3C** (tandis que la première version avait été créée par un seul homme).
- **HTML 3** : Apparue en **1996**, cette nouvelle version du HTML **rajoute de nombreuses possibilités au langage** comme les **tableaux**, les **applets**, les **scripts**, le **positionnement du texte autour des images**, etc.

Introduction - Les versions d'HTML

- **HTML 4** : Elle apparaît pour la première fois en **1998** et propose l'utilisation de frames (*qui découpent une page web en plusieurs parties*), des **tableaux plus complexes**, des **améliorations** sur les **formulaires**, etc. Mais surtout, cette version permet pour la première fois d'exploiter des **feuilles de style**, notre fameux **CSS** !
- **HTML 5** : c'est la dernière version. De plus en plus répandue, elle fait beaucoup parler d'elle car elle apporte de nombreuses améliorations comme la possibilité d'inclure facilement des vidéos, un meilleur agencement du contenu, de nouvelles fonctionnalités pour les formulaires, etc.

Introduction

- La description d'un document **HTML** passe par l'utilisation de **Balises**.
- Une balise est délimitée par les signes "<" et ">" entre lesquels figure le nom de la balise. Par exemple, la balise de retour à la ligne est **
**.
- La plupart du temps, on utilise une balise de début et une balise de fin, qui définissent les propriétés de l'intervalle.



Plan du cours



- Introduction
- **Syntaxe**
- Élément HTML5
- Les formulaires
- Les frames

Syntaxe



La **syntaxe** définit :

- La structure du document.
- Les règles d'écriture.

Syntaxe - Structure minimale

```
<!DOCTYPE html>
<html>
  <!-- en-tête du document -->
  <head>
    <title>Structure minimale</title>
    <meta charset="utf-8">
  </head>
  <!-- corps du document -->
  <body>
    <!-- Contenu de la page -->
  </body>
</html>
```

Doctype : Il permet au navigateur de savoir quel langage il devra interpréter.

Html : C'est le premier marqueur que l'on trouve dans un document HTML et qui va recueillir les deux principaux éléments de la hiérarchie "Head" et "Body".

Head : L'en-tête du document est réservée aux méta-informations (les informations relatives au document) comme son titre.

Title : Contient le titre du document HTML.

Meta "Charset" : Permet de préciser l'encodage des caractères utilisé.

Body : Le corps du document, ce qui sera effectivement affiché par le navigateur Web,

Syntaxe - Règles d'écriture

- À toute balise ouvrante `<element>` doit être associé une balise fermante `</element>`.
- Il existe des balises auto-fermante `<element/>` (Ex. ``, ...)
- Les éléments ne doivent pas se chevaucher: Premier ouvert → Dernier fermé

Ex: `<p>` Début `<code>` Chevauchement `</p>` Suite `</code>`

Illégal !!

`<p>` Début `<code>` Chevauchement `</code>` Suite `</p>`

- Le contenu d'un élément peut être constitué d'autres éléments.
- Les commentaires : `<!-- Ceci est un commentaire -->`

Syntaxe - Règles d'écriture

- Les anciennes versions d'**HTML** nécessitent l'utilisation d'entités :

é : é ; è : è ; ê : ê ; à : à ; É : É ;

- Maintenant l'encodage est supporté. Bonne pratique : **UTF-8**
- L'encodage doit être défini à :

```
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
```

Syntaxe - Règles d'écriture

- Il est possible de transmettre des informations à traiter aux balises :
 - Balise de début :

```
<balise[ attribut1=valeur1[ attribut2=valeur2 ...]]>...</balise>
```

- Balise auto-fermante :

```
<balise[ attribut1=valeur1[ attribut2=valeur2 ...]]/>
```

```

```

Syntaxe - Règles d'écriture

- Il existe 3 types d'éléments :
 - **Élément bloc (*div*)** : Élément précédé et suivi d'un saut de ligne. Il forme une boîte dans lequel est inclus du texte ou d'autres éléments.

Ex.: Les paragraphe, les tableaux, ...

- **Élément inline (*span*)** : Élément qui s'insère dans le fil du texte et ne peut contenir que du texte ou d'autres éléments inlines.
- **Élément auto-fermant** : Élément qui est une balise ouvrante et fermante à la fois. Elle n'a donc pas de contenu. Ce sont soit des balises de type bloc, soit de type inline.

Syntaxe : `<balise>` ou `<balise/>`

Plan du cours



- Introduction
- Syntaxe
- **Élément HTML5**
- Les formulaires
- Les frames

Élément HTML5 - Titres (éléments bloc)



Il existe 6 niveaux de titre. Le niveau le plus élevé est le 1, le plus petit est le 6 :

- `<H1> </H1>`
- `<H2> </H2>`
- `<H3> </H3>`
- `<H4> </H4>`
- `<H5> </H5>`
- `<H6> </H6>`

Titre Niveau 1

Titre Niveau 2

Titre Niveau 3

Titre Niveau 4

Titre Niveau 5

Titre Niveau 6

Élément HTML5 - Paragraphes (éléments bloc)

- L'élément bloc `<p> </p>` permet de construire des paragraphes.
- L'élément inline `
` permet de contrôler les sauts de lignes. elle est dite une balise **Orpheline**.

Ex:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Paragraphe</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8"/>
  </head>
  <body>
    <p> Ceci est une première ligne <br/> Ma deuxième ligne </p>
  </body>
</html>
```

Élément HTML5 - Mise en forme (éléments inline)

- Les éléments comme `` et `<i>` permettent le formatage de sortie du texte, comme en **gras** ou en *italique*.
 - `` : Bold - Texte en gras
 - `` : Texte important
 - `<i>` : Texte en italique
 - `<small>` : Texte en petite taille
 - `<mark>` : Texte marqué
 - `<code>` : Portion au code informatique

Élément HTML5- Images (éléments inline)

- La balise `` permet d'insérer une image dans un document.
- Les attributs suivants sont obligatoires:
 - `src` : L'URL où se situe l'image.
 - `alt` : Courte description de l'image.

```

```

Élément HTML5- Liens (éléments inline)

- L'élément ` ... ` permet d'insérer un lien dans un document.
- Le contenu de l'élément est celui qui sera affiché en tant que lien.
- L'attribut `href` contient l'URL vers laquelle le lien pointe.
 - URL : <http://www.google.com>
 - URL (mail) : <mailto:t.aitbaha@uiz.ac.ma>
 - Fichier local avec chemin relatif : `./dossier/autre_page.html`
 - Fichier local avec chemin absolu : `/www/dossier/autre_page.html`

```
<a href="lien_vers_une_autre_page.html"> Texte affiché </a>
```

Élément HTML5- Listes (éléments bloc) (1/7)



HTML définit 2 types de listes :

- Les listes **numérotées** ou **ordonnées** (ou Ordered (numbered) lists).
- Les listes **non numérotées** dites listes à puces dont les entrées sont signalées par un signe typographique.

Chaque élément d'une liste est déclaré par ` . . . `.

Il existe aussi des listes de définitions (`<dl> . . . </dl>`), les éléments sont déclarés par les balises `<dt>` pour le terme et `<dd>` pour la définition associée.

Élément HTML5- Listes (éléments bloc) (2/7)

Les listes numérotées ou ordonnées:

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

1. item
2. item
3. item

Élément HTML5- Listes (éléments bloc) (3/7)

Les listes numérotées ou ordonnées:

L'attribut type: On peut définir la manière dont on veut lister nos élément en ajoutant un attribut **type** à la balise ``.

Il existe 5 types :

- **"1"** : La liste sera numérotée par des nombres.
- **"A"** : La liste sera numérotée par des lettres en majuscule.
- **"a"** : La liste sera numérotée par des lettres en minuscule.
- **"I"** : La liste sera numérotée par des nombres romain majuscule.
- **"i"** : La liste sera numérotée par des nombres romain minuscule.

```
<ol type="1">  
  <li> item </li>  
  <li> item </li>  
  <li> item </li>  
</ol>
```

Élément HTML5- Listes (éléments bloc) (4/7)

Les listes non numérotées:

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

- item
- item
- item

Élément HTML5- Listes (éléments bloc) (5/7)

Les listes non numérotées:

L'attribut style: On peut définir la manière dont on veut lister nos éléments en ajoutant un attribut **style** dans lequel on indique le type du marqueur des éléments en utilisant **list-style-type**.

Il existe 4 types de marqueurs :

- “**square**” : Pour utiliser le marqueur carreau.
- “**circle**” : Pour utiliser le marqueur cercle.
- “**disc**” : Pour utiliser le marqueur disque (Celui utiliser par défaut).
- “**none**” : Pour n'utiliser aucun marqueur .

```
<ul style="list-style-type: disc;">
  <li> item </li>
  <li> item </li>
  <li> item </li>
</ul>
```

Élément HTML5- Listes (éléments bloc) (6/7)

Les listes de définition:

```
<dl>
  <dt> item </dt> <dd> definition </dd>
  <dt> item </dt> <dd> definition </dd>
  <dt> item </dt> <dd> definition </dd>
</dl>
```

```
item
  definition
item
  definition
item
  definition
```

Élément HTML5- Exercice

Cette page est encore vide. Ecrivez vos premières lignes de HTML en présentant vos cours préférés !

- Ajoutez un titre de niveau 1.
- A la suite du titre, ajoutez un paragraphe indiquant "Voici mes cours préférés:".
- Ajoutez une liste ordonnée d'au moins trois de vos cours préférés.
- Mettez en valeur (emphase forte) le premier cours de la liste

```
<!DOCTYPE html>
<html>
  <!-- en-tête du document -->
  <head>
    <title>Titre</title>
    <meta charset="utf-8">
  </head>
  <!-- corps du document -->
  <body>
    <!-- Contenu de la page -->
  </body>
</html>
```

Élément HTML5- Exercice

```
<!DOCTYPE html>
<html>
  <!-- en-tête du document -->
  <head>
    <title>Titre</title>
    <meta charset="utf-8">
  </head>
  <!-- corps du document -->
  <body>
    <!-- Contenu de la page →
    <H1>Exercice</H1>
    <p>Voici mes cours préférés:</p>
    <ol>
      <li><strong>Technologies du Web</strong></li>
      <li>Programmation C</li>
      <li>Structure des données</li>
    </ol>
  </body>
</html>
```

Élément HTML5- Tableaux (éléments bloc) (1/6)

```
<table>
  <caption>
    <!--titre du tableau-->
  </caption>
  <thead>
    <!--entête de tableau-->
  </thead>
  <tfoot>
    <!--pied de tableau-->
  </tfoot>
  <tbody>
    <!--corps de la table-->
    <tr>
      <td>cellule 1</td>
      <td>cellule 2</td>
    </tr>
  </tbody>
</table>
```

- Les éléments `<thead>` et `<tfoot>` permettent de répéter l'élément dans les tableaux sur plusieurs pages (impression).
- L'élément `<caption>` indique le titre du tableau
- La balise `<tr>` déclare une ligne.
- Les balises `<td>` (cellule normale) ou `<th>` (cellule titre/grasse) déclarent les cellules dans la ligne.
- Les attributs `rowspan` et `colspan` fusionnent les cellules.

Élément HTML5- Tableaux (éléments bloc) (2/6)

Passagers du vol 377

En-tête du tableau	Nom	Age	Pays	<thead>
Corps du tableau	Carmen	33 ans	Espagne	<tbody>
	Michelle	26 ans	Etats-Unis	
	François	43 ans	France	
	Martine	34 ans	France	
	Jonathan	13 ans	Australie	
Pied du tableau	Xu	19 ans	Chine	<tfoot>
	Nom	Age	Pays	

Les éléments **<thead>** et **<tfoot>** permettent de répéter l'élément dans les tableaux sur plusieurs pages.

Élément HTML5- Tableaux (éléments bloc) (3/6)

- La balise `<tr>` déclare une ligne.
- Les balises `<td>` (cellule normale) ou `<th>` (cellule titre/grasse) déclarent les cellules dans la ligne.

The diagram shows a table with three columns: Nom, Age, and Pays. The first row contains 'Anne', '27 ans', and 'France'. The second row contains 'Carmen', '33 ans', and 'Espagne'. The third row contains 'Michelle', '26 ans', and 'Etats-Unis'. The fourth row contains 'Ogasaku Nyagatosoka', '18 ans', and 'Japon'. A red box highlights the second row, labeled 'Tr (ligne)'. Green boxes highlight the three cells of the second row, labeled 'Td (cellules)'.

Nom	Age	Pays
Anne	27 ans	France
Carmen	33 ans	Espagne
Michelle	26 ans	Etats-Unis
Ogasaku Nyagatosoka	18 ans	Japon

Élément HTML5- Tableaux (éléments bloc) (4/6)



Pour effectuer une fusion, on rajoute un attribut à la balise `<td>`. Il faut savoir qu'il existe deux types de fusion :

- **La fusion de colonnes** : La fusion s'effectue horizontalement. On utilisera l'attribut `colspan`.
- **La fusion de lignes** : Deux lignes seront groupées entre elles. La fusion s'effectuera verticalement. On utilisera l'attribut `rowspan`.

Élément HTML5- Tableaux (éléments bloc) (5/6)

Écrire le code qui affichera le tableau ci-dessous sur une page web

```
<table>
  <caption>
    <!--titre du tableau-->
  </caption>
  <thead>
    <!--entête de tableau-->
  </thead>
  <tfoot>
    <!--pied de tableau-->
  </tfoot>
  <tbody>
    <!--corps de la table-->
    <tr>
      <td>cellule 1</td>
      <td>cellule 2</td>
    </tr>
  </tbody>
</table>
```

Titre du tableau

Titre 1	Titre 2	Titre 3
cellule 1	cellule 2	cellule 3
cellule 4	cellule 5	cellule 6
cellule 7		
Titre 1	Titre 2	Titre 3

Élément HTML5- Tableaux (éléments bloc) (6/6) Exercice

```
<table border="1">
  <caption>
    <h2>Titre du tableau</h2>
  </caption>
  <thead>
    <tr>
      <th>Titre 1</th>
      <th>Titre 2</th>
      <th>Titre 3</th>
    </tr>
  </thead>
  <tfoot>
    <tr>
      <th>Titre 1</th>
      <th>Titre 2</th>
      <th>Titre 3</th>
    </tr>
  </tfoot>
  <tbody>
    <tr>
      <td>cellule 1</td>
      <td>cellule 2</td>
      <td>cellule 3</td>
    </tr>
    <tr>
      <td>cellule 4</td>
      <td>cellule 5</td>
      <td rowspan="2">cellule 6</td>
    </tr>
    <tr>
      <td colspan="2">cellule 7</td>
    </tr>
  </tbody>
</table>
```

Plan du cours



- Introduction
- Syntaxe
- Élément HTML5
- **Les formulaires**
- Les frames

Les formulaires



- Permettent **d'interagir avec les utilisateurs** en leur proposant d'**entrer des informations**.
- Seule l'interface de formulaire qui sera faite en **HTML**.
- Les informations que les utilisateurs saisissent, seront traiter par des autres **scripts**.

Les formulaires - Déclaration d'un formulaire

- L'élément `<form>` ... `</form>` permet de déclarer un formulaire.
- Les attributs:
 - **action** : URL du script auquel sera soumis le formulaire.
 - **method** : Spécifie la méthode d'acheminement des données (**GET** par défaut ou **POST**)

GET

- Seule une quantité limitée de données peut être envoyée car les données sont envoyées dans l'en-tête.
- La requête GET n'est pas sécurisé car les données sont exposées dans la barre d'URL.

Post

- Une grande quantité de données peut être envoyée car les données sont envoyées dans le corps.
- La requête POST est sécurisée car les données ne sont pas exposées dans la barre d'URL.

Les formulaires - Déclaration d'un formulaire (suite)

- Les deux méthodes **GET** et **POST** sont utilisées pour transférer des données du client au serveur avec le protocole **HTTP**.
- La différence clé entre les méthodes **POST** et **GET** est que **GET** transporte le paramètre dans la chaîne d'URL, tandis que **POST** transporte le paramètre dans le corps du message, ce qui le rend plus sûr le **transfère des données du client au serveur avec le protocole HTTP**.

```
<form action="Nom_de_votre_page.html" method="POST">  
  
</form>
```


Les formulaires - Ensemble de champs



En HTML, il est interdit de mettre des champs de formulaire directement à l'intérieur d'un `<form>`. Il faut d'abord les regrouper dans :

- L'élément `<fieldset>` ... `</fieldset>` permet de définir un regroupement, des champs de formulaire de sémantique proche, dans un formulaire.

L'élément `<legend>` ... `</legend>` permet de donner une légende à un `fieldset`.

- Des divisions `<div>` ... `</div>`.

Les formulaires - Étiquettes

La plupart des champs sont naturellement accompagnés d'une étiquette

- L'élément `<label> ... </label>` permet de définir une étiquette.
- On peut la placer où on veut, en général juste à gauche ou à droite du champ.
- Son attribut `for` référence l'attribut `id` du champ correspondant.
- Dans les navigateurs graphiques, un clic sur l'étiquette d'un champ permet en général de sélectionner le champ.

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

Les formulaires - Champs de saisie

- La balise `<input/>` a une utilisation très vaste dans les formulaires. Elle représente un champ de saisie.
- L'élément `<input/>` contient les attributs suivants:
 - **type** : détermine le type (*texte, mot de passe, liste, etc.*) du champ.
 - **name** : nom du paramètre de la requête HTTP, est **obligatoire** (sauf pour les types **"reset"** et **"submit"**), il permet de préciser au serveur à quelle saisie on fait référence.
 - **value** : permet de préciser la valeur par défaut (*facultatif*).

```
<form>
  <b> Nom </b>
  <input type="text" name="nom" />
</form>
```

Nom

Les formulaires - Saisie d'une ligne de texte

- Le type **text** est utilisé pour la saisie d'un texte dont la taille est inférieure à une ligne.
- La taille maximale de la chaîne de caractères à saisir peut être spécifiée à l'aide de l'attribut **maxlength** (facultatif).

```
<form>
  <b> Localité </b>
  <input type="text" name="localite" maxlength=80 value="Agadir" />
</form>
```

Les formulaires - Saisie d'un mot de passe

- Le type **password** est utilisé pour la saisie d'un texte dont les caractères sont remplacés par des astérisques : c'est généralement utilisé pour la saisie des mots de passe. Le mot de passe est quand même transmis en clair au serveur.
- La taille maximale de la chaîne de caractères à saisir peut être spécifiée à l'aide de l'attribut **maxlength** (facultatif).

```
<form>  
  <b> Mot de passe : </b>  
  <input type="password" name="pwd" value="123456" />  
</form>
```

Mot de passe :

Les formulaires - Choix multiples dans une liste

- Le type **checkbox** permet de choisir plusieurs éléments parmi une liste de possibilités.
- Cela se matérialise sous forme de cases à cocher.
- La valeur retournée est **obligatoirement** précisée à l'aide de l'attribut **value**.
- L'attribut **checked="checked"** permet de cocher la case par défaut.

```
<form>
  <b> Cours : </b>
  <input type="checkbox" name="chk1" value="Structure de données"/> Structure de données
  <input type="checkbox" name="chk1" value="Programmation C" /> Programmation C
  <input type="checkbox" name="chk1" value="Technologies du Web" checked="checked" />
Technologies du Web
</form>
```

Cours : ☐ Structure de données ☐ Programmation C ☒ Technologies du Web

Les formulaires - Choix unique dans une liste

- Le type **radio** permet de choisir un seul élément parmi une liste de possibilités.
- Cela se matérialise sous forme de boutons radio.
- La valeur retournée est **obligatoirement** précisée à l'aide de l'attribut **value**.
- L'attribut **checked="checked"** permet de cocher la case par défaut.

```
<form>
  <b> Civilité : </b>
  <input type="radio" name="chk1" value="Mademoiselle"/> Mademoiselle
  <input type="radio" name="chk1" value="Madame" /> Madame
  <input type="radio" name="chk1" value="Monsieur" checked="checked" /> Monsieur
</form>
```

Civilité : ☐ Mademoiselle ☐ Madame ☒ Monsieur

Les formulaires - Boutons (1/3) - Réinitialisation de formulaire

- Le type **reset** permet de réinitialiser le formulaire en affectant aux différents champs leur valeur par défaut.
- L'attribut **value** permet de changer le texte du bouton correspondant.


```
<input type="reset" value="Reset" />
```



Les formulaires - Boutons (2/3) - Soumission de formulaire

- Le type `submit` permet de soumettre le formulaire.
- Le client envoie le contenu du formulaire à l'adresse précisée par l'attribut `action` de la balise `<form>`.
- L'attribut `value` permet de changer le texte du bouton correspondant.

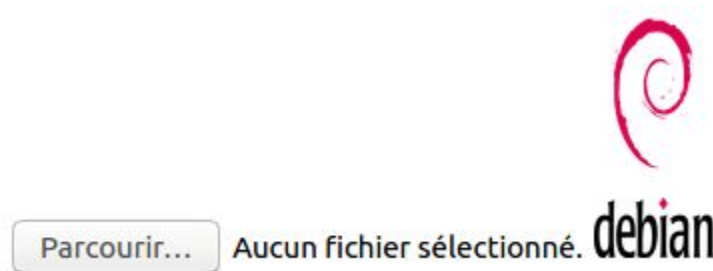
```
<input type="submit" value="Soumettre le formulaire"/>
```



Les formulaires - Boutons (3/3)

- Un élément `<input/>` de type :
 - `image` : Bouton avec image pour l'envoi des données du formulaire.
 - `file` : Bouton d'ouverture de boîte de recherche de fichiers.

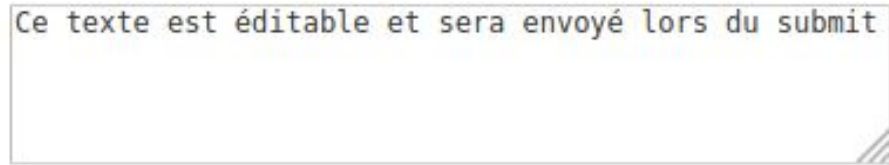
```
<input type="file" name="uFichier" id="fichier"/>  
<input type="image" src="debian.png" name="action" value="Delete"/>
```



Les formulaires - Champs de texte

- Pour les saisies multiligne, on utilise la balise `<textarea> ... </textarea>`.
- Le texte délimité par cette balise permet d'initialiser la valeur par défaut du champ.
- La balise fermante est **obligatoire** même si le champ est vide.
- Les attributs `rows` et `cols` (**obligatoires**) permettent de spécifier la taille en lignes et colonnes de la fenêtre de saisie.

```
<textarea rows="4" cols="50">Ce texte est éditable et sera envoyé lors du  
submit</textarea>
```

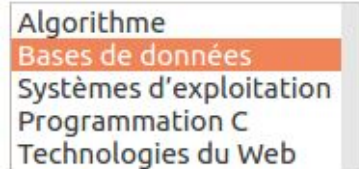


Ce texte est éditable et sera envoyé lors du submit

Les formulaires - Menu de sélection

- La balise `<select>` permet d'ajouter une liste de sélection.
- L'attribut facultatif `size` permet de préciser le nombre de choix apparaissant sur la page Web. Par défaut, ce nombre est initialisé à 1.
- L'attribut `multiple="multiple"` permet d'autoriser des réponses multiples.
- Les choix du menu sont indiqués à l'aide de la balise `<option>`:
 - L'attribut `selected="selected"` permet de spécifier le(s) choix par défaut.
 - L'attribut `value` permet de spécifier la valeur associée au choix.

```
<select name="laliste" size="5" multiple="multiple">
  <option value="1">Algorithme</option>
  <option selected="selected" value="2">Bases de données</option>
  <option value="3">Systèmes d'exploitation</option>
  <option value="4">Programmation C</option>
  <option value="5">Technologies du Web</option>
</select>
```



Les formulaires - Exemple

```
<form action="mailto:t.aitbaha@uiz.ac.ma" method="POST">
  <fieldset>
    <legend>Exemple de formulaire</legend>
    <label>Nom :</label>
    <input type="text" name="monNom" id="nom"/>
    <label>Prénom :</label>
    <input type="text" name="monPrenom" id="prenom"/>
    <br/>
    <input type="checkbox" name="maNewsletter" id="newsletter" />
    <label for="newsletter">Une checkbox</label>
    <input type="radio" name="monSexe" id="homme" />
    <label for="homme">Homme</label>
    <input type="radio" name="monSexe" id="femme" />
    <label for="femme">Femme</label>
    <br />
    <label for="photo">Fichier:</label>
    <input type="file" name="maPhoto" id="photo" />
    <br />
```

```
<select name="laliste" size="3" multiple="multiple">
  <option value="1">toto</option>
  <option selected="selected" value="2">titi</option>
  <optgroup label="les autres">
    <option value="3">tata</option>
    <option value="4">tutu</option>
    <option value="5">tete</option>
  </optgroup>
</select>
<br />
<textarea name="texte" rows="10" cols="80">Raconte-moi une histoire
</textarea>
<br />
<input type="submit" name="maSoumission" id="soumission" />
<input type="submit" name="action" value="Insert" />
<input type="submit" name="action" value="Update" />
<input type="image" src="Images/logoasi.png" alt="logoasi.png"
name="action" width="75" />
<input type="reset" value="Reset" />
</fieldset>
</form>
```

Les formulaires - Exercice

Ecrire le code source réalisant le formulaire suivant:

Fiche de renseignements

Nom : Prénom : Mot de passe :

Quel est votre niveau en HTML : ☒ Excellent ☐ Bien ☐ Moyen

Quelle est votre discipline préférée ?

Les formulaires - Petit résumé des champs

`input type = text`

Votre nom

`input type = checkbox`

☐ Le Jazz ☒ La techno

`input type = radio`

☐ une femme ☒ un homme

`input type= submit`

`select + option`

`textarea`

Que souhaitez-vous me dire ?

Plan du cours



- Introduction
- Syntaxe
- Élément HTML5
- Les formulaires
- **Les frames**

Les frames



- Les **frames** permettent d'**afficher plusieurs pages en même temps** (*chaque page est une frame*).
- Cela est très utile pour **faire un menu fixe par exemple à gauche de la fenêtre** et **un contenu à droite qui change**.
- Tous les arrangements de frames sont permis et le nombre n'est pas limité non plus.
- Chaque frame correspond à une page HTML.

Les frames - Comment créer les frames ?



Le concept de base est assez simple :

- Utilisez l'élément **Frameset** à la place de l'élément **body** dans un document HTML.
- Utilisez l'élément **Frame** pour créer des cadres pour le contenu de la page Web.
- Utilisez l'attribut **src** pour identifier la ressource qui doit être chargée à l'intérieur de chaque frame.
- Créez un fichier différent pour chaque frame.

Les frames

- Pour créer un ensemble de trois colonnes verticales, nous devons utiliser l'élément **Frameset** avec l'attribut **cols**.
- L'attribut **cols** est utilisé pour définir le nombre et la taille des colonnes que le **Frameset** contiendra.

Création de colonnes verticales

- Dans notre cas, nous avons trois fichiers à afficher, nous avons donc besoin de trois frames. Pour créer trois frames, nous devons attribuer trois valeurs séparées par des virgules à l'attribut **cols**.
- Pour simplifier les choses, nous allons attribuer la valeur ***** à chacun des cadres, cela entraînera leur redimensionnement automatique pour remplir l'espace disponible.

```
<!DOCTYPE html>
<html>
  <body>
    <h1>Frame 1</h1>
    <p>Contents of Frame 1</p>
  </body>
</html>
```

Frame1.html

```
<!DOCTYPE html>
<html>
  <body>
    <h1>Frame 2</h1>
    <p>Contents of Frame 2</p>
  </body>
</html>
```

Frame2.html

```
<!DOCTYPE html>
<html>
  <body>
    <h1>Frame 3</h1>
    <p>Contents of Frame 3</p>
  </body>
</html>
```

Frame3.html

Les frames



```
<!DOCTYPE html>
<html>
  <frameset cols="*,*,*">
    <frame src="Frame1.html">
    <frame src="Frame2.html">
    <frame src="Frame3.html">
  </frameset>
</html>
```

Frame 1

Contents of Frame 1

Frame 2

Contents of Frame 2

Frame 3

Contents of Frame 3

Les frames

Création de lignes horizontales

- Des rangées de cadres peuvent être créées en utilisant l'attribut `rows` plutôt que l'attribut `cols` comme indiqué dans le code HTML ci-dessous.
- En faisant ce seul changement, les cadres se chargent désormais en quatre rangées empilées les unes sur les autres.

```
<!DOCTYPE html>
<html>
  <frameset rows="*,*,*">
    <frame src="Frame1.html">
    <frame src="Frame2.html">
    <frame src="Frame3.html">
  </frameset>
</html>
```

Frame 1

Contents of Frame 1

Frame 2

Contents of Frame 2

Frame 3

Contents of Frame 3

Les frames

Mélange de colonnes et de lignes

- Les colonnes et les lignes de cadres "**Frames**" peuvent apparaître sur la même page Web en imbriquant un jeu de cadres "**Frameset**" à l'intérieur d'un autre.
- Pour ce faire, nous créons d'abord un **Frameset**, puis imbriquons un **Frameset** enfant dans l'élément parent.
- Voici un exemple de la façon dont nous pourrions imbriquer deux lignes dans un ensemble de trois colonnes.

```
<!DOCTYPE html>
<html>
  <frameset cols="*,*,*">
    <frameset rows="*,*">
      <frame src="Frame1.html">
      <frame src="Frame2.html">
    </frameset>
    <frame src="Frame2.html">
    <frame src="Frame3.html">
  </frameset>
</html>
```

Les frames

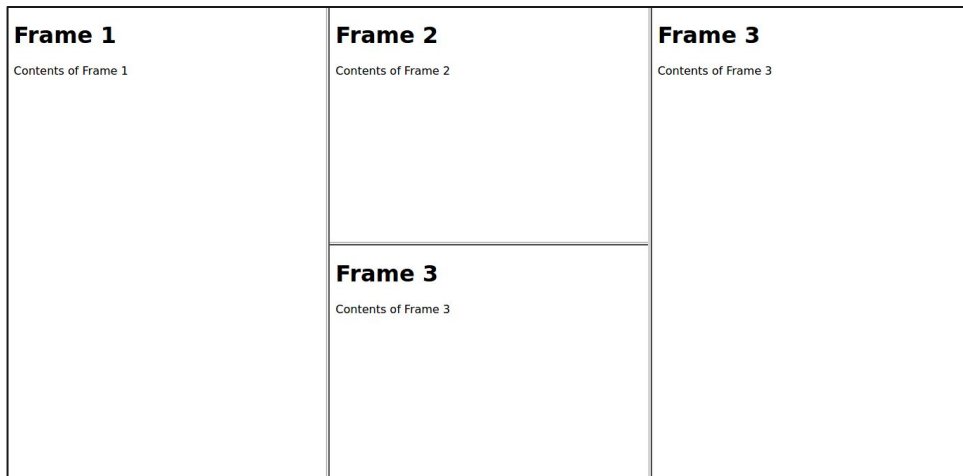


Frame 1 Contents of Frame 1	Frame 3 Contents of Frame 3	Frame 3 Contents of Frame 3
Frame 2 Contents of Frame 2		

Les frames

- L'élément imbriqué peut être placé dans n'importe quelle position. Par exemple, si nous voulions que l'élément imbriqué apparaisse en position centrale, nous réorganiserions simplement les éléments comme ceci.

```
<!DOCTYPE html>
<html>
  <frameset cols="*,*,*">
    <frame src="Frame1.html">
    <frameset rows="*,*">
      <frame src="Frame1.html">
      <frame src="Frame2.html">
    </frameset>
    <frame src="Frame3.html">
  </frameset>
</html>
```

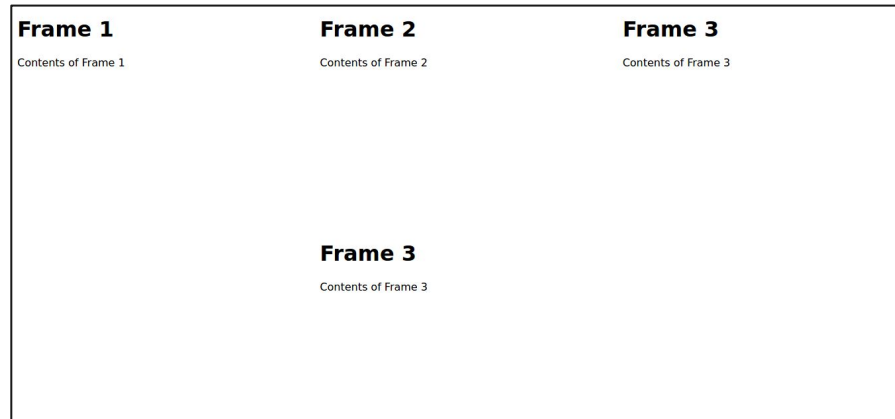


Les frames - Dimensionnement des cadres

- Les cadres peuvent être dimensionnés en **pixels** “**px**” ou en **pourcentages** “**%**”, ou ils peuvent être définis pour **s’ajuster automatiquement** en fonction de l’espace disponible.
- Pour spécifier la taille d'un cadre, insérez la valeur souhaitée dans l'attribut **cols** ou **rows**.
- Par défaut, sauf si l'attribut **noresize** est ajouté à un cadre, les visiteurs du site Web peuvent utiliser leur souris pour faire glisser la bordure entre deux cadres pour redimensionner les cadres.
- Si cela n'est pas souhaitable, l'attribut **noresize** peut être appliqué à un élément frame pour empêcher le redimensionnement.

Les frames - Marges et bordures du cadre

- Nous pouvons *augmenter* ou *diminuer* la marge entre les cadres en utilisant les attributs `marginheight` ou `marginwidth`, et également *supprimer la bordure* entre les cadres si nous le souhaitons à l'aide de l'attribut `frameborder`.
- Prenons l'exemple précédent, nous pouvons supprimer les bordures entre les trois colonnes et ajouter également une marge autour du contenu.



Les frames - Marges et bordures du cadre

```
<!DOCTYPE html>
<html>
  <frameset cols="*,*,*">
    <frame src="Frame1.html" marginheight="15">
    <frameset rows="*,*">
      <frame src="Frame1.html" frameborder="0">
      <frame src="Frame2.html" frameborder="0">
    </frameset>
    <frame src="Frame3.html" frameborder="0">
  </frameset>
</html>
```

Frame 1

Contents of Frame 1

L'attribut **marginheight** appliqué à la première image ajoutera **15pixels** de marge au-dessus et en dessous du contenu chargé dans le premier cadre.

La valeur **frameborder** de **0** supprime les bordures autour des quatre cadres.

Frame 1

Contents of Frame 1

Frame 3

Contents of Frame 3

Frame 2

Contents of Frame 2

Les frames - Cibler des cadres avec des liens



- L'une des utilisations les plus courantes des cadres consiste à ***créer une navigation dans un cadre*** qui est toujours visible quelle que soit la position du contenu des autres cadres.
- Lorsqu'ils sont correctement mis en œuvre, ***les liens de navigation entraînent le chargement de nouvelles ressources dans un cadre*** tandis que ***les autres cadres restent statiques***.

Les frames - Cibler des cadres avec des liens

Comment le Faire ?

- La première étape pour y arriver est d'*attribuer un nom au cadre où nous voulons que les liens s'ouvrent*.
- Nous voudrions probablement utiliser *la colonne de gauche pour notre navigation* et *la colonne centrale comme cadre cible*.
- Pour ce faire, nous devons *attribuer un nom à notre cible*.

```
<!DOCTYPE html>
<html>
  <frameset cols="*,*,*">
    <frame src="Frame1.html" marginheight="15">
    <frame src="Frame2.html" name="mid_col" frameborder="0">
    <frame src="Frame3.html" frameborder="0">
  </frameset>
</html>
```

Les frames - Cibler des cadres avec des liens

Comment le Faire ?

- Maintenant que nous avons nommé la colonne centrale `name="mid_col"`, nous pouvons créer quelques liens dans notre document source de colonne de gauche *Frame1.html* et *cibler la colonne centrale*.

```
<!DOCTYPE html>
<html>
  <body>
    <h1>Frame 1</h1>
    <p>Contenu de la Frame 1</p>
    <ul>
      <li><a href="Frame1.html" target="mid_col">Frame 1</a></li>
      <li><a href="Frame2.html" target="mid_col">Frame 2</a></li>
      <li><a href="Frame3.html" target="mid_col">Frame 3</a></li>
    </ul>
  </body>
</html>
```

Les frames - Exercice



**Ecrire le code source qui permet de diviser
votre page Web en 4 Frames**

Frame 1	Frame 2	Frame 3
Frame 4		



Modifiez le code source de façon de cibler la Frame 3 à travers l'index en
Frame 1

<ul style="list-style-type: none">- Bleu- Jaune- Rouge- Vert	Frame 2	
Frame 4		

Comment le Faire ?

- **La première étape** pour y arriver est d'*attribuer un nom au cadre où nous voulons que les liens s'ouvrent*.
- Nous voudrions probablement utiliser *la colonne de gauche pour notre navigation* et *la colonne centrale comme cadre cible*.
- Pour ce faire, nous devons **attribuer un nom à notre cible**.
- Maintenant que nous avons nommé la colonne centrale `name="mid_col"`, nous pouvons créer quelques liens dans notre document source de colonne de gauche **Frame1.html** et *cibler la colonne centrale*.