

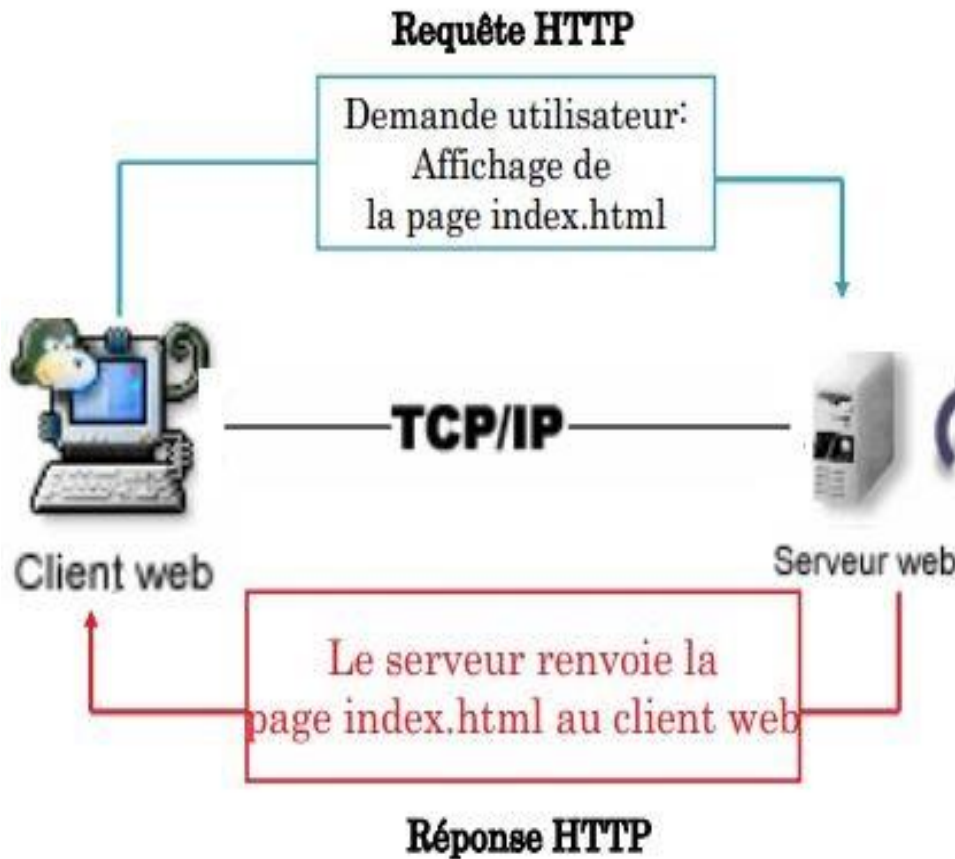
Introduction aux technologies WEB



Pr. Hajar IGUER
hajar.iguer@uic.ac.ma



Fonctionnement du WEB



1. Le navigateur effectue une requête spécifiée à travers l'URL
2. Le serveur retourne un flot de données
3. Le navigateur interprète le flot de données et l'affiche

- Décodage de la requête HTTP
- localisation de la page index.html
- construction de la réponse HTTP

HTTP (HyperText Transfer Protocol): protocole du web : protocole de communication client serveur:
 But: permettre un transfert de fichiers (essentiellement au format HTML) localisés grâce à une chaîne de caractères appelée URL entre un navigateur (le client) et un serveur Web

Serveurs Web

- Un serveur Web peut être:
 - Un ordinateur tenant le rôle de serveur informatique sur lequel fonctionne un logiciel serveur HTTP;
 - Le serveur HTTP lui-même;
 - Un ensemble de serveurs permettant le fonctionnement d'applications Web.
- Le serveur Web est responsable de:
 - Accepter les demandes de contenu décrit par l'url
 - Vérifier si l'accès est autorisé et demander l'authentification, si nécessaire
 - Envoi du contenu au navigateur
- Il existe plusieurs serveurs Web
 - Apache Windows/ UNIX/Mac OS X (gratuit) (www.apache.org)
 - Microsoft IIS(Internet Information Services) :le serveur Web payant de Microsoft (Windows)

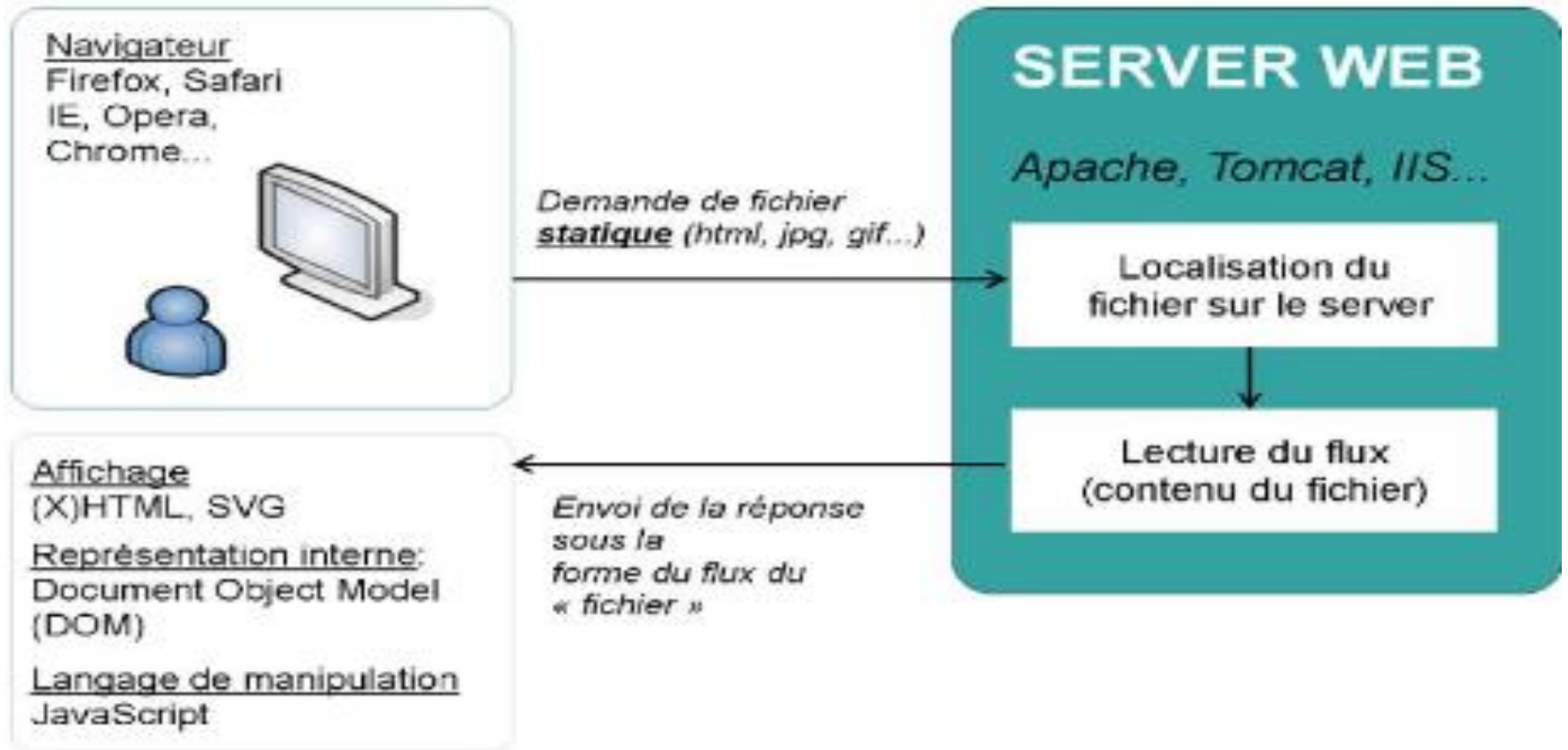


Serveurs Web

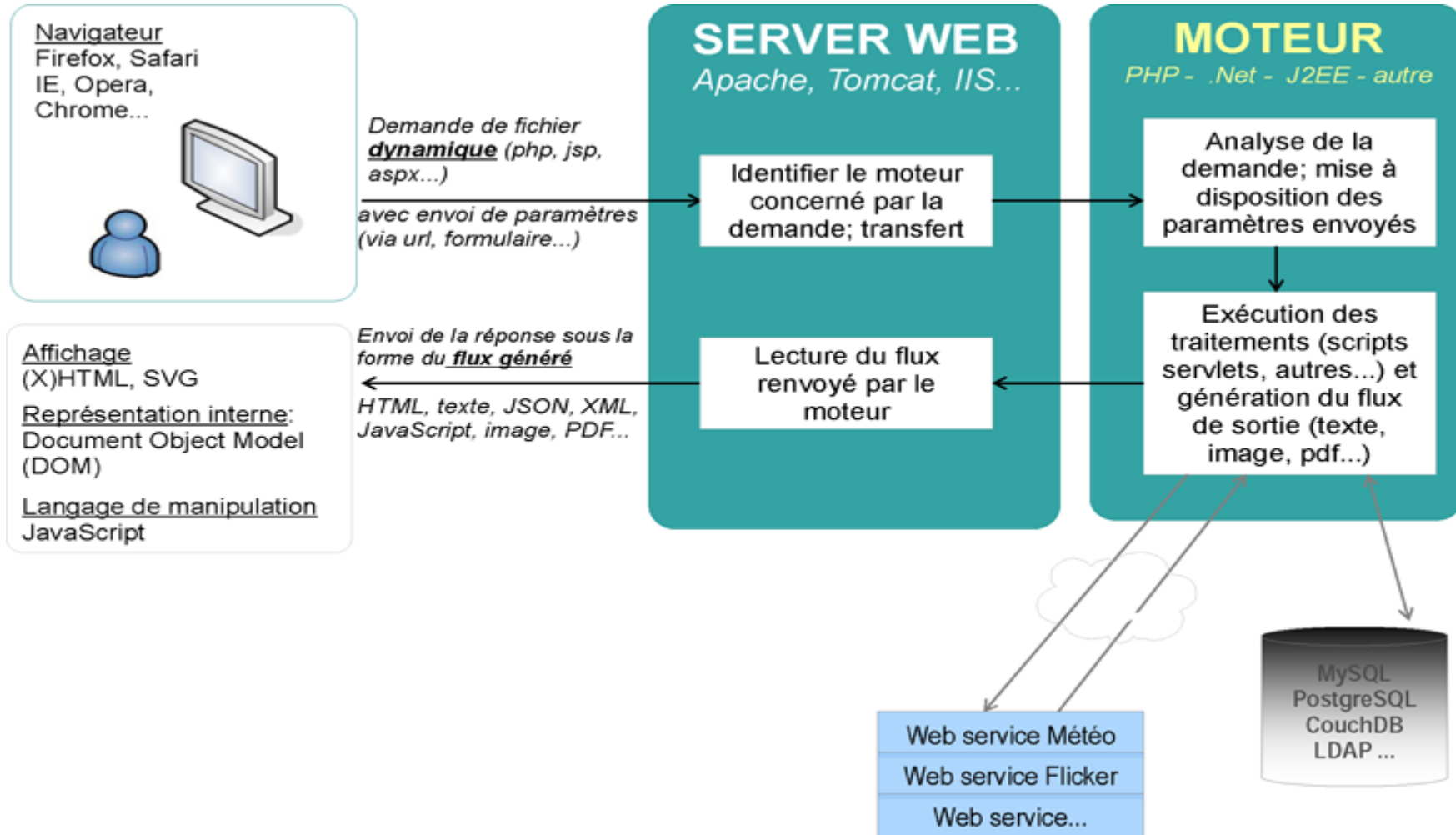
- Les navigateurs et les serveurs WEB communiquent via HTTP.
- HyperText Transfer Protocol est le protocole standard pour le transfert de contenu Web
- Le serveur écoute sur le port 80 en attendant les connexions .
- Le navigateur se connecte au serveur, et envoie une demande .
- Le serveur répond par un code d'erreur(ex : 404) ou le contenu Web.



Fonctionnement de Site Web Statique



Fonctionnement de Site Web Dynamique



Le Langage HTML/ XHTML

HTML



Langage à balises HTML

Nous innovons pour votre réussite !

- Un fichier html est un fichier texte contenant des balises appelant des commandes dont l'**action** est limitée entre le **début** et la **fin de la balise**
- Un fichier dont l'extension est **.htm** ou **.html**



Structure d'un document HTML4

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML
 4.01//EN" "http://www.w3.org/TR/html4/
strict.dtd">
<HTML>
<HEAD>
<TITLE>Page HTML 4 Type</TITLE>
</HEAD>
<BODY>
<P>Hello world! </P>
</BODY>
</HTML>
```



Structure d'un document HTML5

```
<!DOCTYPE html>  
<html>  
<header>  
<title>Page HTML 5 Type</title>  
<meta http-equiv="content-type" content="text/html; charset=utf-8" />  
</header>  
<body>  
<p>Hello world!</p>  
</body>  
<footer>  
</footer>  
</html>
```

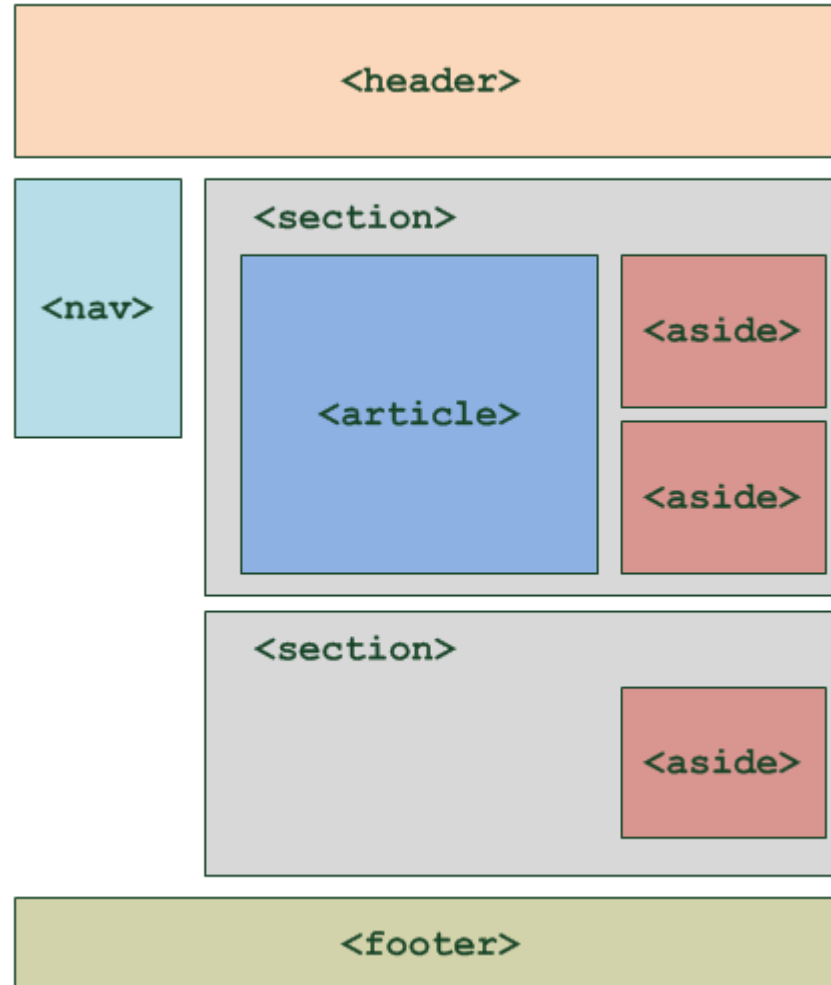


L'évolution vers le HTML5

- Avantages: Introduction de nouvelles API
 - Meilleure gestion des erreurs
 - Plus de contrôle moins de JavaScript
 - Réduire le besoin de plugins externes (flash)
 - L'élément <canvas> pour les dessin 2D
 - Les balises d'application Web <video> and <audio> pour le multimédia
 - Gestion d'un stockage local
 - De nouvelles balises de structuration: <article>, <footer>, <header>, <nav>, <section>
 - Des contrôles pour les formulaires : calendar, date, time, email, url, search



Structure d'une page HTML5



Structure d'une page HTML5

- Créer une partie de navigation du SiteWeb
 - `<nav>... </nav>`
- Créer des sections
 - `<section></section>`: blocs qui vont contenir d'autres balises.
- Créer un bloc d'informations complémentaires
 - `<aside> </aside>`
- Ajouter un article indépendant
 - `<article>.....</article>`



Structure d'une page HTML5

- Exemple de la structure de l'en-tête:

```
<header>
```

```
  <!-- Placez ici le contenu de l'en-tête de votre  
page -->
```

```
</header>
```

- Exemple de la structure du pied de page:

```
<footer>
```

```
  <!-- Placez ici le contenu du pied de page -->
```

```
</footer>
```



Structure d'une page HTML5

Exemple d'une structure de la balise navigation :

```
<nav>
```

```
<ul>
```

```
<li><a href="index.html">Accueil</a></li>
```

```
<li><a href="forum.html">Test</a></li>
```

```
</ul>
```

```
</nav>
```



Structure d'une page HTML5

- Exemple de la structure d'une section d'une page:

```
<section>
```

```
  <h1>Ma section de page</h1>
```

```
  <p>Paragraphe 1</p>
```

```
</section>
```

- Exemple de la structure d'une section supplémentaire

```
<aside>
```

```
  <!-- Placez ici des informations complémentaires -->
```

```
</aside>
```



Structure d'une page Web

Nous innovons pour votre réussite !

- Exemple de la structure d'un article:

```
<article>
```

```
  <h1>Mon article</h1>
```

```
  <p>Paragraphe 2</p>
```

```
</article>
```



Du HTML au XHTML

- Le XHTML est une évolution du HTML dans laquelle la syntaxe est plus rigide, mais plus rigoureuse : cela simplifie beaucoup la maintenance des pages web.
- Aussi, rien de compliqué dans le passage du HTML 4 au XHTML 1. Il suffit d'adopter quelques règles de notation et surtout d'utiliser les feuilles de style CSS 2 (Cascading Style Sheets) qui complètent cette norme, pour ne pas mélanger le contenu du texte et sa mise en forme



Langage à balises XHTML

Nous innovons pour votre réussite !

- Le XHTML est une évolution du HTML, initiales de « Hypertext Markup Language », c'est-à-dire langage de marquage hypertexte.
- Le «X» de XHTML vient de XML, soit «eXtensible Markup Language », langage plus complexe et plus strict que le HTML.
- C'est lui qui a inspiré la transition du HTML vers la forme plus rigoureuse qu'est le XHTML.



Structure d'un document XHTML 1.0

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en"
  lang="en">
<head>
<title>Page XHTML Type</title>
<meta http-equiv="content-type" content="application/xhtml+xml;
  charset=utf-8" />
</head>
<body>
<p>Hello world!</p>
</body>
</html>
```



EN XHTML : Règles

- **Toute balise ouvrante doit être fermée**, et les balises dites «vides» sont écrites avec une barre oblique finale (exemple: `
`).
- Les noms des balises et des attributs sont écrits en minuscules.
- Les valeurs des attributs sont encadrés par des 'quotes' (apostrophes droites) ou des "double quotes" (guillemets droits).
- Chaque attribut doit avoir une valeur (pas d'attribut vide comme checked, qui doit être écrit `checked="checked"`).
- Les éléments HTML doivent être correctement imbriqués
(`contenu` et pas `contenu`).



Convertir une page du HTML au XHTML

- Ajouter un XHTML `<!DOCTYPE>` de la première ligne de chaque page
- Ajouter un attribut `xmlns` à l'élément HTML de chaque page
- Changer tous les noms d'éléments en minuscules
- Fermez tous les éléments vides
- Changer tous les noms d'attributs en minuscules
- Citer toutes les valeurs d'attributs



Passage du HTML au XHTML

- Saut de ligne: `
` au lieu de `
`
- Tracer d'une ligne horizontale : `<hr />` au lieu de `<hr>`
- Texte en gras : `` au lieu de ``
- Texte en italique : `<i>` au lieu de ``
- Image: `` au lieu de ``



Pourquoi XHTML qu' HTML?

- XHTML permet un apprentissage plus simple de HTML. Pour deux raisons:
- comme il est **plus strict**, on évite des erreurs d'interprétation des navigateurs (si on imbrique mal les éléments, que doit faire le navigateur?);
- comme les **écritures possibles sont moins nombreuses** (pas de balises tour à tour en minuscules ou majuscules, pas de balises fermantes un coup présentes et un coup manquantes...), la syntaxe se maîtrise plus facilement



La balise `<head>``</head>`

- L'élément `<head>` est un conteneur pour tous les éléments d'en-tête.
 - La balise `<title>` définit le titre du document.
 - La balise `<base>` spécifie la base URL / cible pour toutes les URL relatives à une page.
 - La balise `<link>` définit la relation entre un document et une ressource externe. Elle est la plus utilisée pour lier les feuilles de style.
 - La balise `<style>` est utilisé pour définir les informations de style pour un document HTML.
 - La balise `<meta>` fournit des métadonnées sur le document HTML. Meta éléments sont généralement utilisés pour spécifier la description de la page, mots clés, auteur du document,..etc



Propriétés des balises

- Un tag est un nom d'élément entre crochets angulaires
- Certains éléments n'ont pas de contenu par exemple `
` ou `<hr>`
- Des éléments peuvent avoir des propriétés (attributs)
- Les attributs et leurs valeurs apparaissent dans la balise de début par exemple `<div id="section1">`



Caractères spéciaux

Nous innovons pour votre réussite !

- Il existe des codes HTML pour plusieurs **caractères spéciaux**
- Toutefois les navigateurs actuels reconnaissent les caractères accentués, ainsi vous pouvez entrer des caractères accentués directement sous votre éditeur de texte, mais votre page HTML **risque d'être illisible** depuis l'étranger ou à partir de certains navigateurs.
- Exemples:
 - <: < &: &
 - >: > " : "

