



République Tunisienne
Ministère de l'Enseignement Supérieur,
de la Recherche Scientifique



Technologies et programmation web

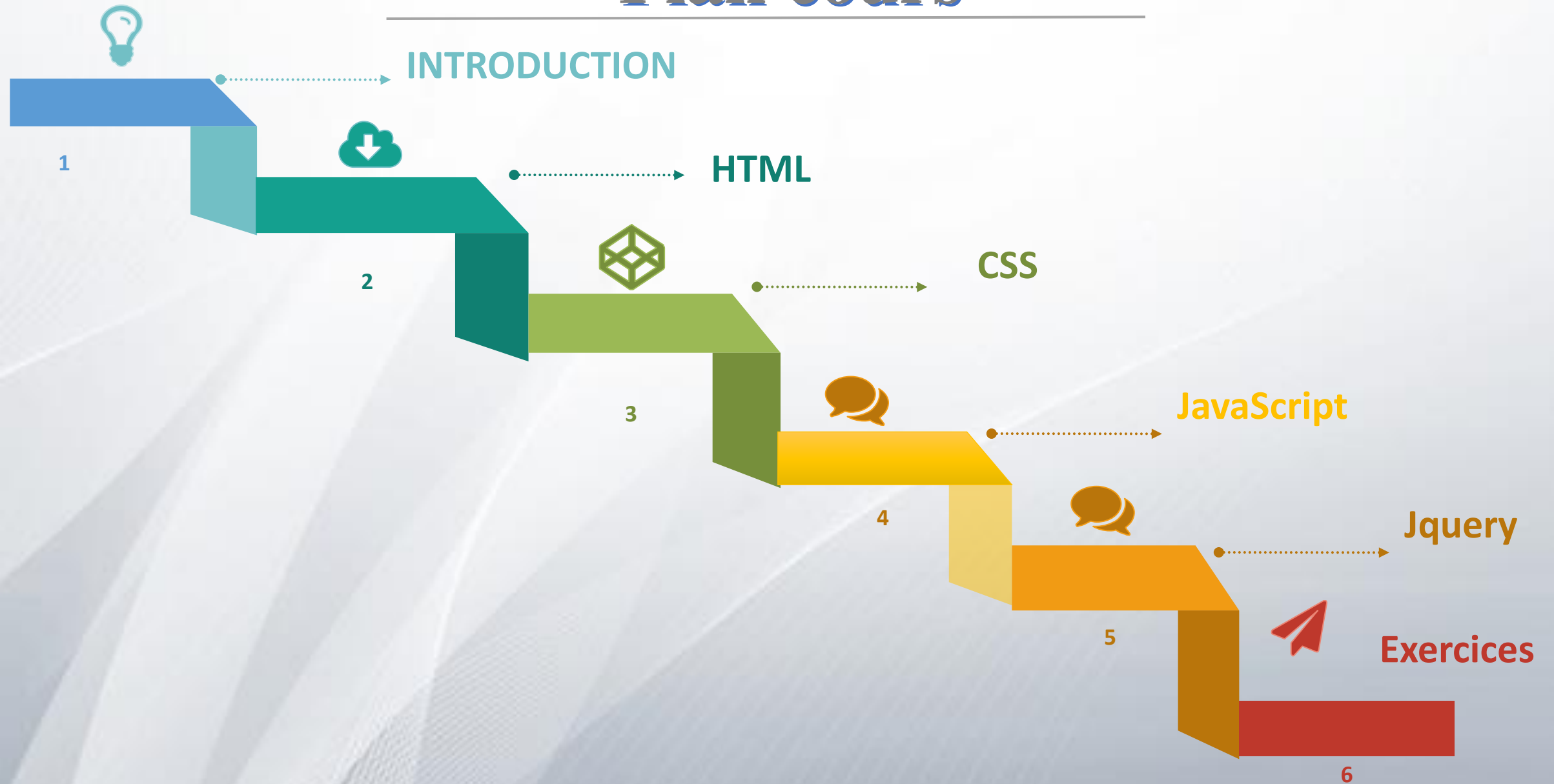
HTML

Njeh Maissa

Audiences: DLSI-ADBD

Année universitaire : 2023/2024

Plan cours



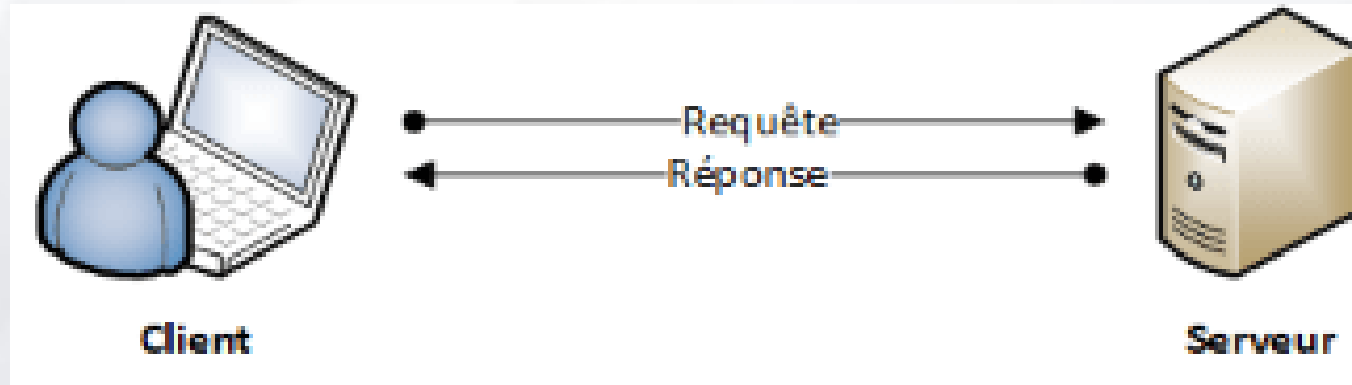
Architecture Client/Serveur

4 Architecture Client/Serveur

Le web est un des services disponibles sur Internet.

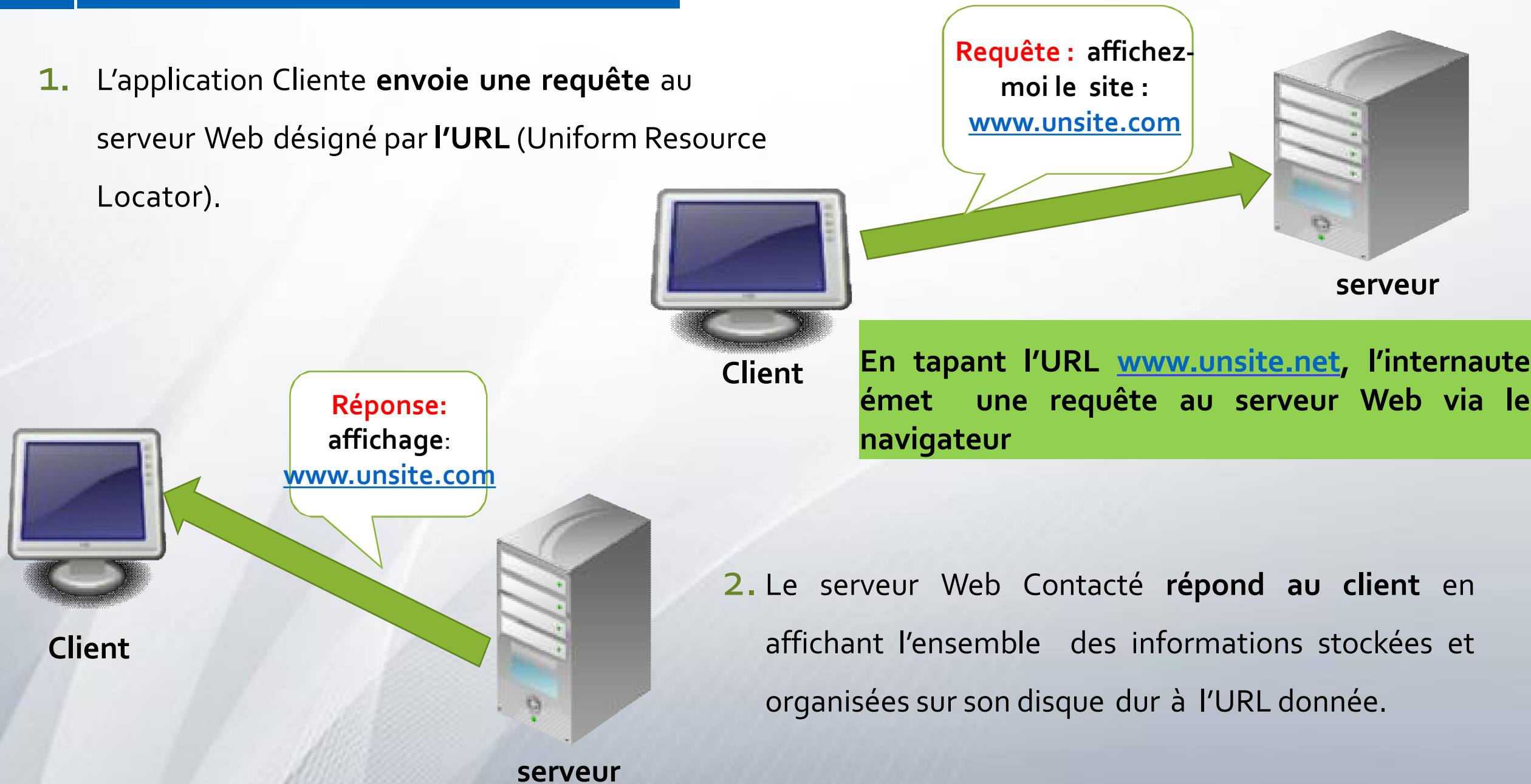
- ✓ Internet est un réseau de services.
- ✓ Internet est basée sur l'architecture client/serveur

➔ Les applications développées pour les sites Internet reposent sur le modèle Client/Serveur.



- Les services Internet sont tous fournis par des serveurs.
- Les demandeurs du service sont nommés les clients.
- Les clients émettent des requêtes en se basant sur un protocole.
- Après traitement des requêtes, les serveurs renvoient des réponses en se basant sur un protocole.

1. L'application Client **envoie une requête** au serveur Web désigné par l'**URL** (Uniform Resource Locator).



2. Le serveur Web Contacté **répond au client** en affichant l'ensemble des informations stockées et organisées sur son disque dur à l'URL donnée.

- Un **Serveur** est une application située sur un ordinateur très puissant, capable de gérer un grand nombre de requêtes simultanément.
- Un **Client** est une application qui se connecte à un serveur web pour obtenir ou modifier des informations à l'aide de requêtes (par exemple le navigateur web).

- L'accès à un site Internet nécessite les éléments suivants :

- Une application cliente (navigateur Web : Mozilla Firefox, Google Chrome, Internet Explorer) auquel nous fournissons l'adresse du site recherché du type <http://www.unsite.net>.
- Une application serveur (serveur Web : ApacheHTTP Server, IIS-Internet Information Server, Tomcat),
- Un protocole HTTP.

Chaque service est associé à un protocole (web = HTTP).

7 Architecture Client/Serveur

- **HTTP** (Hypertext Transfer Protocol) est un protocole de **communication** entre **un client et un serveur**.



- Le client **demande** une page (ressource) au serveur en **envoyant** une requête et
- le serveur réagit en **envoyant une réponse**, qui est en général **une page Html**.
- Quand on surfe sur Internet chacun de nos clics provoque en général cet échange.

- Le **WWW** (World Wide Web) est un ensemble mondial de documents **hypertextes** et **hypermédi**as placés dans des fichiers au format spécifique appelés **pages HTML**
- Une **page Web** est un fichier de données (fichier contenant du code HTML) situé sur un ordinateur distant
 - Utilisation du protocole Internet **HTTP** pour télécharger la page
 - Utilisation d'un **navigateur** pour l'afficher

HTML



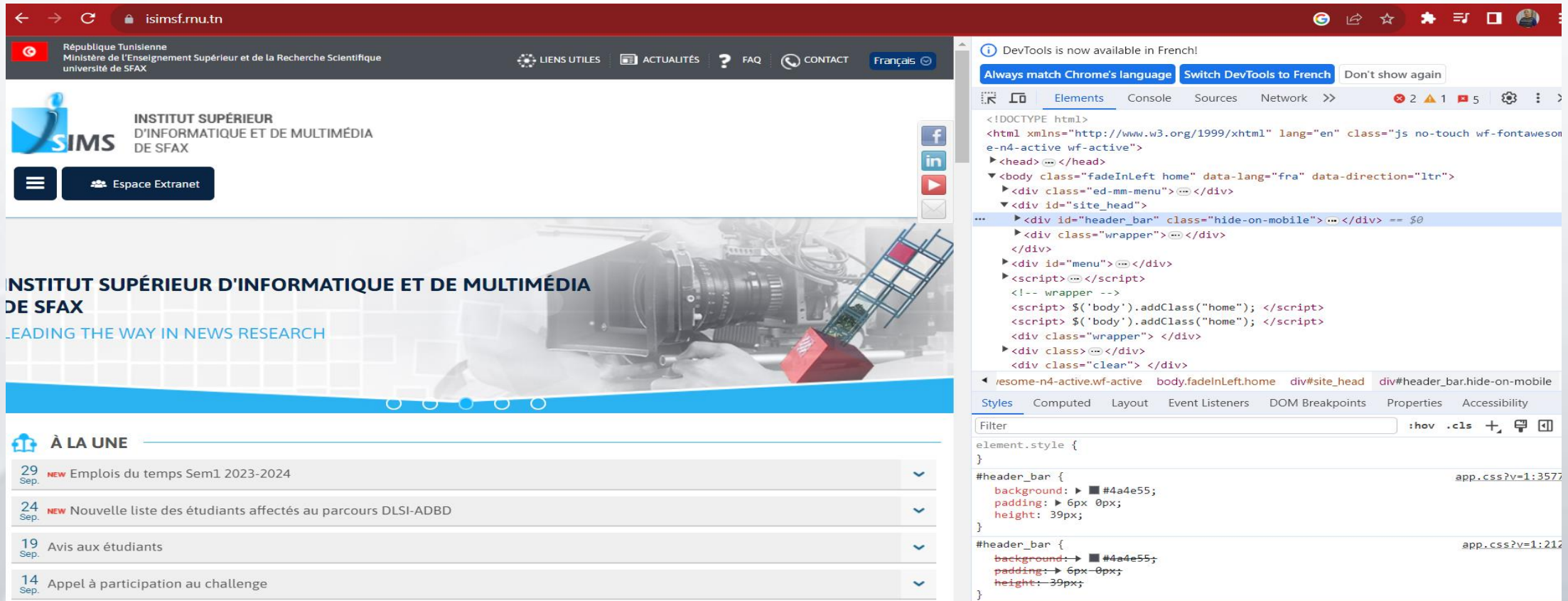
- La meilleure façon de plongée dans le **monde web** et pour Être capable de mener à bien la construction **d'un site ou d'une page web**.

- On commence par la base : Apprendre à écrire en **langage HTML5 & CSS3**





- Tous les sites qui **navignent sur le internet** sont créés avec ces 2 langues.



- Dans ce cours, vous allez être capable de:
 - mériter les bases de **html**
 - faire la mise en forme avec **Css**
 - agencer le contenu de vos **pages** ou créez-vous propre **formulaire**

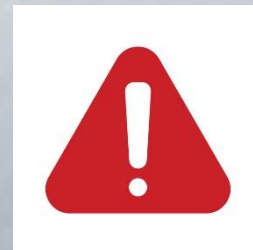
HTML : *HyperText Markup Language*.

C'est un **langage de description de document** conçu pour représenter des **documents hypertextuels**, appelés aujourd'hui « **pages web** ».

HTML n'est ni un protocole ni un langage de programmation

Il ne contient ni structure itérative, ni instruction de boucles, ni variable, ni fonction

Ce langage est pourvu d'un système de **balisage** qui va permettre de structurer notre document

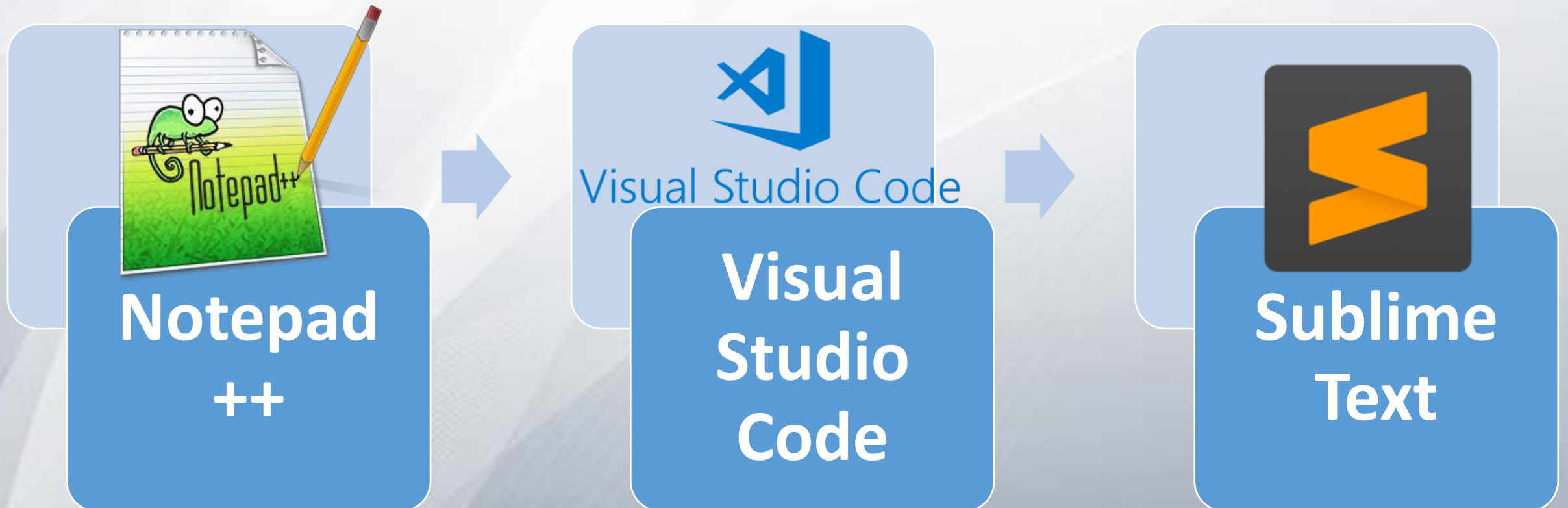


7 HTML : Editeur de texte

Créer des documents HTML avec un **éditeur de texte**

Un éditeur de texte est un logiciel destiné à la création et l'édition de fichiers textes.

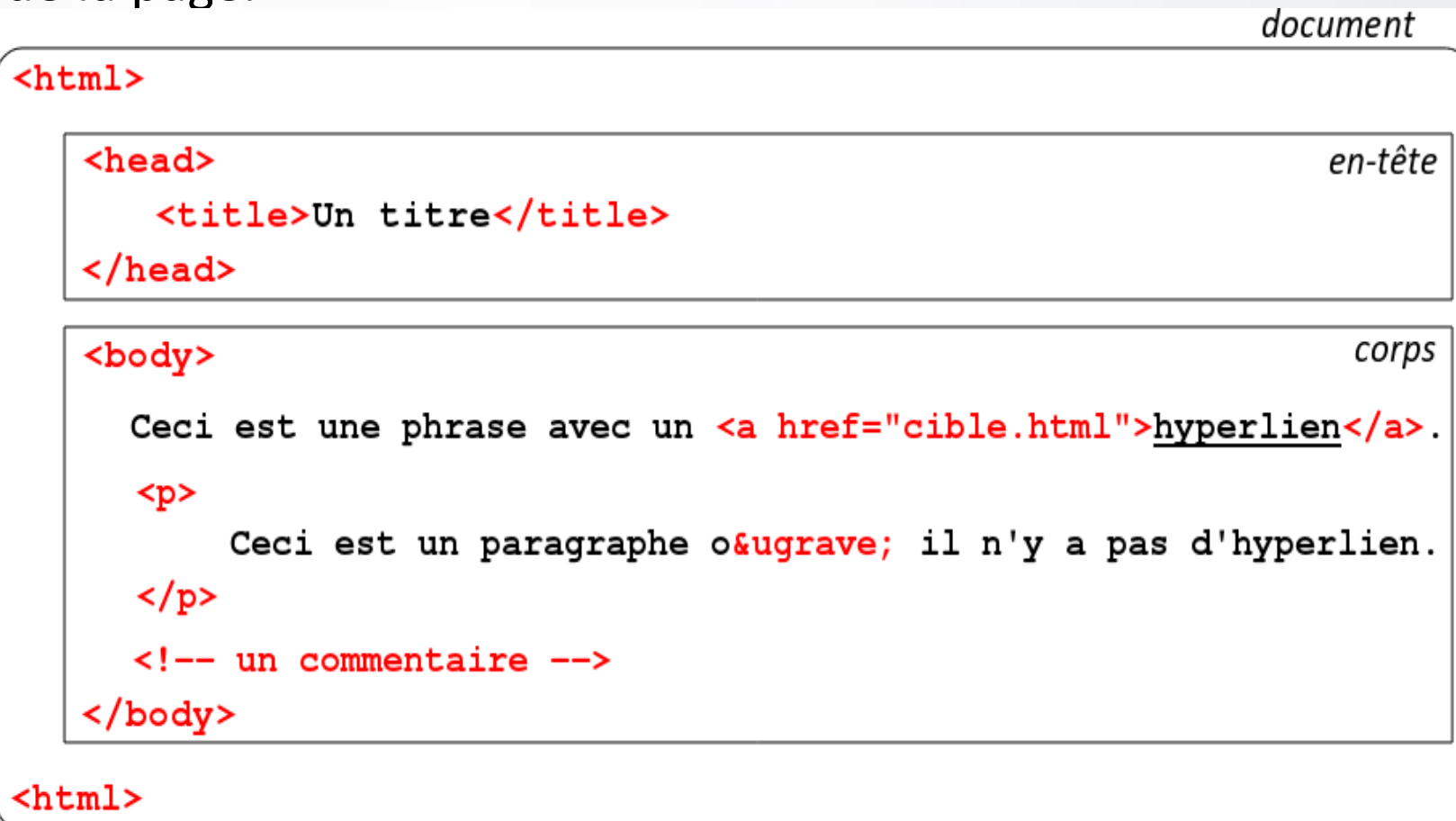
- un fichier de contenu en HTML « qui aura l'extension **.html** »
- un fichier de style en CSS « qui aura l'extension **.css** »



8 HTML : Syntaxe

- Une page html est **un simple fichier texte** commençant par la balise **<html>** et finissant par la balise **</html>**.
- Elle contient également un **en-tête** décrivant le titre de la page, puis un **corps** dans lequel se trouve le contenu de la page.

- L'**en-tête** est délimité avec les balises `<head>` `</head>`
- Le **corps** est délimité avec les balises `<body>` `</body>`



9 HTML : Syntaxe

<html>

<head>

<!-- l'en-tête

</head>

<body>

<!-- le corps

</body>

</html>

Informations qui **ne seront pas affichées** sur la page HTML

titre de la page :

<title>ma page html</title>

Informations utilisées par les moteurs de recherche :

<meta name="description" content="Formation HTML" />

Informations qui vont **s'afficher** dans le navigateur titres, texte, images, ...

- Les documents HTML ne peuvent contenir que les symboles du clavier américains (lettres, chiffre et signes de ponctuation (; , : ! ? » '))
- Les autres caractères sont codé :
 - é est codé par `é`;
 - & est codé par `&`;
 - < est codé par `<`; et > par `>`;
 - ...

Remarque :

- les noms de balise et d'attribut sont insensibles à la casse :
on peut tout aussi bien écrire `<html>` `<HTML>` ou `<HtmL>`

On distingue deux types de balises :

- Les **balises en paires** (une balise ouvrante et une balise fermante)

Une balise ouvrante : **<title>** ; Et une balise fermante : **</title>** .

- Les **balises orphelines** (une seule balise) elles permettent d'insérer un élément à un endroit précis (par exemple une image). Les balises sont parfois accompagnées d'attributs pour donner des indications supplémentaires, ou paramétrer un élément : **** .

La syntaxe **<html><body></html></body>** est **incorrecte** : les balises s'entremêlent.

La syntaxe **<html><body></body></html>** est **correcte** : une balise qui est ouverte à l'intérieur d'une autre balise doit aussi être fermée à l'intérieur de celle-ci.

<html>

<head>

<meta charset="utf-8">

<title> Mon titre </title>

</head>

<body>

</body>

</html>

12 Structuration du texte

Mise en forme

Tout document Html contiendra en **majorité du texte**.

On pourra l'agrémenter par quelques balises élémentaires pour la mise en forme.

Gras	[Bold]	
Italique	[Italic]	<i></i>
Taille de caractère	[Font size]	...
Couleur de caractère	[Font color]	 ...
Retour à la ligne	[Line Break]	
Centrage	[Center]	<center></center>

13 Structuration du texte

- Exemple

```
<!DOCTYPE html>
<html>
<head><title></title></head>
<body>texte simple<br>
<b>texte en gras</b><br>
<strong>texte en gras</strong><br>
<i>texte en italique</i><br>
<em>texte en italique</em><br>
<b><i>texte en gras et en
italique</i></b><br>
<font size=5>texte</font>
<font color="#0000ff">en bleu</font>
<!--c'est fini-->
</body>
</html>
```

texte simple
texte en gras
texte en gras
texte en italique
texte en italique
texte en gras et en italique
texte en bleu

14 Structuration du texte

Les titres & les paragraphes

- Les **titres et le sous titres** sont indiqués par la balise **<h1>** à **<h6>** (heading — six niveaux de titres sont possibles).
 - Le niveau 1 est le niveau le plus haut dans la hiérarchie du document, suivi du niveau 2, etc.
- La balise **<p>** sert à créer un paragraphe
 - Habituellement, il est isolé par défaut du reste du texte par un petit espace vertical avant et après
 - le texte va automatiquement à la ligne à la fin d'un paragraphe.

- Exemple :

```
<body>
<h1>Un titre de niveau 1
</h1>

<p>Un paragraphe.</p>

<h2>Un titre de niveau
2</h2>

<p>Un paragraphe.</p>

<h3>Un titre de niveau 3
</h3>

<p>Etc.</p>

</body>
```

15 Structuration du texte

Les listes

□ Deux types de listes

- Liste non ordonnée (unordred list)
- Liste numérotée (ordred list)

liste numérotée

- Dans une liste numérotée la balise accepte deux attributs : start et type.
 - start: permet de changer le numéro du premier l'élément dans la liste, qui est par défaut à 1.
 - type: spécifie le type du système de numérotation à utiliser.
 - Cinq types de numérotation sont possibles : numérique, alphabétique majuscule ou minuscule, romain majuscule ou minuscule (1, A, a, I, i).

liste non numérotée

- La définition d'une liste non numérotée se fait d'une manière analogue à une liste numérotée en remplaçant la balise par la balise et tout en gardant la balise .
- Pour spécifier le type de puce l'attribut type a trois valeurs (disc(par défaut) , square, circle)
- L'attribut start n'a évidemment pas de sens dans une liste non numérotée.

16 Structuration du texte

Les listes

les fleurs

<ul type="circle">

Rose

Tulipe

Iris

les plantes

<ul type="disc">

Caroube

Combava

Surelle

les plantes de foret

<ul type="square">

Aigremoine
eupatoire

Berce

Callune

les plantes d'interieur

Ficus natacha

Kentia

Fougère de Boston

les fleurs

- Rose
- Tulipe
- Iris

les plantes

- Caroube
- Combava
- Surelle

les plantes de foret

- Aigremoine eupatoire
- Berce
- Callune

les plantes d'interieur

- Ficus natacha
- Kentia
- Fougère de Boston

Les tableaux

- Tableau : Ensemble de lignes elles-mêmes constituées de cellules

Définition d'un tableau	<code><table> ... </table></code>
Légende	<code><caption> ... </caption></code>
Définition d'une ligne	<code><tr> ... </tr></code>
Définition d'une cellule d'en-tête	<code><th> ... </th></code>
Définition d'une cellule normale	<code><td> ... </td></code>

Les tableaux: exemple

```
<table>
<caption>Les totaux</caption>
<tr>
<th>Jour de la semaine</th>
<th>Montant</th>
<th>Total</th>
</tr>
<tr>
<td>Lundi</td>
<td>456 euros</td>
<td>456 euros</td>
</tr>
<tr>
<td>Mardi</td>
<td>200 euros</td>
<td>200 euros</td>
</tr>
</table>
```

Les totaux

Jour de la semaine	Montant	Total
Lundi	456 euros	456 euros
Mardi	200 euros	200 euros

19 Les tableaux

Les tableaux:

- Si on souhaite y adjoindre des bordures :
 - Bordure de cadre `<TABLE border=?></TABLE>`
 - Exemple:
`<TABLE border=2>`
`<TR><TD>1</TD><TD>2</TD></TR>`
`<TR><TD>3</TD><TD>4</TD></TR>`
`</TABLE>`
- Il est possible de changer la couleur de la cellule.
 - Couleur de la cellule `<TD BGCOLOR="#$$$$$$">`
- Il y a encore trois éléments (définis par défaut mais modifiables):
 - L'espace entre les cellules ou l'épaisseur des lignes du quadrillage
 - `<TABLE cellspacing=?>`

- L'enrobage des cellules ou l'espace entre le bord et le contenu
 - `<TABLE cellpadding=?>`
- La largeur de la table
 - `<TABLE width=?>` `<TABLE width=%>`
- Exemple:
`<TABLE border=2 cellspacing=10>`
`<TR><TD>1</TD><TD>2</TD></TR>`
`<TR><TD>3</TD><TD>4</TD></TR>`
`</TABLE>`
`<TABLE border=2 cellpadding=10>`
`<TR><TD>1</TD><TD>2</TD></TR>`
`<TR><TD>3</TD><TD>4</TD></TR>`

1	2
3	4

1	2
3	4

Fusion des cellules

- Fusion de cellules (<td> ou <th>)
- Cellules sur plusieurs colonnes : attribut **colspan**

```
<table border =« 1 »>
```

```
<tr>
```

```
<td>C1</td>
```

```
<td>C2</td>
```

```
</tr>
```

```
<tr>
```

```
<td colspan="2">C12</td>
```

```
</tr>
```

```
</table>
```

C1	C2
C12	

- Cellules sur plusieurs lignes : attribut **rowspan**

```
<table border =« 1 »>
```

```
<tr>
```

```
<td rowspan="2">L12</td>
```

```
<td>L1</td>
```

```
</tr>
```

```
<tr>
```

```
<td>L2</td>
```

```
</tr>
```

```
</table>
```

L12	L1
	L2

21 Les Tableaux Structurés (HTML 5)

- Pour les gros tableaux, il est possible de les **diviser** en trois parties :

- En-tête : il se définit avec les balises **<thead></thead>**
- Corps du tableau : il se définit avec les balises **<tbody></tbody>**
- Pied de tableau : il se définit avec les balises **<tfoot></tfoot>**

- Exemple :

```
<table>
<caption>Passagers du vol 377</caption>
<thead> <!-- En-tête du tableau -->
<tr>
  <th>Nom</th>
  <th>Âge</th>
  <th>Pays</th>
</tr>
</thead>
```


```
<tfoot> <!-- Pied de tableau -->
<tr>
  <th>Nom</th>
  <th>Âge</th>
  <th>Pays</th>
</tr>
</tfoot>
<tbody> <!-- Corps du tableau -->
<tr>
  <td>Carmen</td>
  <td>33 ans</td>
  <td>Espagne</td>
</tr>
```

```
<td>François</td>
  <td>43 ans</td>
  <td>France</td>
</tr>
<tr>
  <td>Xu</td>
  <td>19 ans</td>
  <td>Chine</td>
</tr>
</tbody>
</table>
```

22 Structuration du texte

Passagers du vol 377				
En-tête du tableau	Nom	Age	Pays	<thead>
	Carmen	33 ans	Espagne	<tbody>
Corps du tableau	Michelle	26 ans	Etats-Unis	
	François	43 ans	France	
	Martine	34 ans	France	
	Jonathan	13 ans	Australie	
	Xu	19 ans	Chine	
Pied du tableau	Nom	Age	Pays	<tfoot>

Syntaxe

- L'insertion d'une image dans une page Web se fait grâce à la balise  **** (img src pour Image Source).
- Les autres attributs de la balise img:
 - L'argument **alt** permet de **décrire l'image**. Cette description est visible lorsque l'on passe sur l'image.
 - L'argument **align** définit **l'alignement de l'image** sur la page.
 - L'argument **border=n** définit un cadre d'une épaisseur de N pixels
 - Exemple:

```

```

24 Les Hyperliens

Syntaxe

La balise `<a> ` sert à définir un hyperlien (une ancre hypertextuelle) dans une page Web.

- Format: `LapagedeLapresseTunisie`

href=url(Uniform Resource Locator)

- Spécifie un lien externe ou interne (avec un # au début de l'URL, dans le 2ème cas).

target=emplacement cible:

- ``

Syntaxe	Description
target="_blank"	Ouvre la page dans une nouvelle fenêtre
target="_top"	Ouvre la page dans la même fenêtre
target="_parent"	Ouvre le document dans la fenêtre parent des frames
target="_self"	Ouvre la page dans la même frame

Types de liens

- Deux types de liens hypertextes : les liens externes et les liens internes
 - **Lien externe** : pointeur vers un document différent du document contenant l'ancre
 - **Lien interne** : permet de naviguer dans l'espace d'un seul document
- On peut définir un signet pour la création de lien interne à une page
 - **Le signet** est un nom repérant une position dans la page
 - **Exemple:** ``
On l'utilise ensuite: `Partie5 de cette page `
- Il est aussi possible de définir un **lien externe** qui pointe directement vers une étiquette définie dans un autre document.
`page de l'économie`
- Noter qu'il est possible de coller une étiquette sur un ancre. Autrement dit la même balise `<a>` peut contenir les deux attributs href et name en même temps.

26 Les Hyperliens

- Un hyperlien peut pointer vers : Une adresse E-Mail
- Exemple

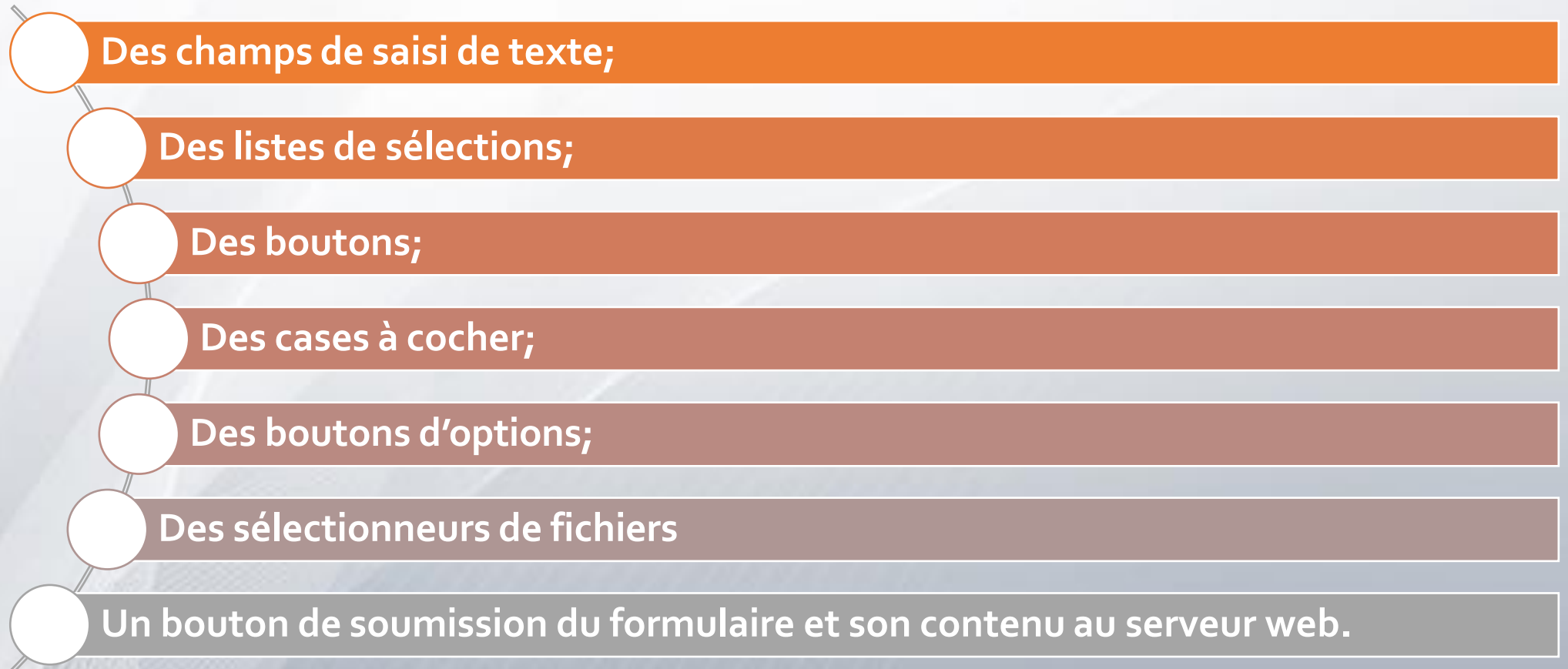
` contactez-moi`

- Un hyperlien peut pointer vers d'autres protocoles :
 - Lien FTP, Fichier, etc.
- Lien sous forme d'image
 - Pour qu'une image soit un hyperlien, il suffit d'écrire :
 - `< a href="page cible"> `
- Exemple
 - `< a href="http://www.tunisiatv.com">`

27 Les formulaires

- Les formulaires permettent la saisie de données dans une page Web
- **Définition :** `<form> ... </form>`
- Attribut **action**: URL de la page où envoyer le formulaire
- Attribut **method**: méthode de transmission des données du formulaire (valeurs possibles : *get* et *post*)

Un formulaire
peut contenir :



28 Les formulaires

Traitement du formulaire par le serveur web

- Récupération des valeurs des champs de formulaire;
- Traitements des données soumis;
- Génération d'une nouvelle page web;
- Renvoi de la page générée au client soumissionnaire du formulaire.

BALISE	ATTRIBUTS	FONCTIONS
<FORM>...</FORM>		Balise qui permet de regrouper plusieurs éléments d'un formulaire (boutons, champs ,...)
	name = "nom"	Un nom qui permet de différencier plusieurs formulaires sur une même page.
	action = " http://url " action = "mailto:url"	Envoyer les données vers une adresse Web Envoyer les données vers une adresse E-Mail.
	Method = "get" method = "post"	get: apparition des valeurs saisies en paramètres de l'URL de la page action post: valeurs saisies cachées et quantité de données saisies plus importante
	ENCTYPE = "?"	"text/plain": Formulaire classique. Pour envoyer les informations par E-Mail. "multipart/form-data": Si le formulaire doit contenir un fichier attaché.

Entrée d'une ligne de texte

BALISE	ATTRIBUTS	FONCTIONS
<INPUT OPTIONS...>	type = "text"	Indique une boîte d'une seule ligne permettant l'entrée du texte
	name = "nom"	Identifier la boîte de texte pour pouvoir repérer/identifier le contenu
	size = nombre	Spécifier la largeur de la boîte en nombre de caractères
	maxlength = n	Spécifier le nombre maximum de caractères permis dans la boîte Si ce nombre est plus grand que SIZE alors le texte se défilera à l'horizontale

Entrée d'un « Mot de Passe »

BALISE	ATTRIBUTS	FONCTIONS
<input options...>	type = "password"	Indentique à une boîte de texte sauf que l'écriture est remplacée par des * * Les mots de passe sont, malgré la saisie cachée, transmis en clair sur Internet
	name = "nom"	Donner un nom pour identifier/répérer le contenu de la boîte de dialogue.
	size = nombre	Spécifier la largeur de la boîte en nombre de caractères
	value = "texte"	Indiquer le mot de passe qui apparaîtra sous forme ***** (Pas sécuritaire!!)

31 Les formulaires

Entrée de textes dans un boîte

BALISE	ATTRIBUTS	FONCTIONS
<Textarea> </textarea>		boite de texte éditable multi-lignes. si le texte dépasse la limite verticale, une barre de déroulent verticale apparaîtra.
	name = "nom"	identifier la boîte de texte
	cols = nombre	spécifier la largeur de la boîte de texte en nombre de caractères
	rows = nombre	spécifier le nombre de lignes pour la hauteur de la boîte de texte
	value = "texte"	permet de prédéfinir du texte qui sera envoyé par défaut si le champ de saisie n'est pas modifié par l'utilisateur.
	Readonly	spécifie que la boîte est en mode lecture seulement.

Entrée d'un choix par « Bouton Radio »

BALISE	ATTRIBUTS	FONCTIONS
<INPUT OPTIONS...>	type = "radio"	Disposera des boutons où un seul choix pourra être sélectionné. Tous les boutons dans un même groupe doivent avoir le même nom.
	name = "nom"	Identifie à quel groupe de boutons le 'bouton-radio' appartient.
	value = "texte"	Indiquer le choix sélectionné. Peut être différent du choix affiché.
	Checked	Indique un choix par défaut parmi les options proposées.

Entrée de plusieurs choix par « Cases à Cocher »

BALISE	ATTRIBUTS	FONCTIONS
<INPUT OPTIONS...>	TYPE = "checkbox"	Disposera des cases où plusieurs choix pourront être sélectionnés. Toutes les cases dans un même groupe doivent avoir le même nom (NAME).
	NAME = "nom"	Identifie à quel groupe de cases la 'case à cocher' appartient.
	VALUE = "texte"	Indiquer le(s) choix sélectionné(s). Peut-être différent du choix affiché.
	CHECKED	Indique un choix pré-définie parmi les options proposées.

Liste déroulante

BALISE	ATTRIBUTS	FONCTIONS
<SELECT>... </SELECT>		Permet d'inclure une liste déroulante dans laquelle on pourra faire un choix. Les balises <OPTION> permettent de définir les choix.
	name = "nom"	Assignera un nom à l'item sélectionné par l'utilisateur.
	size = "n"	Indique le nombre d'items à afficher dans la fenêtre. Si l'attribut n'est pas spécifié, une liste déroulante est utilisé avec Size = 1)
	Multiple	Permet de sélectionner plus qu'un item en enfonçant le bouton 'Ctrl' ou 'Shift'. Si l'attribut n'est pas indiqué, un seul item de la liste pourra être sélectionné.
	Disabled	Permet de créer une liste désactivée, c'est-à- dire affichée en grisée.

Éléments (Items) d'une liste déroulantes

BALISE	ATTRIBUTS	FONCTIONS
<OPTION> ... </OPTION>		Permet de détailler les items d'une liste déroulante.
<OPTION> ... </OPTION>	value = "texte"	Indiquera la phrase qui sera envoyé lors de la sélection de l'item.
	Selected	Permet de pré-identifier un item de la liste déroulante.

Les boutons

- Bouton « envoyer »

BALISE	ATTRIBUTS	FONCTIONS
<INPUT	type = "submit"	Envoyer le formulaire à l'adresse indiquée dans <FORM ACTION="...">
	value = "texte"	Permet de modifier le texte affiché sur le bouton. (Par défaut = Submit)

- Bouton «reset»

BALISE	ATTRIBUTS	FONCTIONS
<INPUT	type = "reset"	Remise à zéro de toutes les entrées du formulaire.
	value = "texte"	Permet de modifier le texte affiché sur le bouton. (Par défaut = Reset)

Les boutons

- Bouton général

BALISE	ATTRIBUTS	FONCTIONS
<INPUT	type = "button"	Utilisé pour placer une bouton cliquable à l'écran. Sans l'utilisation de 'fonctions JavaScript', le bouton ne peut rien faire.
	name = "nom"	Donner un nom unique au bouton pour identifier le clic de la souris.
	value = "texte"	Permet de placer un ou des mots descriptifs sur le bouton.

- Bouton Image

BALISE	ATTRIBUTS	FONCTIONS
<INPUT	type = "image"	Identique au "Submit" mais le bouton est remplacé par une image. Les coordonnées x et y de l'emplacement où l'utilisateur a cliqué sur l'image seront aussi envoyées avec les autres informations du formulaire.

Sélectionneur de fichiers

BALISE	ATTRIBUTS	FONCTIONS
<INPUT	type = « file"	définir un élément pour transmettre un fichier
	name = "nom"	Donner un nom unique au bouton pour l'identifier
		On peut attribuer une valeur par défaut au champ par exemple : »c:/images/img1.png »
	maxlength	le navigateur Web devrait interpréter le chiffre noté à la suite comme la taille maximale du fichier en octets.
	accept	limiter les types de fichiers permis. Par exemple : accept =text/*: tous les fichiers texte sont acceptés

Il est important de rappeler que l'attribut `enctype="multipart/form-data"` du formulaire (`<form>`) est obligatoire quand le formulaire contient un bouton-fichier.

Les nouveautés de HTML 5:



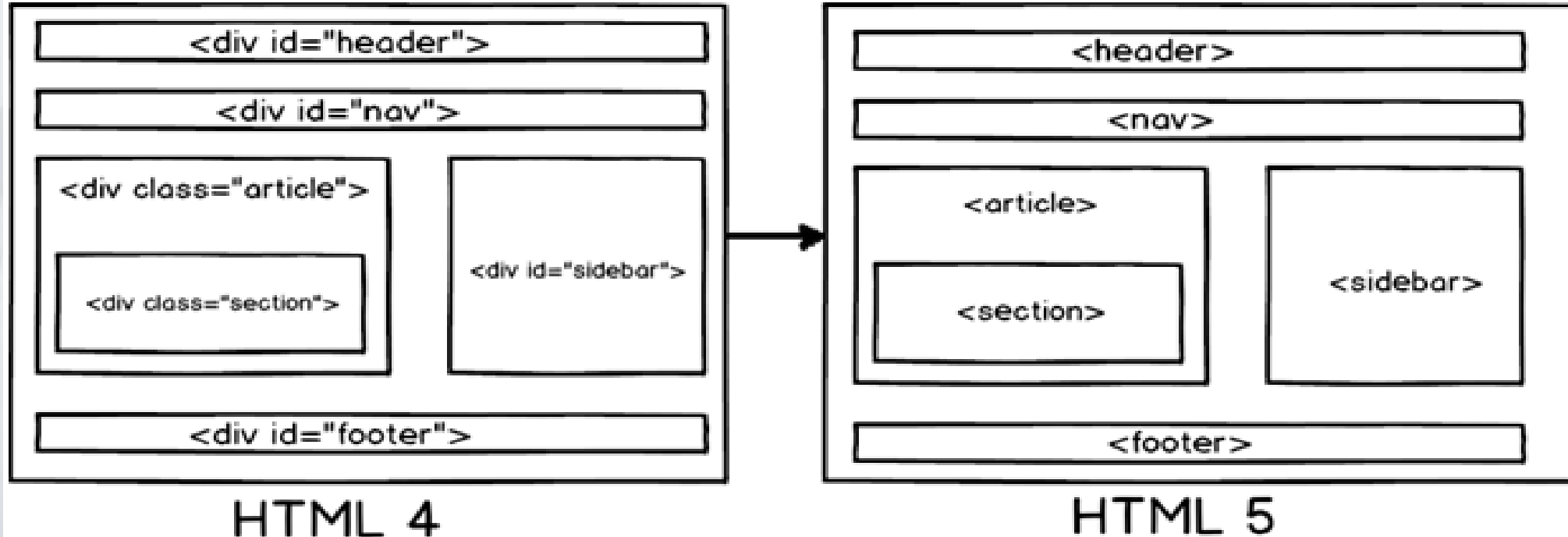
Balises sémantiques

Balises multimédia

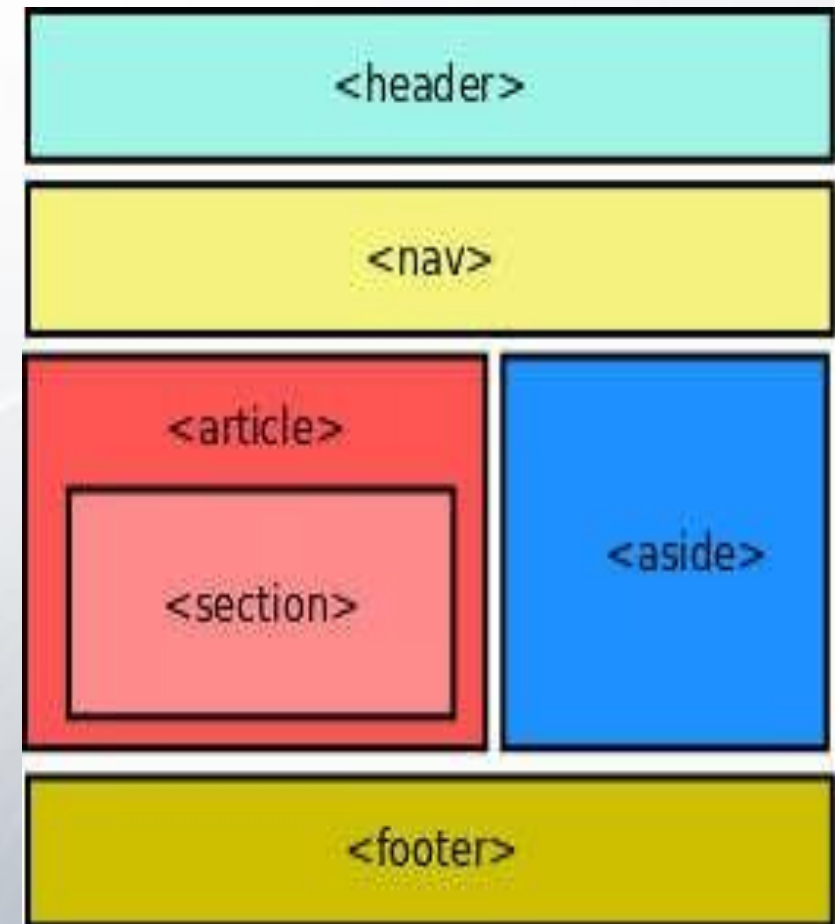
Balises de formulaires 2.0

41 Balises sémantiques

- Un **élément sémantique** décrit clairement sa signification tant pour le navigateur que pour le développeur.
- De nombreux sites Web contiennent du code HTML tel que :
`<div id="nav">` `<div class="header">` `<div id="footer">` pour indiquer la navigation, l'en-tête et le pied de page.
- En HTML5, certains éléments **sémantiques** peuvent être utilisés pour définir différentes parties d'une page Web :



- *Les balises sémantiques de HTML 5 est une nouvelle façon de penser*
- Les éléments de section (**section**, **article**, **nav**, **aside**, **header**, **footer**) segmentent des portions du document, qui possèdent une valeur sémantique particulière ;



Nom	Rôle
<code><section></code>	Section générique regroupant un même sujet, une même fonctionnalité, ou bien section d'application web
<code><article></code>	Section de contenu indépendante
<code><nav></code>	Section possédant des liens de navigation (au sein du document ou vers d'autres pages)
<code><aside></code>	Section dont le contenu est un complément par rapport à ce qui l'entoure, qui n'est pas forcément en lien direct avec le contenu mais qui peut apporter des informations supplémentaires.
<code><header></code>	Section d'introduction d'un article, d'une autre section ou du document entier (en-tête de page).
<code><footer></code>	Section de conclusion d'une section ou d'un article, voire du document entier (pied de page).

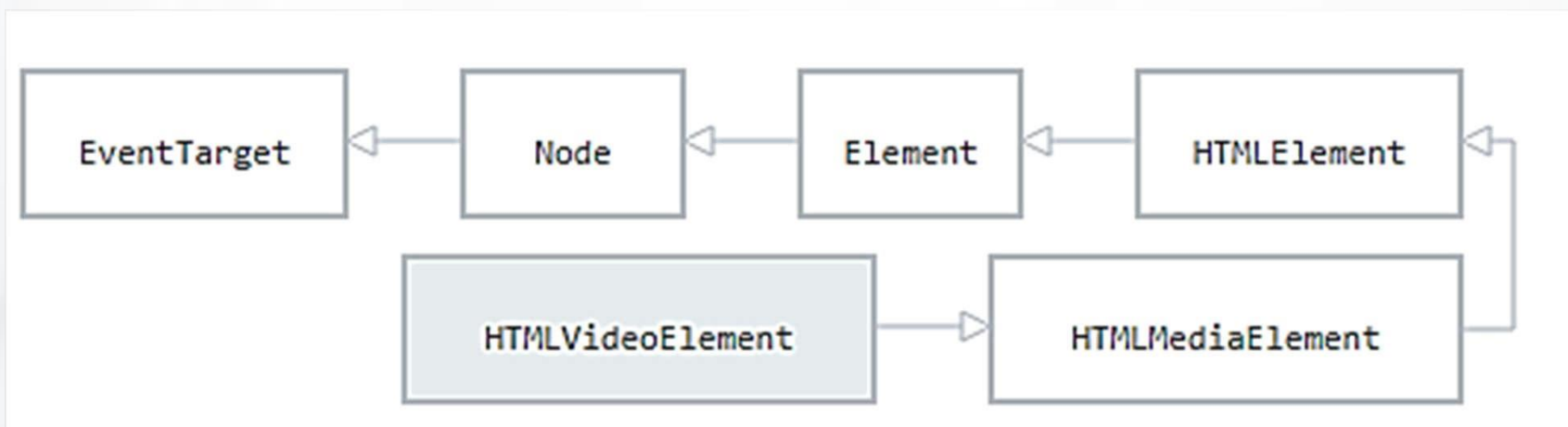
Insertion des séquences sonores et vidéo:

- Avant le HTML5, il n'est pas un standard pour lire les fichiers audio et vidéos dans les pages web.
- Ces fichiers sont lus uniquement avec un plug-in (flash par exemple).
- La balise **<audio>** spécifie une méthode standard pour insérer de l'audio dans la page web.

```
<audio controls>  
  <source src="horse.ogg" type="audio/ogg">  
  <source src="horse.mp3" type="audio/mpeg">  Votre  
navigateur ne support pas l'element audio.  
</audio>
```

HTML5 : La vidéo

- La balise `<video>` spécifie une méthode standard pour insérer de la vidéo dans la page web.



```
<video width="320" height="240" autoplay>  
  <source src="movie.mp4" type="video/mp4">  
  <source src="movie.ogg" type="video/ogg">  
  Votre navigateur ne supporte pas la balise video.  
</video>
```

➔ Plusieurs balises `<source>` : plusieurs versions de la vidéo encodée avec différents codecs. Le navigateur choisit automatiquement le premier format qu'il peut lire

- Les attributs :
- **preload="auto"**
permet de spécifier au navigateur de débuter le téléchargement de la vidéo tout de suite, en anticipant sur le fait que l'utilisateur lira la vidéo.
- **autoplay="true"**
comme son nom l'indique, permet de lancer la lecture automatiquement.
- **poster="img.jpg"**
permet d'indiquer une image à afficher par défaut dans l'espace réservé par la vidéo, avant la lecture.
 - **Loop**: indique que la lecture doit s'effectuer en boucle.

- **videoHeight / videoWidth** retourne un entier contenant l'**hauteur / largeur** de la ressource en pixels CSS, en tenant compte de: les dimensions, aspect ratio, résolution, etc., comme définit pour le format utilisé par la ressource.
 - Si la propriété **readyState** de l'élément est **HAVE_NOTHING**, la valeur = **0**.
- **height / width**
une propriété de type **DOMString** qui reflète l'attribut HTML **l'hauteur / largeur**, qui spécifie l'**hauteur / largeur** de la zone d'affichage, en pixels CSS.

Canvas

- L'élément est utilisé pour dessiner des graphiques sur une page Web.
- L'élément HTML `<canvas>` est utilisé pour dessiner des graphiques, à la volée, via JavaScript.
- L'élément `<canvas>` n'est qu'un conteneur pour les graphiques. Il faut utiliser JavaScript pour dessiner les graphiques.
- Canvas a plusieurs méthodes pour dessiner des lignes, des rectangles, des cercles, du texte et l'ajout d'images.

Canvas: exemple

```
<canvas id="myCanvas" width="200" height="100"  
style="border:1px solid #000000;">  
</canvas>
```

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

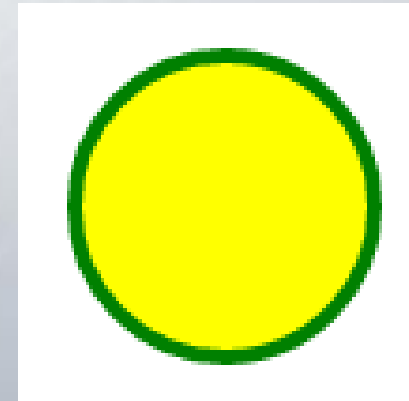


SVG

- SVG signifie Scalable Vector Graphics
- SVG sert à définir des graphiques pour le Web
- SVG est une recommandation du W3C
- L'élément HTML <svg> est un conteneur pour les graphiques SVG.
- SVG dispose de plusieurs méthodes pour dessiner des chemins, des boîtes, des cercles, du texte et des images graphiques.

SVG:Exemple

```
<!DOCTYPE html>
<html>
<body>
<svg width="100" height="100">
  <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />
</svg>
</body>
</html>
```





Google Maps : Google Maps permet d'afficher des cartes sur une page Web

Example

```
<div id="map" style="width:400px;height:400px">
```

```
<script>
function myMap() {
    var mapOptions = {
        center: new google.maps.LatLng(51.5, -0.12),
        zoom: 10,
        mapTypeId: google.maps.MapTypeId.HYBRID
    }
    var map = new google.maps.Map(document.getElementById("map"),
    mapOptions);
}
</script>
```


52 Balises de formulaires 2.0

- Avec Html 4, les types de champs n'étaient pas nombreux.
- HTML5 apporte plus d'une douzaine de nouveaux types.
- HTML5 introduit de nombreuses nouveautés pour les formulaires pour améliorer l'aide à la saisie et les contrôles disponibles pour l'utilisateur.
- Plusieurs attributs simples à mettre en place améliorent la prise en charge des formulaires, tout en se passant de JavaScript.

- `<input type=`

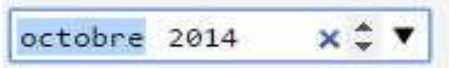
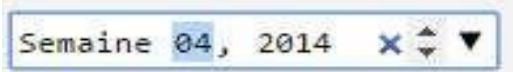
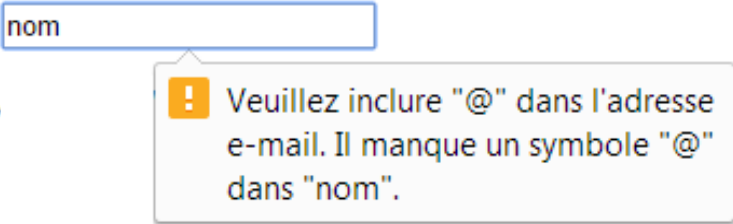
- `search`
- `tel`
- `email`
- `url`
- `date`
- `number`
- `range`
- `Color ..`

53 Balises de formulaires 2.0


Champs	Syntaxe
Date	<code><input type="date" name="d" value="<?php echo \$today?>"></code>
Date et temps	<code><input type="datetime" name="dt" value="<?php echo \$now?>"></code>
Time	<code><time>2013-06-27</time></code>
Nombre	<code><input type="number" name="nombre" value="12345"></code>

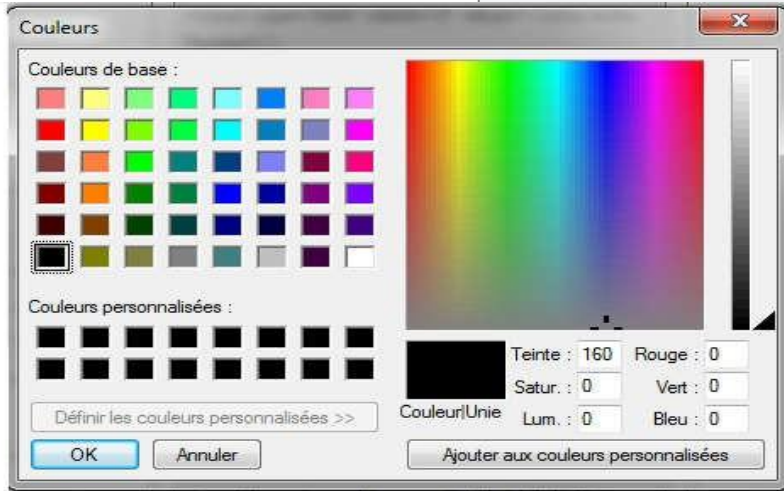


54 Balises de formulaires 2.0



Champs	Syntaxe
Month	<code><input type="month" name="holidays"></code> 
Week	<code><input type="week" name="int_week"></code> 
datetime-local	<code><input type="datetime-local" name="bdaytime"></code> Note: type="datetime-local" n'est pas supporté par Firefox et Internet Explorer.
email	<code><input type="email" name="email"></code> 

55 Balises de formulaires 2.0

Champs	Syntaxe
Couleur	<code><input type="color" name="couleur" value=""></code> 
Champ de recherche	<code><input type="search" name="" list="datalist" value=""></code>
Data	<code><data value="10">Dix</data></code>



56 Balises de formulaires 2.0

Champs	Syntaxe
Range	<div><pre><input type="range" min="-100" max="100" value="0" step="2" name="power" list="powers"></pre><pre><datalist id="powers"></pre><pre><option value="0"></pre><pre><option value="-30"></pre><pre><option value="30"></pre><pre><option value="+50"></pre><pre></datalist></pre></div> <div>A horizontal range slider with a light gray track. A small gray handle with three horizontal lines is positioned in the center. There are tick marks at the ends of the track.</div>
Progress	<div><pre><progress id="prog" max=100</pre></div> <div>A horizontal progress bar consisting of a row of 10 small squares. The first 8 squares are light gray, and the last 2 squares are filled with a dark blue color.</div>

- **HTML5 a ajouté les attributs suivants au champ <input>:**
 - autocomplete
 - autofocus
 - form
 - formaction
 - formenctype
 - formmethod
 - formnovalidate
 - formtarget
 - height et width
 - list
 - min et max
 - multiple
 - pattern (regexp)
 - placeholder
 - required
 - step
- **Les 2 attributs suivants à l'élément <form>:**
 - autocomplete
 - novalidate

58 Balises de formulaires 2.0

Exemples pour les nouveaux attributs :

```
<form action="/action_page.php" method="get"> First  
  name: <input type="text" name="fname"><br> Last name:  
  <input type="text" name="lname"><br>  
  <input type="submit" value="Submit">  
  <input type="submit" formmethod="post" formaction="action_page_post.asp"  
  value="Submit using POST">  
</form>
```

```
<form action="/action_page.php">  
  First name: <input type="text" name="fname"><br> Last  
  name: <input type="text" name="lname"><br>  
  <input type="submit" value="Submit as normal">  
  <input type="submit" formtarget="_blank"  
  value="Submit to a new window">  
</form>
```

Exemples pour les nouveaux attributs :

L'attribut **pattern** spécifie une expression régulière sur laquelle la valeur de l'élément `<input>` est vérifiée.

```
<input type="text" name="country_code" pattern="[A-Za-z]{3}" title="Three letters">
```

L'attribut **placeholder** spécifie un indice qui décrit la valeur attendue d'un champ de saisie. L'indice est affiché dans le champ de saisie avant que l'utilisateur n'écrive une valeur.

```
<input type="text" name="fname" placeholder="First name">
```

Les 2 attributs **pattern** et **placeholder** fonctionnent avec les types d'entrée suivants:
text, search, url, tel, email, password

Les éléments suivants de Html 4 ont été retirés de Html 5

- <acronym>
- <applet>
- <basefont>
- <big>
- <center>
- <dir>
-
- <frame>
- <frameset>
- <noframes>
- <strike>
- <tt>

- La déclaration de type de document (*doctype*) indique la **définition de type de document** (DTD) en vigueur pour le document.
- Tous les documents HTML doivent commencer par une déclaration. <!DOCTYPE>
- La déclaration n'est pas une balise HTML. Il s'agit d'une "**information**" destinée au navigateur sur le type de document auquel il peut s'attendre.
- Un **document HTML valide** doit donc **déclarer la version HTML** qu'il utilise :
 - Pour HTML 5, il faut préciser : **<!DOCTYPE html>**
 - HTML 4.01 spécifie trois DTD (**Strict**, Transitoire et Frameset) et les DTD varient selon les éléments qui y sont gérés. Par exemple :

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">
```

Le **processus de détection de l'encodage** a également été modifié et s'effectue dans l'ordre

- la vérification de la présence d'un **en-tête HTTP Content-Type** :

//version courte en HTML5 `<meta charset="utf-8">`

//ou version longue `<meta http-equiv="Content-Type" content="text/html; charset= UTF-8">`

- ensuite la détection du BOM (Byte Order Mark) en début de fichier, qui indique l'utilisation d'un encodage unicode ainsi que l'ordre des octets.

Le choix de l'UTF-8 est désormais préconisé par le W3C.

- La **<meta>** balise définit les métadonnées sur un document HTML. Les métadonnées sont des données (informations) sur les données.
- **<meta>** Les balises se trouvent toujours à l'intérieur de l'élément **<head>** et sont généralement utilisées pour spécifier le jeu de caractères, la description de la page, les mots-clés, l'auteur du document et les paramètres de la fenêtre d'affichage.
- Les **métadonnées** ne seront pas affichées sur la page, mais sont analysables par la machine.
- Les **métadonnées** sont utilisées par les navigateurs ,les moteurs de recherche (mots-clés) et d'autres services Web.

Définir des mots-clés pour les moteurs de recherche :

```
<meta name="keywords" content="HTML, CSS, JavaScript">
```

Définissez une description de votre page Web :

```
<meta name="description" content="Free Web tutorials for HTML and CSS">
```

Définir l'auteur d'une page :

```
<meta name="author" content="John Doe">
```

64 L'élément META

On peut utiliser l'élément META pour spécifier les informations par défaut sur un document dans les cas suivants :

- le langage de script par défaut;
- le langage de feuille de style par défaut;
- l'encodage de caractères du document.

Attribute	Value	Description
charset	<i>character_set</i>	Specifies the character encoding for the HTML document
content	<i>text</i>	Specifies the value associated with the http-equiv or name attribute
http-equiv	content-security-policy content-type default-style refresh	Provides an HTTP header for the information/value of the content attribute
name	application-name author description generator keywords viewport	Specifies a name for the metadata

- Le corps du document (élément **BODY**) représente **le contenu du document**.

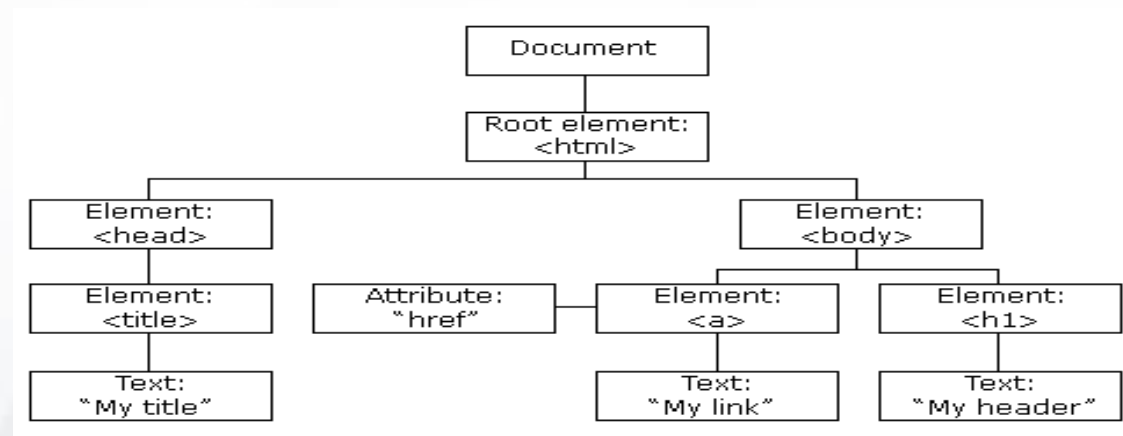
On peut assimiler le corps à un canevas dans lequel s'inscrit le contenu : le texte, les images, les couleurs, les graphiques, etc.

Puisque les feuilles de style (CSS) sont désormais le moyen conseillé de spécifier la présentation d'un document, les attributs de présentation des éléments sont donc proscrits.

Le W3C recommande aux auteurs et aux développeurs de séparer la structure et la présentation.

66 DOM (Document Object Model)

- Lorsqu'une page Web est chargée, le navigateur crée un **modèle d'objet de document** de la page.
- Le modèle **HTML DOM** est construit comme une arborescence d' **Objets** :



Le HTML DOM est un modèle **objet** standard et **une interface de programmation** pour HTML.

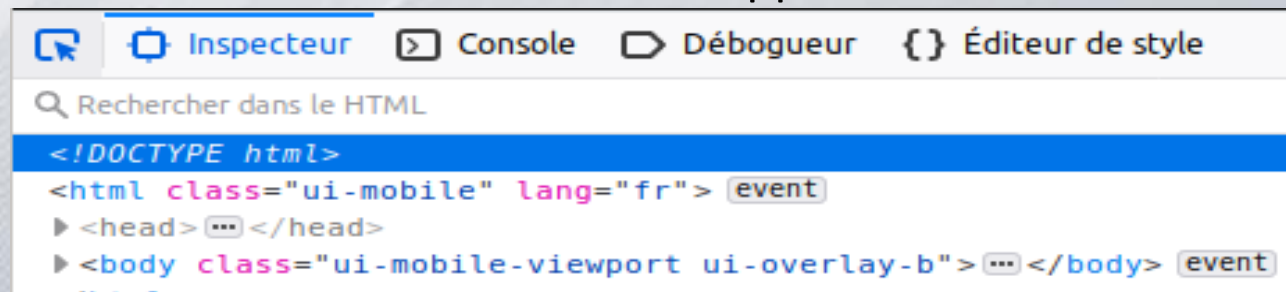
Il définit :

- Les éléments HTML en tant **qu'objets**
- Les **propriétés** de tous les éléments HTML
- Les **méthodes** pour accéder à tous les éléments HTML
- Les **événements** pour tous les éléments HTML

En d'autres termes : **le HTML DOM est un standard permettant d'obtenir, de modifier, d'ajouter ou de supprimer des éléments HTML.**

- Avec l'introduction des CSS (*Cascading Style Sheets*) et du DOM (*Document Object Model*), il a fallu considérer que les documents HTML ont une véritable structure en arbre, avec un élément racine contenant tous les autres éléments.
- Le DOM est une recommandation du W3C qui décrit une interface indépendante de tout langage de programmation et de toute plate-forme, permettant à des programmes informatiques (et à des scripts) d'accéder ou de mettre à jour le contenu, la structure ou le style de documents.
- Le DOM est une interface de programmation d'applications (API) pour documents HTML et XML. Il définit la structure logique des documents et la manière dont un document est accédé et manipulé. Spécification : <http://xmlfr.org/w3c/TR/REC-DOM-Level-1/>

On peut « voir » le DOM d'un document avec les outils de développement web inclus dans les navigateurs modernes :



Les éléments DIV et SPAN, en conjonction avec les attributs id et class, offrent un mécanisme générique de regroupement d'éléments qui rajoute de la structure aux documents.

Ces éléments définissent le contenu :

- comme étant en-ligne (**SPAN**) : les éléments en-ligne ne peuvent contenir que des données et d'autres éléments en-ligne.
- ou de bloc (**DIV**) : les éléments de bloc créent des structures « plus grandes » : ils peuvent contenir des éléments en-ligne et d'autres éléments de bloc.

Ces éléments n'imposent aucune présentation sur le contenu et sont notamment très utilisés conjointement avec les feuilles de style (CSS).

L'attribut **id** assigne un **identifiant unique à un élément**. L'attribut id a plusieurs rôles dans HTML

- comme sélecteur dans une feuille de style;
- comme ancre cible de liens hypertextes;
- comme moyen d'appeler un élément particulier à partir d'un script;
- comme nom d'un élément OBJECT déclaré;
- pour un traitement universel par les agents utilisateurs (par exemple, pour identifier les champs lors de l'extraction des données d'un formulaire à partir d'un script PHP, etc.).