

CSI 3540

Structures, techniques et normes du Web

Langages de balisage

Objectifs:

- Créer des documents XHTML (statiques)
- Comprendre les choix sur lesquels reposent ces standards

Lectures:

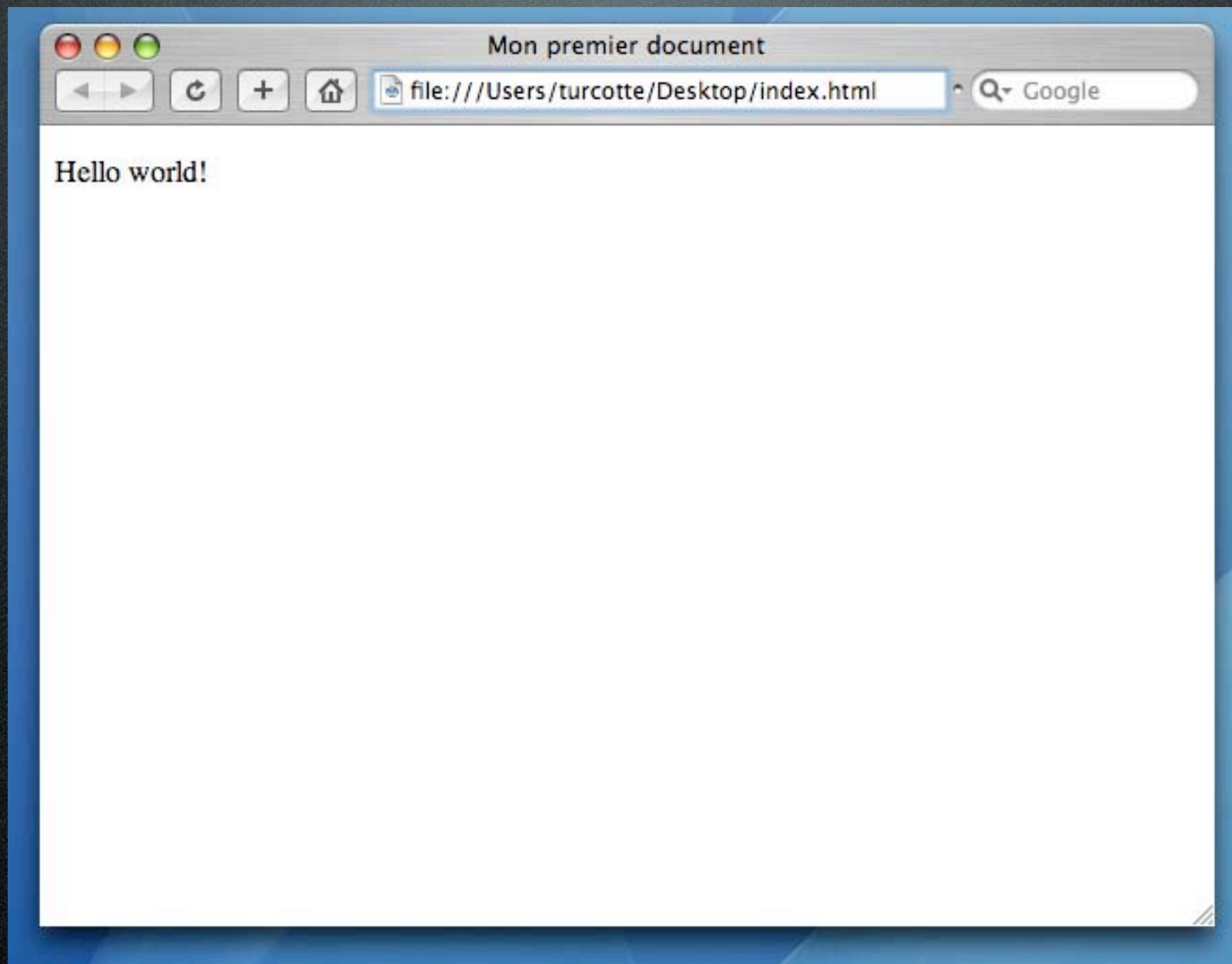
- Web Technologies (2007) pages 56–98

Plan

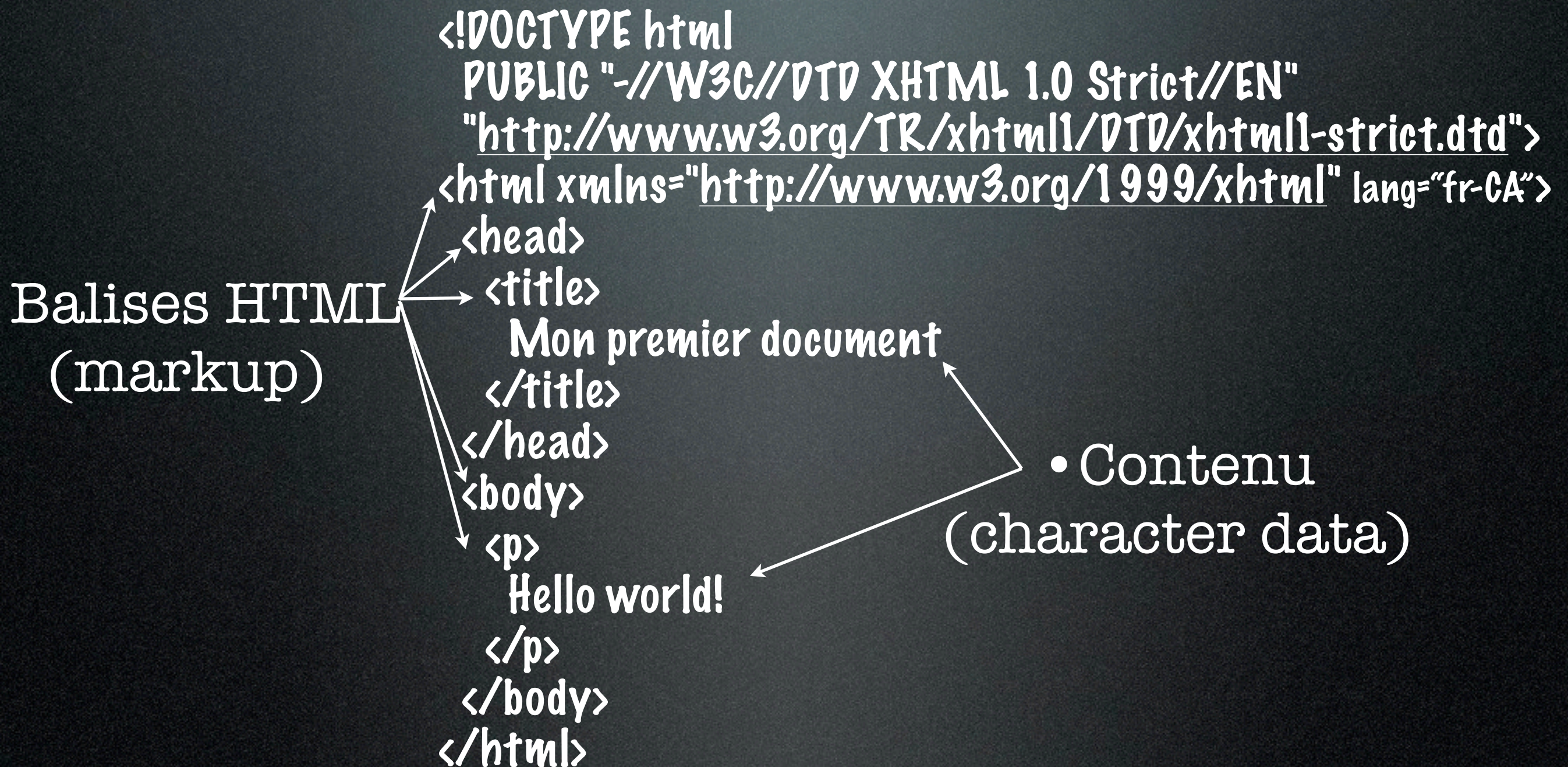
1. Concepts de base du langage HTML
2. Historique du langage
3. Éléments du langage XHTML 1.0

XHTML 1.0 (Strict)

```
<!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" lang="fr-CA">
<head>
  <title>
    Mon premier document
  </title>
</head>
<body>
  <p>
    Hello world!
  </p>
</body>
</html>
```

2 types d'information



déclaration de type du document

```
<!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" lang="fr-CA"> instance  
  <head>  
    <title>  
      Mon premier document  
    </title>  
  </head>  
  <body>  
    <p>  
      Hello world!  
    </p>  
  </body>  
</html>
```


- Indique la **version** de (X)HTML utilisée pour le codage de l'instance
- Indique l'**élément racine** de de l'instance, toujours **html** pour les documents HTML
- L'élément utilisé pour la racine n'apparaît qu'une seule fois dans tout le document
- L'attribut **xmlns** est obligatoire pour les documents XHTML 1.0 Strict

Éléments du langage

Balise (tag)
d'ouverture

```
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr-CA">
```

```
<head>
```

```
<title>
```

```
  Mon premier document
```

```
</title>
```

```
</head>
```

Contenu

```
<body>
```

```
<p>
```

```
  Hello world!
```

```
</p>
```

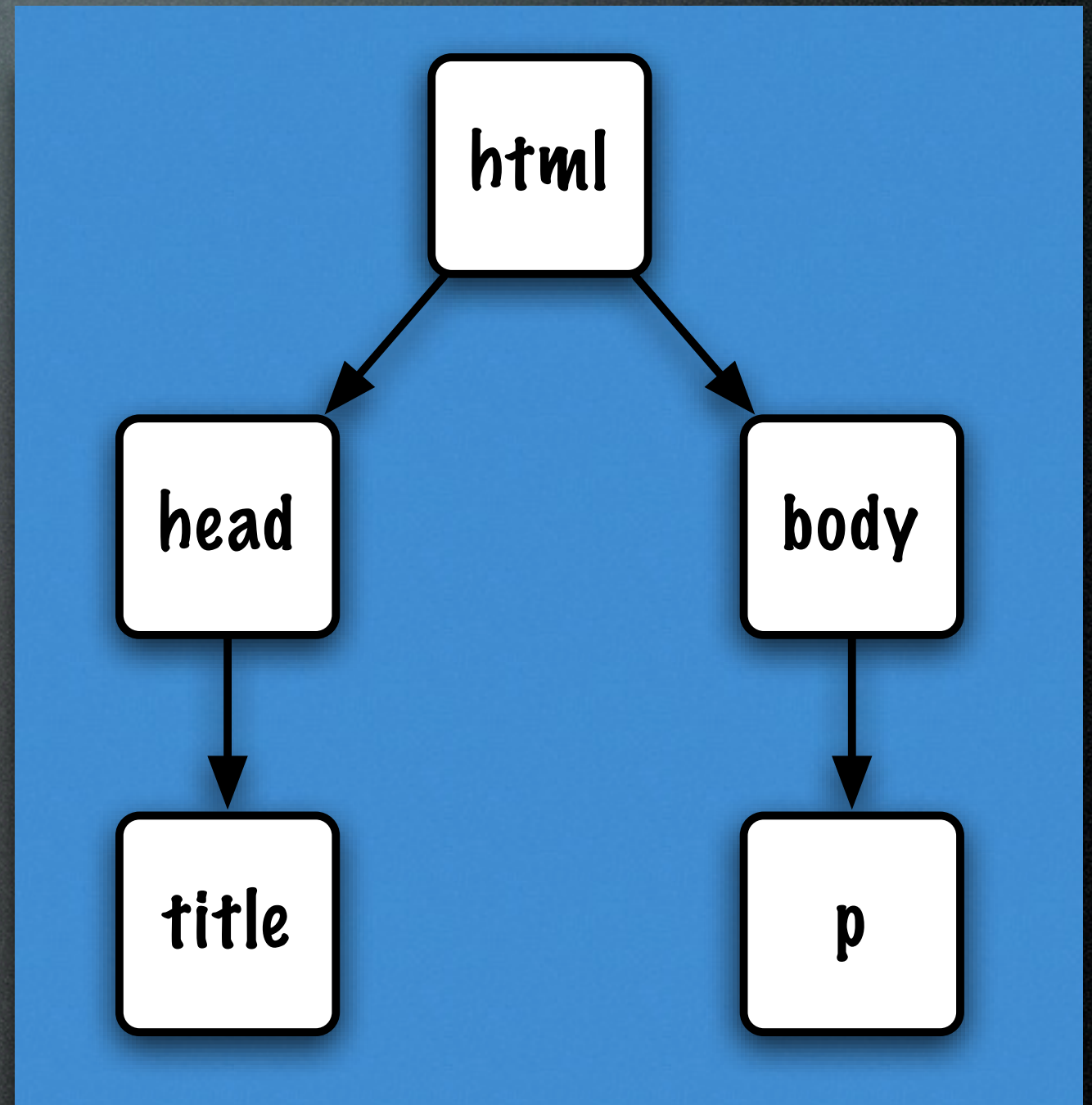
```
</body>
```

Balise (tag) de
fermeture

```
</html>
```

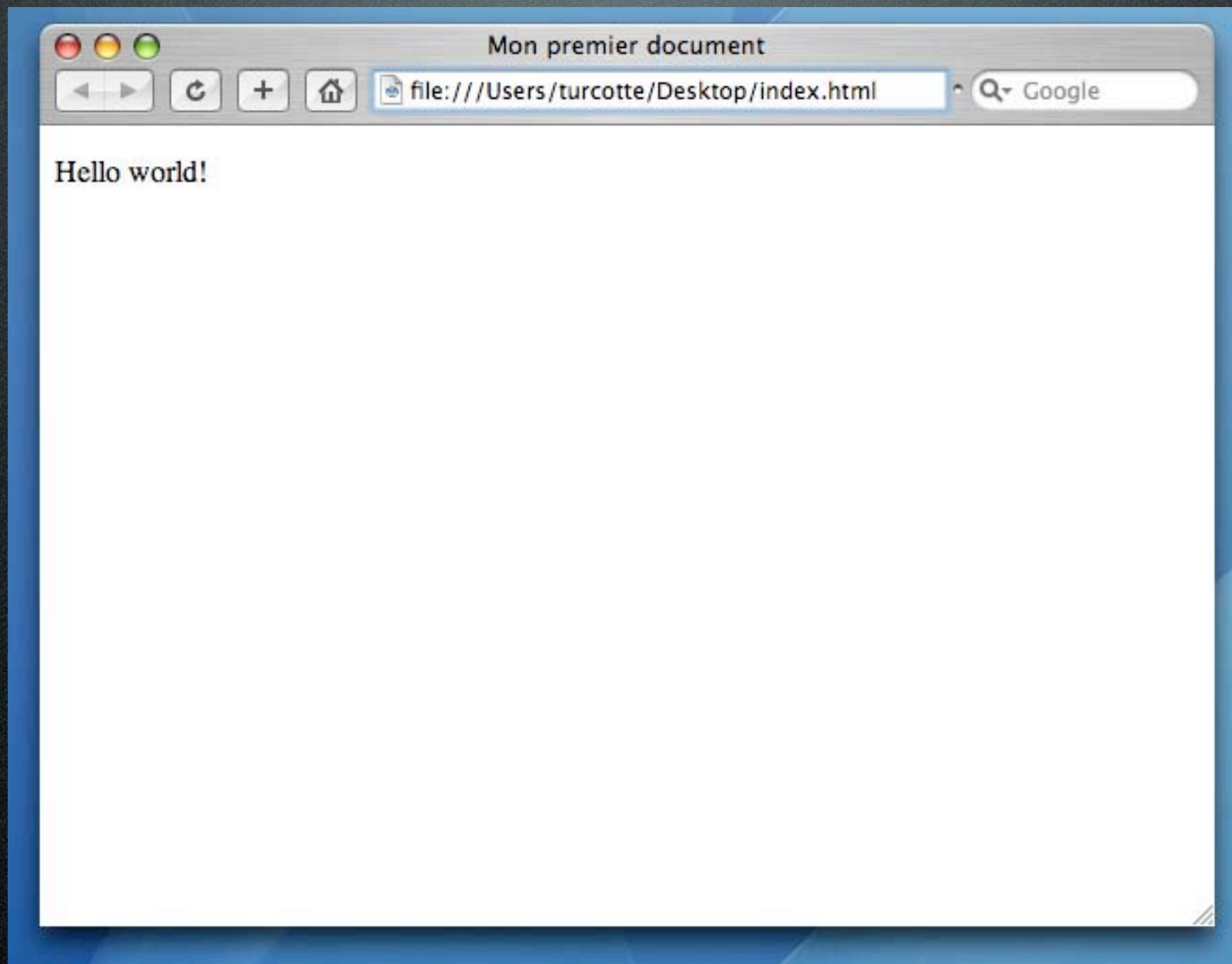

- Les chaînes `< ... >` sont des **balises** (tags)
- Il y a des balises de début (`< ... >`) et des balises de fin (`</ ... >`)
- Les balises sont des annotations, des méta-informations
- Une chaîne `< ... > ... </... >` est un **élément**
- La chaîne utilisée à l'intérieur des balises de début et de fin est le **nom de l'élément**
- Le **contenu de l'élément** est entre les balises de début et de fin (sans les inclure)


```
<!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" lang="fr-CA" >
  <head>
    <title>
      Mon premier document
    </title>
  </head>
  <body>
    <p>
      Hello world!
    </p>
  </body>
</html>
```



Éléments **body** et **head**

- Le contenu de l'élément **body** est affiché dans la zone de visualisation du client
- Le contenu de l'élément **head** contient des informations complémentaires
 - Un titre, scripts (côté client), styles



Éléments du langage

Balise (tag)
d'ouverture

Spécification d'un attribut

`<html xmlns="http://www.w3.org/1999/xhtml">`

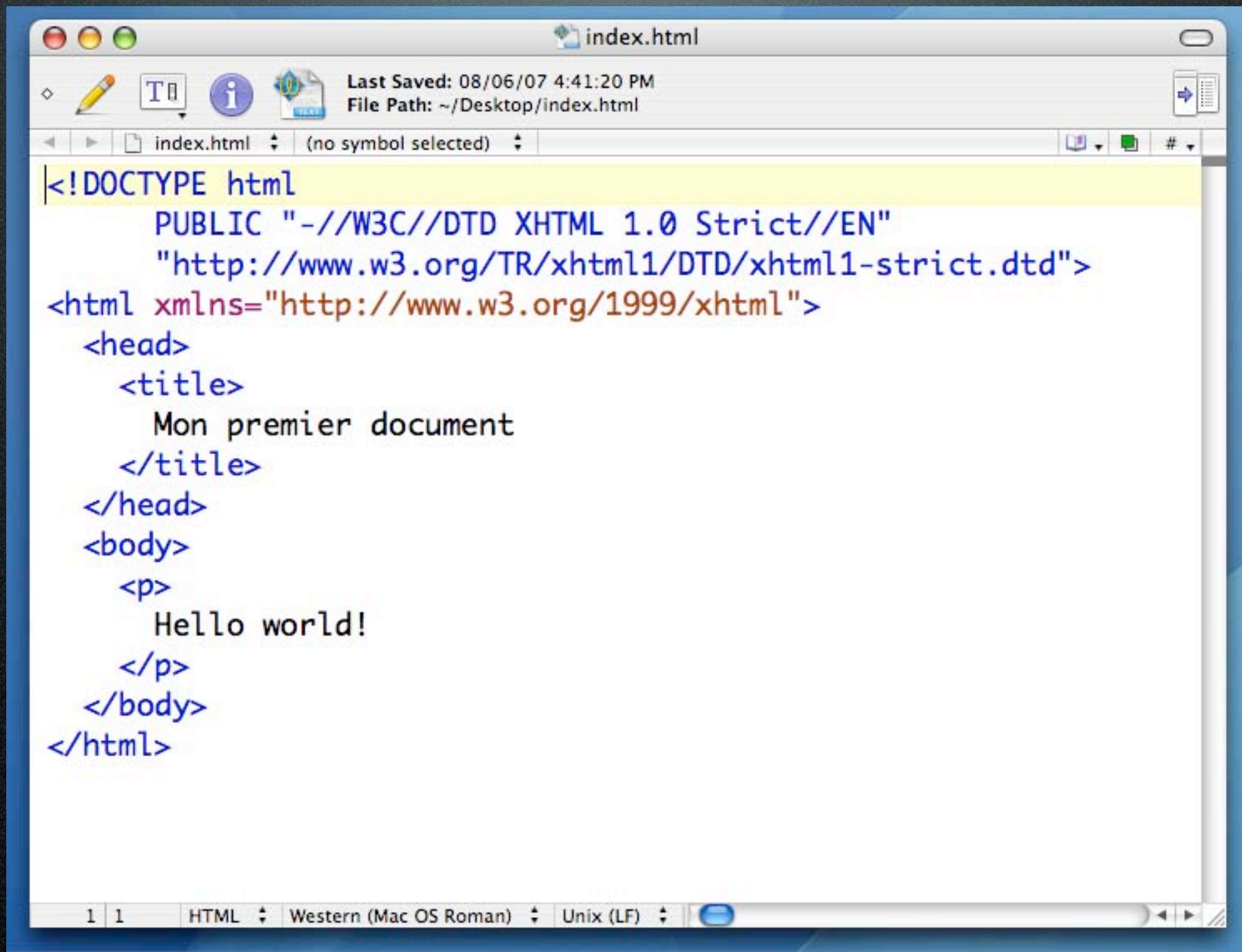
nom

valeur

`</html>`



```
<!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">
  <head>
    <title>Mon premier document</title>
  </head>
  <body>
    <p>
      Hello world!
    </p>
  </body>
</html>
```

The image shows a text editor window titled "index.html". The window's title bar includes standard Mac OS window controls (red, yellow, green buttons) and a close button. Below the title bar is a toolbar with icons for undo, redo, find, and help. The status bar at the top right indicates "Last Saved: 08/06/07 4:41:20 PM" and "File Path: ~/Desktop/index.html". The main editing area contains the following HTML code:

```
<!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">
  <head>
    <title>
      Mon premier document
    </title>
  </head>
  <body>
    <p>
      Hello world!
    </p>
  </body>
</html>
```

The status bar at the bottom of the editor shows "1 1" (line and column numbers), "HTML" (file encoding), "Western (Mac OS Roman)" (font), and "Unix (LF)" (line endings).


```
File Edit Search Markers Folding View Utilities Macros Plugins Help
index.html (/Users/turcotte/Desktop/)
<!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"> .
  <head> .
    <title>Mon premier document</title> .
  </head> .
  <body> .
    <p> .
      Hello world! .
    </p> .
  </body> .
</html> .
3,33-40 All      Shortcut for new marker? [no markers set]      (html,none,MacRoman)- - - - U 5/6Mb
```


Untitled1.xml [/Users/turcotte/Downloads/oxygen/Untitled1.xml] - <oXygen/> XML Editor

External Tools | Saxon-EE

XPath 2.0

Project

- sample.xpr
 - css
 - debugger
 - fo
 - import
 - jsp
 - nvdI
 - relaxng
 - schematron
 - svg
 - wSDL
 - xquery
 - dita
 - docbook
 - ooxml
 - tei

Outline

- html "http://www.w3.org/1999/xhtml"
 - head Hello World
 - body Contenu du document.
 - p Contenu du document.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE html SYSTEM "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml">
4   <head>
5     <title>Hello World</title>
6   </head>
7   <body>
8     <p>Contenu du document.</p>
9   </body>
10 </html>
11

```

Attributes

Attribute	Value
class	
dir	
id	
lang	
onclick	
ondblclick	
onkeydown	
onkeypress	
onkeyup	
onmousedown	
onmousemove	
onmouseout	

Transformation Scenarios

- XML transformation with XSLT
 - XHTML to DITA concept
 - XHTML to DITA reference
 - XHTML to DITA task
 - XHTML to DITA topic
- XML transformation with XQUERY
- DITA OT transformation
- XSLT transformation
- XProc transformation
- XQuery transformation
 - Execute XQuery
- SQL transformation

Type filter text

Elements | Entities | Trans...

Text | Grid | Author

/Users/turcotte/Downloads/oxygen/Untitled1.xml | U+000A | 8:36 | Modified

Petite Histoire d'HTML

- 1990, Tim Berners-Lee au CERN (Conseil Européen pour la Recherche Nucléaire) conçoit le langage HTML
- 1993-97, les années de guerre et d'instabilité
- 1997, W3C présente la version 4.0 de HTML (le standard auquel adhère la majorité des pages du Web)
- 1998, W3C propose XML
- 2000, W3C propose XHTML 1.0

Bio: Tim Burners-Lee

- 1955, né de parents mathématiciens
- 1976, diplômé de physique d'Oxford (Queen's College)
- 1989, invente le Web!
- 1994, fonde W3C (World Wide Web Consortium)
- 2004, on lui offre une chaire en informatique à l'Université de Southampton où il travaille sur le **web sémantique**



SGML, HTML, XML, XHTML, ...

- **SGML**

- Standard Generalized Markup Language
- Charles Goldfarb, Edward Mosher et Raymond Lorie (GML) chez IBM
- Fin des années 60s
- SGML, **ISO 8879:1986**

SGML

- SGML est un métalangage
- Dualité
 - La spécification d'un langage – DTD (document type definition)
 - Instance
- Application bien connue “New Oxford English Dictionary”

SGML vs HTML

- La syntaxe des versions 1.0 à 4.01 de HTML est définie à l'aide de SGML
- **SGML** est très **permissif**. Par exemple, certaines balises de fermeture peuvent être omises, c'est le cas de la balise **<p>**.
- Ce haut niveau de généralité rend l'analyse syntaxique plus complexe

SGML vs XML

- 1998, **W3C** propose **XML** (eXtensible Markup Language) un sous-ensemble de SGML afin de faciliter l'analyse de documents
- **XML** est donc aussi un métalangage (un langage de spécification)

<http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd>

...

```
<!ELEMENT html (head, body)>
```

```
<!ELEMENT head (%head.misc;
```

```
    ((title, %head.misc;, (base, %head.misc;)?) |
```

```
    (base, %head.misc;, (title, %head.misc;)))>
```

```
<!ELEMENT title (#PCDATA)>
```

```
<!ELEMENT base EMPTY>
```

```
<!ELEMENT meta EMPTY>
```

...

XHTML

- Tout simplement, un langage de balisage spécifié à l'aide de XML (plutôt que SGML)
- **XHTML 1.1** est l'équivalent de **HTML 4.01** avec les restrictions imposées par XML :
 - Aucune balise n'est omise
 - Éléments et attributs en minuscules
 - Guillemets autour des attributs
- XHTML 1.1 et HTML 4.01 sont les standards actuels

Avantages de XHTML

- Plusieurs outils disponibles pour l'analyse de documents XML
- Peu d'outils pour l'analyse de documents SGML

Les versions de XHTML

- **XHTML 1.0 Basic** pour les clients à capacités réduites (téléphones portables)
- **XHTML 1.1** : changements au niveau de la grammaire
- **XHTML 2** : en mars 2007, un groupe de travail a été formé afin de définir la prochaine spécification de XHTML

Éléments du langage

XHTML 1.1

XHTML 1.1 vs 1.0

- «On every element, the **lang** attribute has been removed in favor of the **xml:lang** attribute»
- «On the **a** and **map** elements, the **name** attribute has been removed in favor of the **id** attribute»

<http://www.w3.org/TR/xhtml11/changes.html>

XHTML 1.1 (Strict)

```
<?xml version="1.1" encoding="utf-8"?>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"  
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">  
  <head>  
    <title></title>  
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
  </head>  
  <body>  
  </body>  
</html>
```


XHTML 1.0

- 3 “saveurs” de DTD :

```
<!DOCTYPE html  
PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
<!DOCTYPE html  
PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

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


**Java™ Platform
Standard Ed. 6**[All Classes](#)

Packages

[java.applet](#)[java.awt](#)[java.awt.color](#)[java.awt.datatransfer](#)**All Classes**[AbstractAction](#)[AbstractAnnotationValu](#)[AbstractBorder](#)[AbstractButton](#)[AbstractCellEditor](#)[AbstractCollection](#)[AbstractColorChooserP](#)[AbstractDocument](#)[AbstractDocument.Attri](#)[AbstractDocument.Con](#)[AbstractDocument.Elem](#)[AbstractElementVisitor6](#)[AbstractExecutorService](#)[AbstractInterruptibleCha](#)[AbstractLayoutCache](#)[AbstractLayoutCache.N](#)[AbstractList](#)[AbstractListModel](#)[AbstractMap](#)[AbstractMap.SimpleEnt](#)[AbstractMap.SimpleImn](#)[AbstractMarshallerImpl](#)[AbstractMethodError](#)[AbstractOwnableSynchi](#)[AbstractPreferences](#)**Overview** [Package](#) [Class](#) [Use](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV](#) [NEXT](#)[FRAMES](#) [NO FRAMES](#)*Java™ Platform
Standard Ed. 6*

Java™ Platform, Standard Edition 6 API Specification

This document is the API specification for version 6 of the Java™ Platform, Standard Edition.

See:

[Description](#)

Packages

java.applet	Provides the classes necessary to create an applet and the classes an applet uses to communicate with its applet context.
java.awt	Contains all of the classes for creating user interfaces and for painting graphics and images.
java.awt.color	Provides classes for color spaces.
java.awt.datatransfer	Provides interfaces and classes for transferring data between and within applications.
java.awt.dnd	Drag and Drop is a direct manipulation gesture found in many Graphical User Interface systems that provides a mechanism to transfer information between two entities logically associated with presentation elements in the GUI.
java.awt.event	Provides interfaces and classes for dealing with different types of events fired by AWT components.
java.awt.font	Provides classes and interface relating to fonts.

Espace

- Les symboles suivants sont considérés comme des espaces : retour de chariot (13), saut de ligne (10), espace (32) et tabulation (9)

Espace : n.f. caractère électronique jouant le rôle d'une espace.

Espace : n.f. tige métallique insérée entre les caractères de plomb et servant à créer des espaces entre les mots à imprimer.

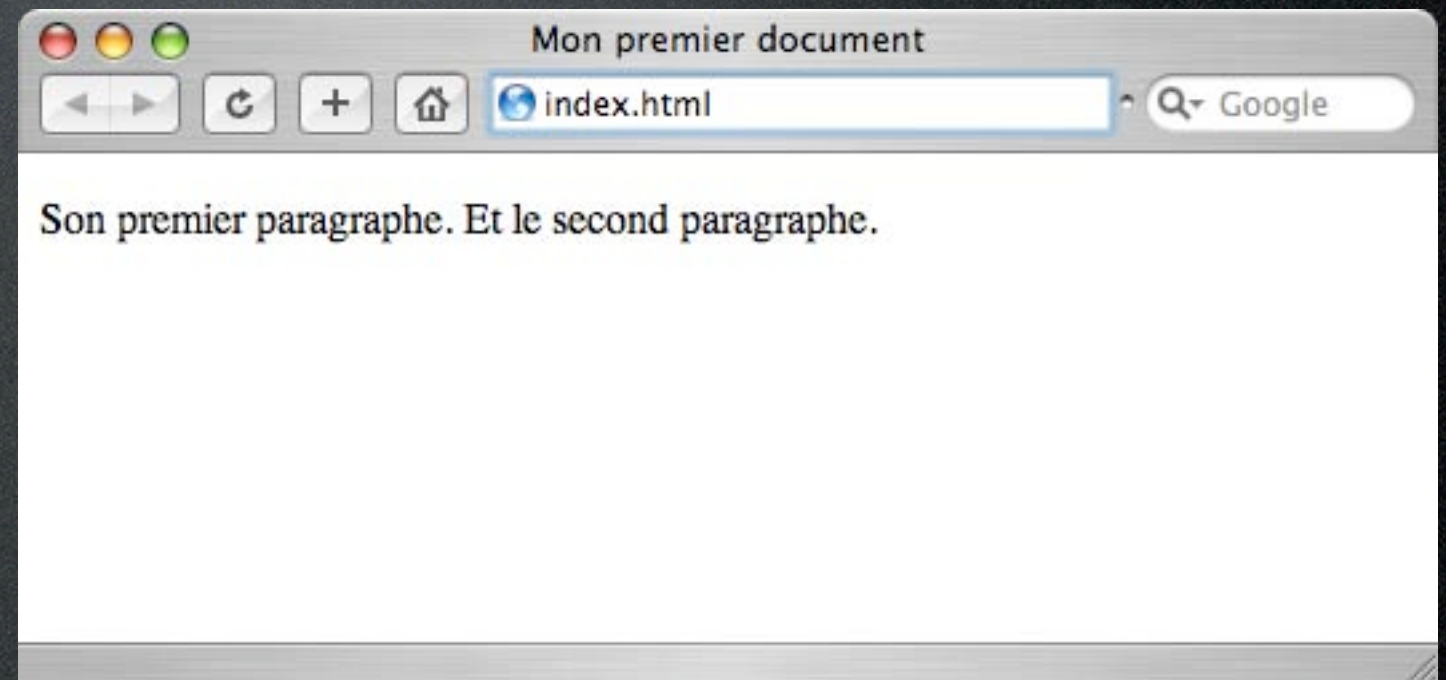
[Dictionnaire Antidote 2006]

Normalisation

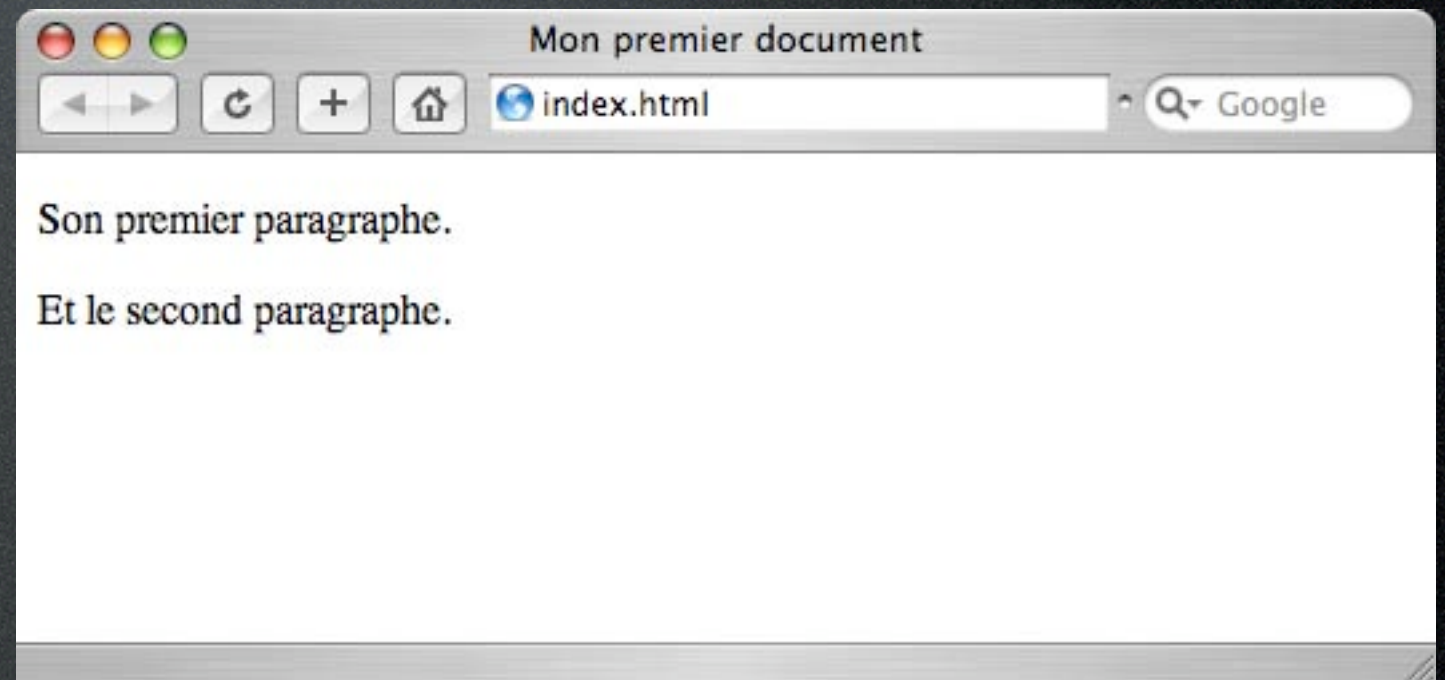
- Les symboles autres que l'espace (32) sont remplacés par l'espace
- Les espaces consécutives sont considérées comme une seule espace
- Les espaces au début et à la fin du contenu sont éliminées (trim)


```
<!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" lang="fr-CA">
  <head>
    <title>Mon premier document</title>
  </head>
  <body>
    <p>
      Son premier paragraphe.

      Et le second paragraphe.
    </p>
  </body>
</html>
```



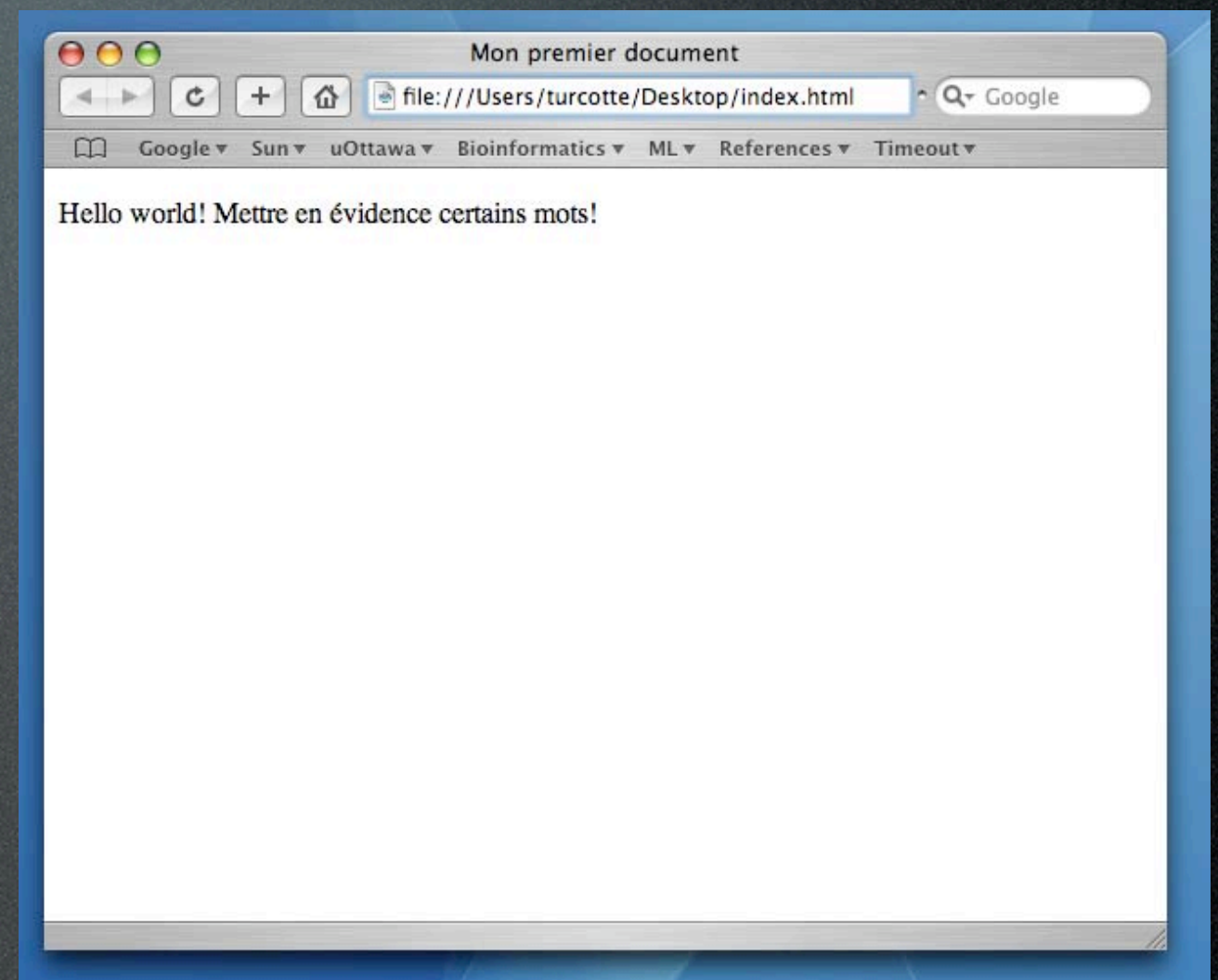

```
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr-CA">
  <head>
    <title>Mon premier document</title>
  </head>
  <body>
    <p>
      Son premier paragraphe.
    </p>
    <p>
      Et le second paragraphe.
    </p>
  </body>
</html>
```



Un fureteur, ça ne se plaint pas !

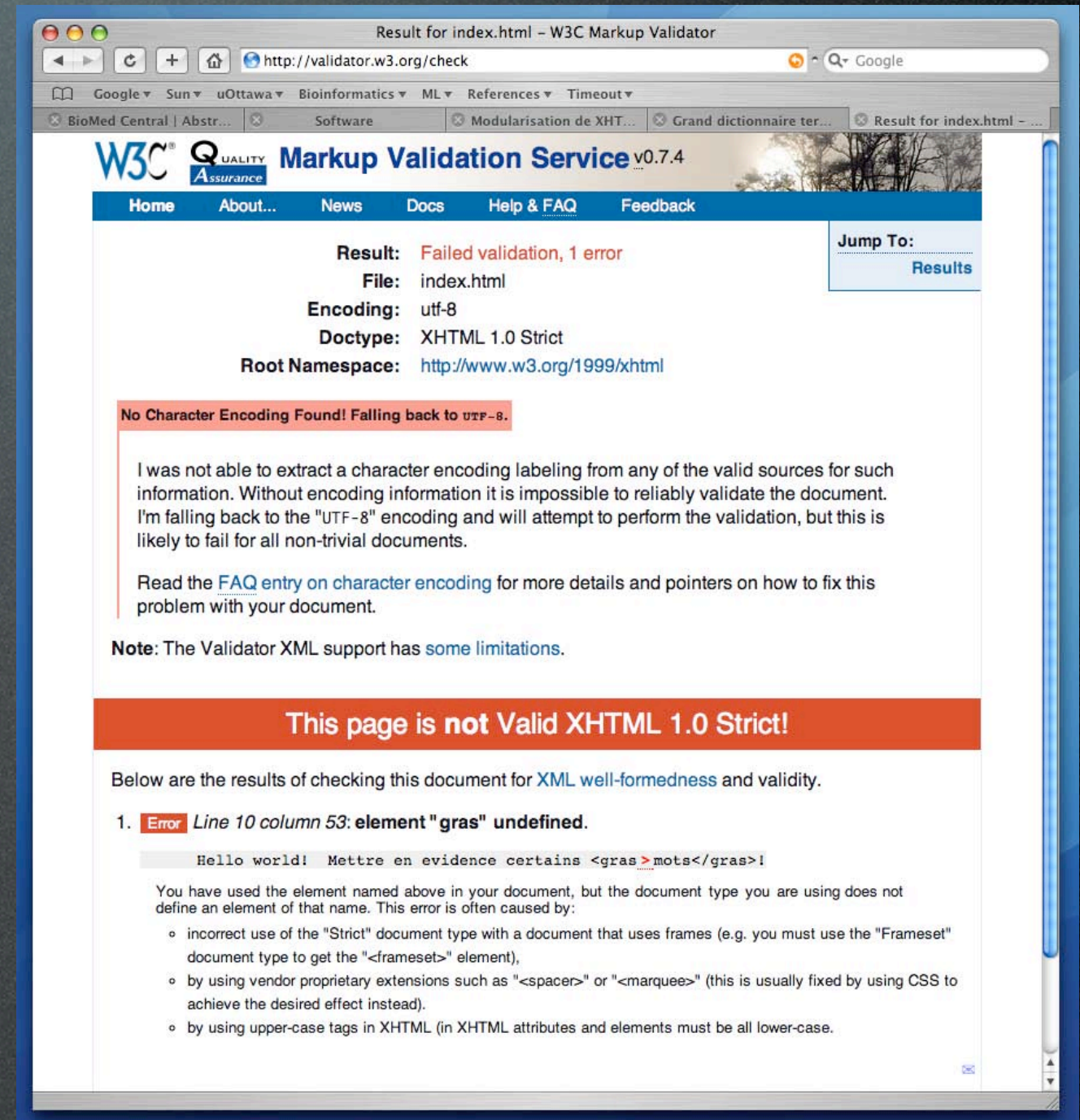
Hello world! Mettre en évidence certains **<gras>mots</gras>**!

- Les navigateurs ignorent les éléments inconnus ou erronés
- Facilite l'intégration de nouveaux éléments pour les générations successives de XHTML



Un fureteur, ça ne se plaint pas ! Par contre ...

- Il existe des outils afin de valider la syntaxe des documents
- validator.w3.org



The screenshot shows the W3C Markup Validation Service interface. The browser address bar displays the URL <http://validator.w3.org/check>. The page title is "Result for index.html - W3C Markup Validator". The main content area shows the following information:

- Result:** Failed validation, 1 error
- File:** index.html
- Encoding:** utf-8
- Doctype:** XHTML 1.0 Strict
- Root Namespace:** <http://www.w3.org/1999/xhtml>

A red box highlights the error message: "No Character Encoding Found! Falling back to UTF-8." Below this, a text block explains: "I was not able to extract a character encoding labeling from any of the valid sources for such information. Without encoding information it is impossible to reliably validate the document. I'm falling back to the 'UTF-8' encoding and will attempt to perform the validation, but this is likely to fail for all non-trivial documents." A link to the FAQ entry on character encoding is provided.

A note states: "Note: The Validator XML support has some limitations."

A prominent orange banner reads: "This page is not Valid XHTML 1.0 Strict!"

Below the banner, the text says: "Below are the results of checking this document for XML well-formedness and validity."

The first error is listed as: "1. Error Line 10 column 53: element 'gras' undefined." The error message is followed by a code snippet: "Hello world! Mettre en evidence certains <gras>mots</gras>!".

The explanation states: "You have used the element named above in your document, but the document type you are using does not define an element of that name. This error is often caused by:"

- incorrect use of the "Strict" document type with a document that uses frames (e.g. you must use the "Frameset" document type to get the "<frameset>" element),
- by using vendor proprietary extensions such as "<spacer>" or "<marquee>" (this is usually fixed by using CSS to achieve the desired effect instead).
- by using upper-case tags in XHTML (in XHTML attributes and elements must be all lower-case).

Caractères spéciaux

- Certains caractères ont un usage **réservé**, c'est le cas du <
- On souhaite parfois produire un caractère qui n'existe pas dans l'encodage utilisé pour sauvegarder le document
 - Comment affiché un “é” alors que le document est sauvegardé en **US-ASCII**?

Caractères spéciaux

- On utilise une **appel de caractère** ou **appel d'entité**
- Un **appel d'entité** débute avec le symbole “&” et se terminent avec le symbole “;”
 - < signifie “<”
 - & signifie “&”
 - © signifie ©
 - é signifie “é”
- &mnemonic; = appel d'entité

Caractères spéciaux

- On peut aussi utiliser les unicodes
 - < signifie “<”
 - & signifie “&”
 - é signifie “é”
- &#Unicode; = appel de caractère

Caractères spéciaux

- Un appel d'entité particulièrement utile pour les documents en langue française
 - ; représente une espace ou l'on ne peut appliquer une césure (**nonbreaking space**)
- “En voici deux : bleu et rouge”
- Ces derniers ne sont pas affectés par le processus de normalisation, et sont donc utilisés pour créer des espaces multiples


```

<!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" lang="fr-CA">
  <head>
    <title>Structures, techniques et normes du Web (CSI 3540)</title>
  </head>
  <body>
    <h1>CSI 3540. Structures, techniques et normes du Web</h1>
    <p>
      Infrastructure de base du Web. Serveurs et navigateurs. Exemples
      de protocoles. Internet et virus. Architecture de moteur de
      recherche. Contenu et présentation Web. Pages Web, leur
      structure et leur interprétation. HTML, XML et leurs
      dérives. Interfaces Web vers les logiciels et bases de
      données. Télémoins et droit à la vie privée.
      Web sémantique et ontologies. Services Web. <strong>(3,1.5,1.5) 3 cr</strong>.
    </p>
  </body>
</html>

```



```
<!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" lang="fr-CA">
  <head>
    <title>Structures, techniques et normes du Web (CSI 3540)</title>
    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
  </head>
  <body>
    <h1>CSI 3540. Structures, techniques et normes du Web</h1>
    <p>
      Infrastructure de base du Web. Serveurs et navigateurs. Exemples
      de protocoles. Internet et virus. Architecture de moteur de
      recherche. Contenu et présentation Web. Pages Web, le
      structure et leur interprétation. HTML, XML et leurs
      dérivés. Interfaces Web vers les logiciels et bases de
      données. Témoins et droit à la vie privé. Web sémantique
      ontologies. Services Web. <strong>(3,1.5,1.5) 3 cr</strong>
    </p>
  </body>
</html>
```

http-equiv signifie
attributs qui auraient pu être passés
dans l'entête du message http

charset = utf-8 (unicode)
charset = ISO-8859-1
windows

Tutoriel <http://www.w3.org/International/tutorials/tutorial-char-enc/>

Attributs

```

```

- Les éléments du langage XHTML (ou tout simplement HTML) possèdent des attributs
- Il y a au moins une espace entre le nom de la balise et le premier attribut
- Il peut y avoir plusieurs attributs

Attributs

```

```

- Le format des attributs est **étiquette = "valeur"**
(guillemets simples ou doubles obligatoires)
- Peuvent contenir des guillemets simples ou double ou des appels ("c'est okay", 'ça "aussi', "c'est okay", 'c'est pas okay')
- Peuvent subir une normalisation par le navigateur (vaut mieux le faire soit même)

Attributes (suite)

`<p id="sec:intro">...</p>`, `<table id="tbl:results">...</table>`

- Pour chaque élément, il peut y voir un attribut **ID**
- L'attribut **ID** doit être unique
- Cet attribut deviendra particulièrement intéressant lors des discussions sur **CSS** et **JavaScript**

Titres de section

- Six degrés de titrage : `h1`, `h2`, `h3`, `h4`, `h5`, `h6`
- Comportent un élément de **sémantique** :
 - `<h1>` Titre du premier niveau `</h1>`
 - `<h2>` Un sous-titre `</h2>`
 - `<h3>` Un sous-sous-titre `</h3>`
 - **Les feuilles de style (CSS) nous permettront de changer l'apparence des titres**



Titre de premier niveau

Un sous-titre

Un sous-sous-titre

Texte préformaté : élément **PRE**

```
<pre>
Reader r = new Reader( program );
while ( r.hasMoreTokens() ) {
    Token t = r.nextToken();

    if (! t.isSymbol()) {
        // ...
    } else if ( t.sValue().equals( ... ) ) {
        // ...
    }
}
</pre>
```



Texte préformaté : élément **PRE**

- Les balises faisant partie du contenu d'un élément **PRE** seront interprétées comme des balises

Balise fermante interdite

- Certains éléments ne peuvent avoir un contenu
- Pour ces éléments la balise fermante est interdite
- Le symbole «/» précède le symbole «>» de la balise ouvrante de l'élément
- L'élément BR, qui sert à forcer la coupure de ligne, est un exemple : `
`

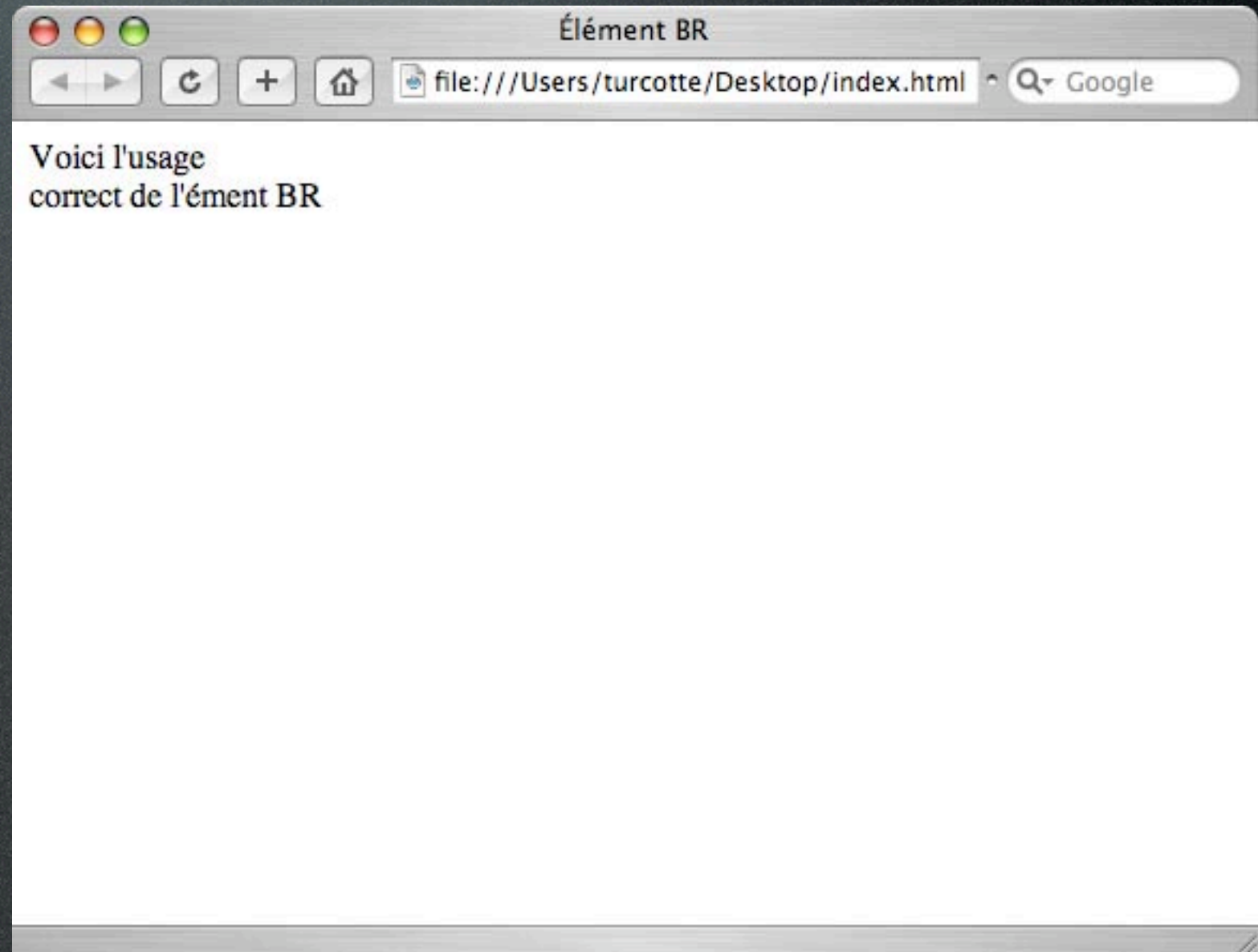
Balise fermante interdite

`
`

Ce segment n'est pas valide (la balise fermante est interdite)

`</br>`

Voici l'usage `
`
correct de l'élément BR



Formattage des phrases

- Éléments prédéfinis : `gras`, `gras`, `italic`, `<i>italic</i>`, `<tt>teletype</tt>`, `<big>gros</big>`, `<small>petit</small>`
- L'élément SPAN est un «wrapper» qu'on utilise afin de modifier les attributs du texte
 - `ceci est un exemple`

Images :

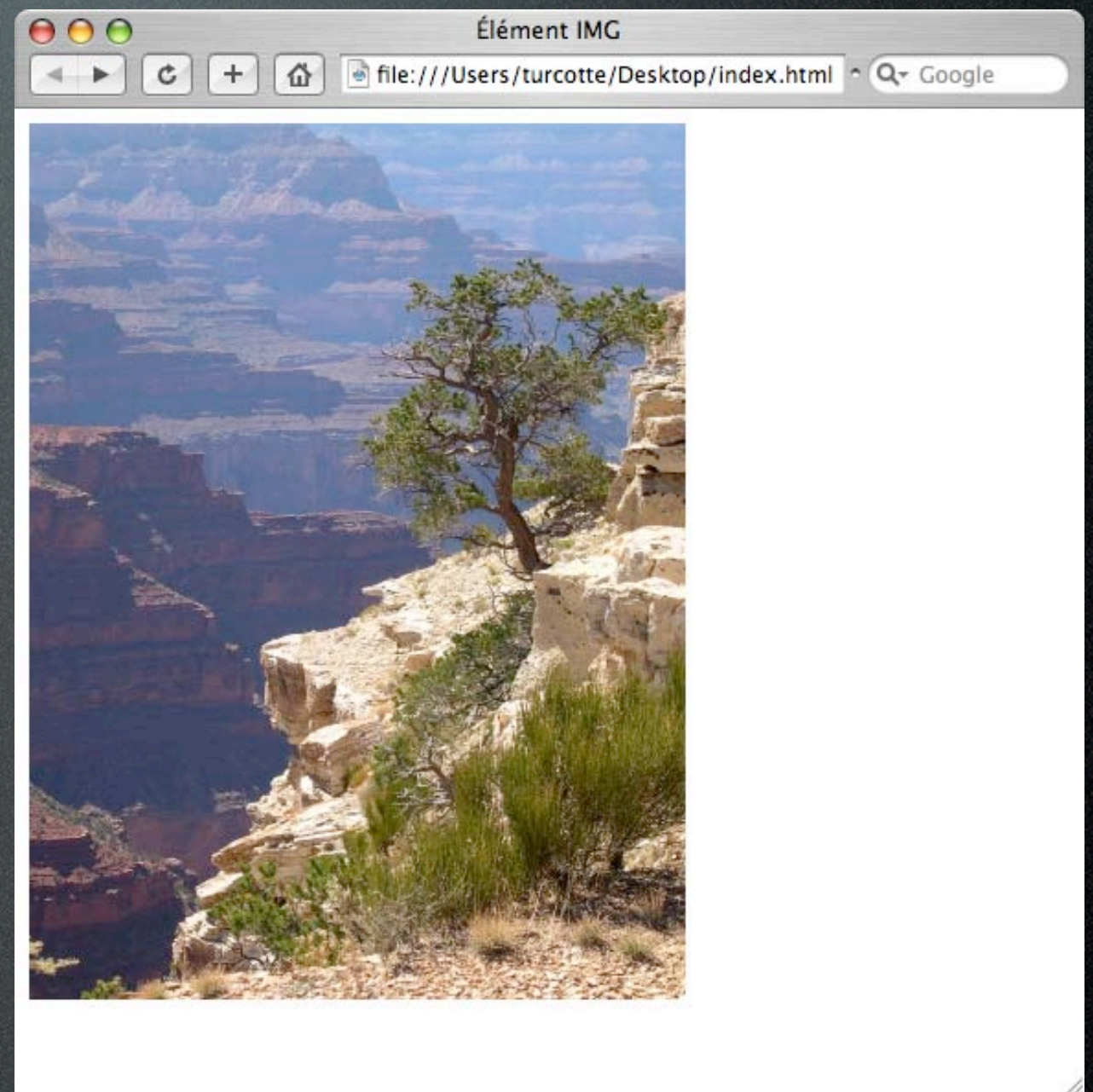
```

```




```
<!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" lang="fr-CA">
  <head>
    <title>Élément IMG</title>
  </head>
  <body>
    <p>
      
    </p>
  </body>
</html>
```

src (l'URL où trouver
l'image) et alt (une
description de l'image)
sont requis



Images

1. Requête HTTP **Get** afin d'obtenir le document HTML
2. Le fureteur analyse le document et découvre les éléments **IMG**
3. **Pour chaque élément IMG**, le fureteur exécute **une nouvelle requête HTTP Get** afin d'obtenir l'image graphique associée

- ``
- Bien que les dimensions soient optionnelles, on devrait toujours les inclure, afin que le fureteur puisse déterminer l'aspect visuel de la page
 - Sinon, il se peut que le fureteur attende d'avoir obtenu toutes images afin de dessiner la page
 - Le fureteur utilisera des dimensions par défaut et redessiner la page, pour chaque image obtenue

- ``
- La valeur des attributs width et height peut être un pourcentage de la de la surface affichable du navigateur

URL : absolue ou relative

- Soit le document **index.html** sauvegarder à l'URL suivante :
- **http://localhost:8080/exemples/d2013a/index.html**
- Imaginez que le répertoire **d2013a** contienne le fichier **arizona-2006.jpg**
- À l'intérieur du document **index.html**, les deux définitions suivantes sont équivalentes :
 - ``
 - ``

La racine sur l'hôte est
domains/domain-1/docroot
pour GlassFish

URL : absolue ou relative

- Similairement :
 - `images/arizona-2006.jpg`
 - réfère au fichier `arizona-2006.jpg` du sous-répertoire `images` du répertoire `d2013a`
- ou encore :
 - `../images/arizona-2006.jpg`
 - équivaut à l'URL absolue suivante :
 - `http://localhost:8080/exemples/images/arizona-2006.jpg`

URL : absolue ou relative

- Finalement :
 - </images/arizona-2006.jpg>
 - équivaut à l'URL absolue suivante :
 - <http://localhost:8080/images/arizona-2006.jpg>

URL : absolue ou relative

- Les URLs débutant par un descripteur (http, ...) sont absolues
- Quels sont les avantages des URLs relatives?
 1. Déplacer des sous-arbres entiers avec un minimum d'efforts
 2. Des copies des mêmes pages peuvent servir par plusieurs serveurs

Hyperliens : élément **A**

- `Visitez Google`
- L'attribut **HREF** de l'élément **A** sert à spécifier l'URL de l'hyperlien
- Suite au clique de l'utilisateur, le navigateur génère une nouvelle requête **Get**, et remplace le contenu du navigateur par le nouveau document
- Au prochain laboratoire, nous verrons qu'un élément **A** peut aussi servir de point d'ancrage

```
<a href="http://www.google.ca" /></a>
```


Commentaires

```
<!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" lang="fr-CA">
  <head>
    <title>Élément IMG</title>
  </head>
  <body>
    <p>
```

<!-- L'élément img doit être placé dans un "bloc", ici p **-->**

```
  
  </p>
</body>
</html>
```


Imbrication

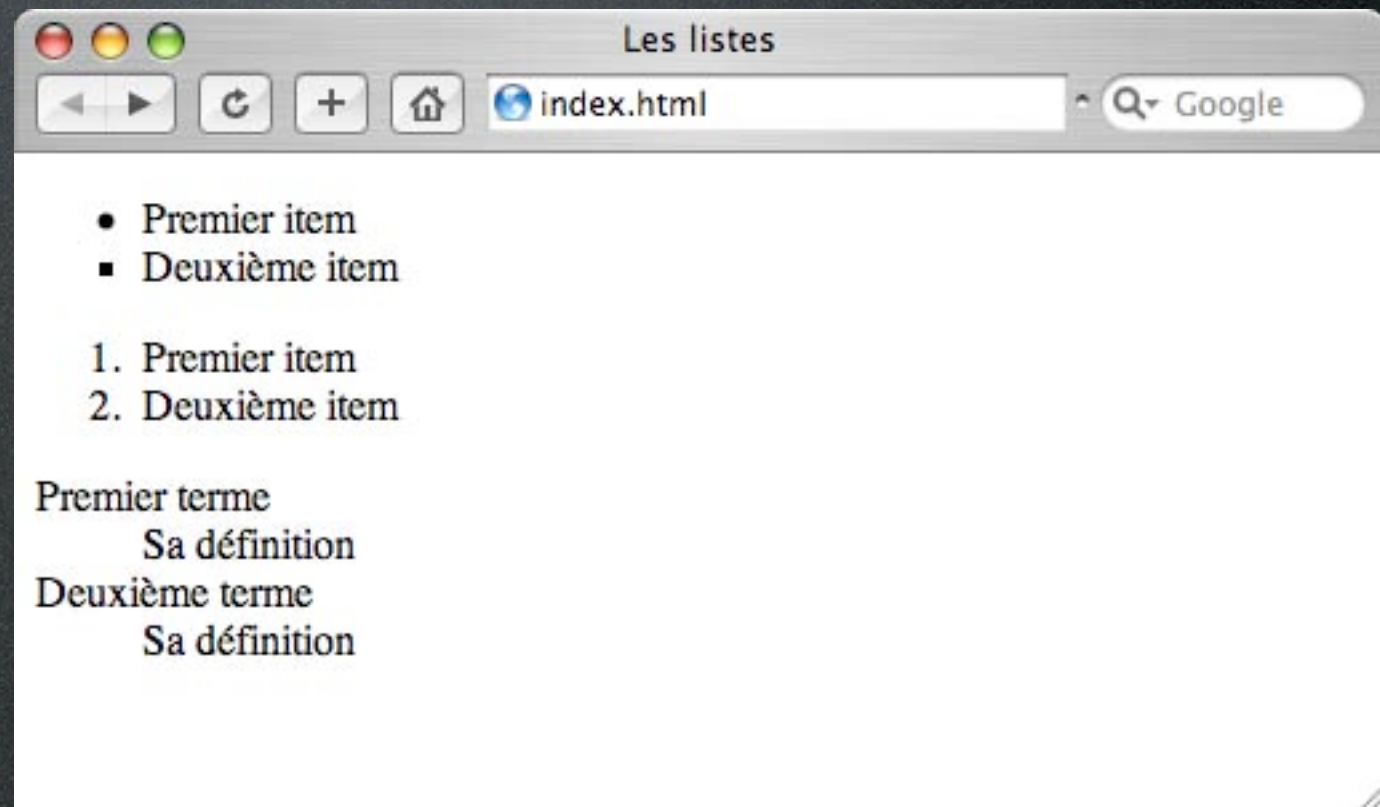
- Les éléments peuvent être imbriqués
 - L'élément imbriqué fait partie du contenu de l'élément parent
 - `aucun<i>enchevêtrementpermis</i>`

2 catégories d'éléments

- **Bloc** (P, DIV, ...) : en général, ces éléments débutent sur une nouvelle ligne. Les éléments imbriqués sont des blocs ou des éléments en ligne.
- **En ligne** (EM, SPAN, ...) : ne force pas une césure. Les éléments imbriqués sont des éléments en ligne.
- Les enfants (immédiats) de l'élément **body** devraient être des éléments de type bloc

3 types de listes

- Non-ordonnées
- Ordonnées
- Définitions



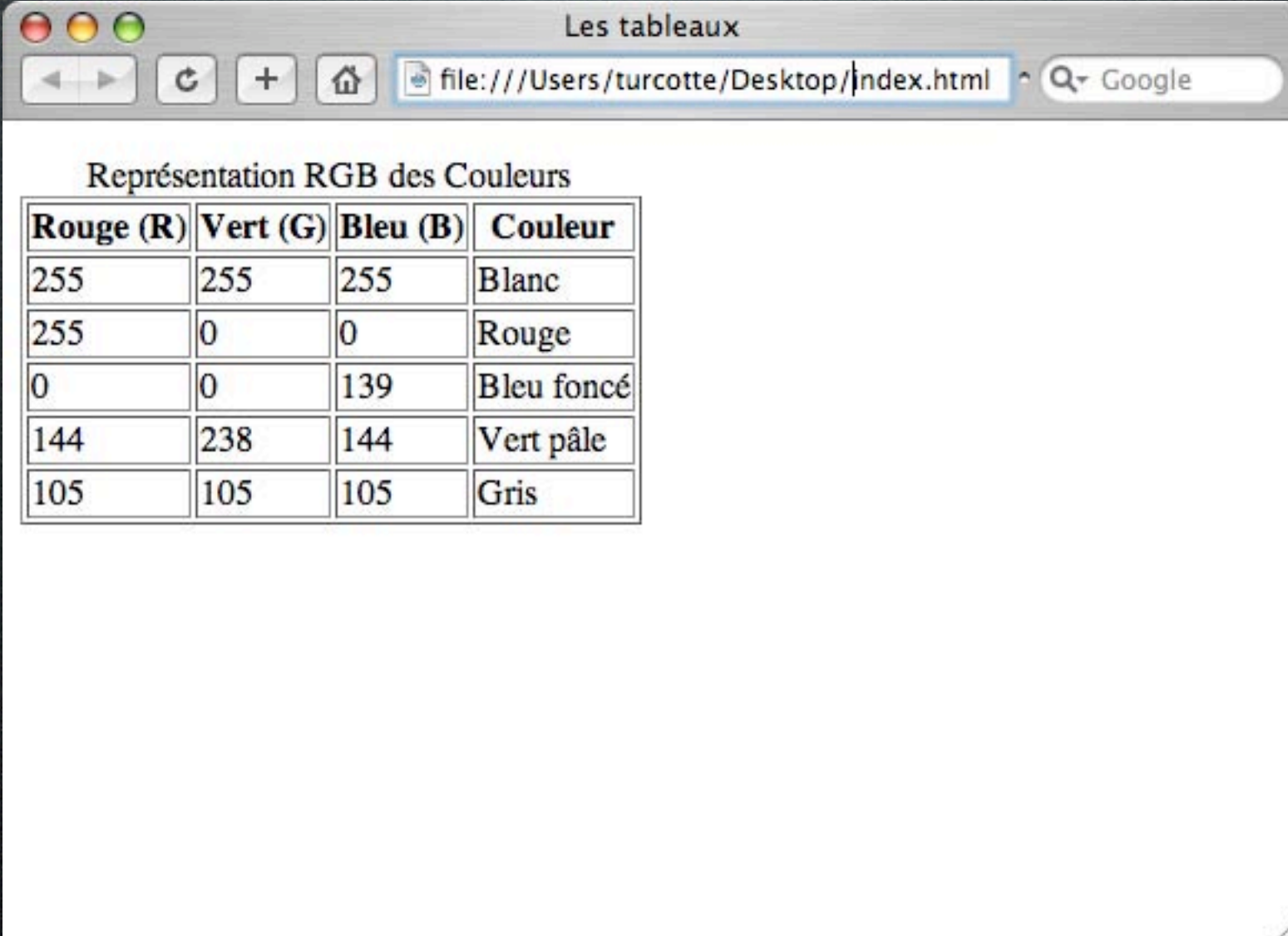

```
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr-CA">
  <head>
    <title>Les listes</title>
  </head>
  <body>
    <p>
      <ul>
        <li>Premier item</li>
        <li type="square">Deuxième item</li>
      </ul>
      <ol>
        <li>Premier item</li>
        <li>Deuxième item</li>
      </ol>
      <dl>
        <dt>Premier terme</dt>
        <dd>Sa définition</dd>
        <dt>Deuxième terme</dt>
        <dd>Sa définition</dd>
      </dl>
    </p>
  </body>
</html>
```


Listes imbriquées



```
<ul>  
  <li>Bulleted list item  
    <ul>  
      <li>Nested list item</li>  
      <li>Nested list item 2</li>  
    </ul>  
  </li>  
  <li>Bulleted list item 2</li>  
</ul>
```


Les tableaux



The screenshot shows a web browser window with the title "Les tableaux". The address bar contains the file path "file:///Users/turcotte/Desktop/index.html" and a search bar with the text "Google". The main content of the page is a table titled "Représentation RGB des Couleurs". The table has four columns: "Rouge (R)", "Vert (G)", "Bleu (B)", and "Couleur". The rows represent different colors with their corresponding RGB values.

Rouge (R)	Vert (G)	Bleu (B)	Couleur
255	255	255	Blanc
255	0	0	Rouge
0	0	139	Bleu foncé
144	238	144	Vert pâle
105	105	105	Gris


```

<table border="1">
  <caption>
    Représentation RGB des Couleurs
  </caption>
  <tr>
    <th>Rouge (R)</th><th>Vert (G)</th><th>Bleu (B)</th><th>Couleur</th>
  </tr>
  <tr>
    <td>255</td><td>255</td><td>255</td><td>Blanc</td>
  </tr>
  <tr>
    <td>255</td><td>0</td><td>0</td><td>Rouge</td>
  </tr>
  <tr>
    <td>0</td><td>0</td><td>139</td><td>Bleu foncé</td>
  </tr>
  <tr>
    <td>144</td><td>238</td><td>144</td><td>Vert pâle</td>
  </tr>
  <tr>
    <td>105</td><td>105</td><td>105</td><td>Gris</td>
  </tr>
</table>

```


Les éléments associés aux tableaux

- Ces éléments sont : `<table>`, `<caption>`, `<tr>` (table row), `<td>` (table data) et `<th>` (table header)
- Ces éléments ont une multitude d'attributs, dont plusieurs contrôlent les l'aspect visuel des tableaux

Tableaux : quelques attributs

Les tableaux

file:///Users/turcotte/Desktop/index.html Google

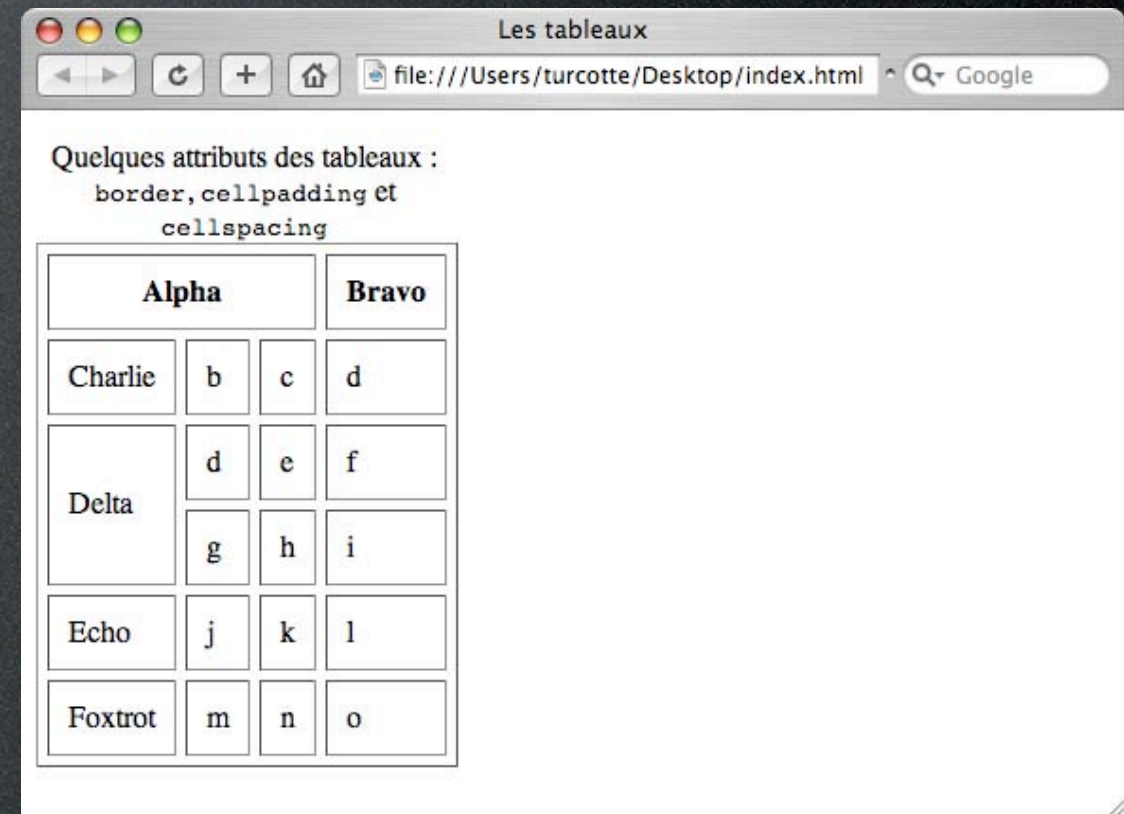
Quelques attributs des tableaux :
border, cellpadding et
cellspacing

Alpha		Bravo	
Charlie	b	c	d
Delta	d	e	f
	g	h	i
Echo	j	k	l
Foxtrot	m	n	o


```

<table border="1" cellpadding="10" cellspacing="5">
  <caption>
    Quelques attributs des tableaux :
    <tt>border</tt>, <tt>cellpadding</tt> et <tt>cellspacing</tt>
  </caption>
  <tr>
    <th colspan="3">Alpha</th><th>Bravo</th>
  </tr>
  <tr>
    <td>Charlie</td><td>b</td><td>c</td><td>d</td>
  </tr>
  <tr>
    <td rowspan="2">Delta</td><td>d</td><td>e</td><td>f</td>
  </tr>
  <tr>
    <td>g</td><td>h</td><td>i</td>
  </tr>
  <tr>
    <td>Echo</td><td>j</td><td>k</td><td>l</td>
  </tr>
  <tr>
    <td>Foxtrot</td><td>m</td><td>n</td><td>o</td>
  </tr>
</table>

```



Utilisations des tableaux

file:///Users/turcotte/Desktop/index.html Google

`border = 1, cellpadding = 10 Et cellspacing = 10`

Utilisations des tableaux

file:///Users/turcotte/Desktop/index.html Google

`border = 1, cellpadding = 10 Et cellspacing = 0`

Utilisations des tableaux

file:///Users/turcotte/Desktop/index.html Google

`border = 1, cellpadding = 0 Et cellspacing = 10`

Utilisations des tableaux

file:///Users/turcotte/Desktop/index.html Google

```
border = 0, cellpadding = 0 & cellspacing = 0
```



A photograph of a white Kenworth dump truck with a chrome grille and bumper, parked in front of a white building. The truck is viewed from a front-quarter angle. The image is displayed within a browser window titled "Utilisations des tableaux". Above the image, the code `border = 0, cellpadding = 0 & cellspacing = 0` is shown in a monospaced font. The browser window includes standard navigation buttons (back, forward, refresh, home) and a search bar with the Google logo.


```

<table border="0" cellpadding="0" cellspacing="0">
  <caption>
    <tt>border = 0</tt>, <tt>cellpadding = 0</tt> et <tt>cellspacing = 0</tt>
  </caption>
  <tr>
    <td></td>
    <td></td>
  </tr>
  <tr>
    <td></td>
    <td></td>
  </tr>
</table>

```

Par défaut, l'élément IMG est de type «en ligne», le navigateur insère donc un espace sous la ligne pour les caractères tels que p, q et g

Les cadres

Élement FRAMESET

**Java™ Platform
Standard Ed. 6**[All Classes](#)

Packages

[java.applet](#)[java.awt](#)[java.awt.color](#)[java.awt.datatransfer](#)**All Classes**[AbstractAction](#)[AbstractAnnotationValu](#)[AbstractBorder](#)[AbstractButton](#)[AbstractCellEditor](#)[AbstractCollection](#)[AbstractColorChooserP](#)[AbstractDocument](#)[AbstractDocument.Attri](#)[AbstractDocument.Con](#)[AbstractDocument.Elem](#)[AbstractElementVisitor6](#)[AbstractExecutorService](#)[AbstractInterruptedException](#)[AbstractLayoutCache](#)[AbstractLayoutCache.N](#)[AbstractList](#)[AbstractListModel](#)[AbstractMap](#)[AbstractMap.SimpleEnt](#)[AbstractMap.SimpleImn](#)[AbstractMarshallerImpl](#)[AbstractMethodError](#)[AbstractOwnableSynchi](#)[AbstractPreferences](#)**Overview** [Package](#) [Class](#) [Use](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV](#) [NEXT](#)[FRAMES](#) [NO FRAMES](#)*Java™ Platform
Standard Ed. 6*

Java™ Platform, Standard Edition 6 API Specification

This document is the API specification for version 6 of the Java™ Platform, Standard Edition.

See:

[Description](#)

Packages

java.applet	Provides the classes necessary to create an applet and the classes an applet uses to communicate with its applet context.
java.awt	Contains all of the classes for creating user interfaces and for painting graphics and images.
java.awt.color	Provides classes for color spaces.
java.awt.datatransfer	Provides interfaces and classes for transferring data between and within applications.
java.awt.dnd	Drag and Drop is a direct manipulation gesture found in many Graphical User Interface systems that provides a mechanism to transfer information between two entities logically associated with presentation elements in the GUI.
java.awt.event	Provides interfaces and classes for dealing with different types of events fired by AWT components.
java.awt.font	Provides classes and interface relating to fonts.

PackageListFrame

PackageFrame

ClassFrame

**Java™ Platform
Standard Ed. 6**[All Classes](#)

Packages

[java.applet](#)[java.awt](#)[java.awt.color](#)[java.awt.datatransfer](#)**All Classes**[AbstractAction](#)[AbstractAnnotationValu](#)[AbstractBorder](#)[AbstractButton](#)[AbstractCellEditor](#)[AbstractCollection](#)[AbstractColorChooserP](#)[AbstractDocument](#)[AbstractDocument.Attri](#)[AbstractDocument.Con](#)[AbstractDocument.Elem](#)[AbstractElementVisitor6](#)[AbstractExecutorService](#)[AbstractInterruptibleCha](#)[AbstractLayoutCache](#)[AbstractLayoutCache.N](#)[AbstractList](#)[AbstractListModel](#)[AbstractMap](#)[AbstractMap.SimpleEnt](#)[AbstractMap.SimpleImn](#)[AbstractMarshallerImpl](#)[AbstractMethodError](#)[AbstractOwnableSynchi](#)[AbstractPreferences](#)**Overview** [Package](#) [Class](#) [Use](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV](#) [NEXT](#)[FRAMES](#) [NO FRAMES](#)*Java™ Platform
Standard Ed. 6*

Java™ Platform, Standard Edition 6 API Specification

This document is the API specification for version 6 of the Java™ Platform, Standard Edition.

See:

[Description](#)

Packages

java.applet	Provides the classes necessary to create an applet and the classes an applet uses to communicate with its applet context.
java.awt	Contains all of the classes for creating user interfaces and for painting graphics and images.
java.awt.color	Provides classes for color spaces.
java.awt.datatransfer	Provides interfaces and classes for transferring data between and within applications.
java.awt.dnd	Drag and Drop is a direct manipulation gesture found in many Graphical User Interface systems that provides a mechanism to transfer information between two entities logically associated with presentation elements in the GUI.
java.awt.event	Provides interfaces and classes for dealing with different types of events fired by AWT components.
java.awt.font	Provides classes and interface relating to fonts.


```
<!DOCTYPE html
PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
<html>
  <head>
    <title>Java SE 6</title>
  </head>
  <frameset cols="20%,80%">
    <frameset rows="3*,5*">
      <frame src="overview-frame.html" name="PackageListFrame" title="Tous les packages">
      <frame src="allclasses-frame.html" name="PackageFrame" title="Toutes les classes">
    </frameset>
    <frame src="overview-summary.html" name="ClassFrame" title="Description" scrolling="yes">
  </frameset>
  <noframes>
    <h1>C't'un vieux fureteur que t'as là!</h1>
    Voici <a href="sans-cadres.html">la version sans cadres de ce document</a>.
  </noframes>
</frameset>
</html>
```


PackageListFrame

PackageFrame

ClassFrame

overview-frame.html

```
<!DOCTYPE html
  PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
  "http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Liste des packages (Java SE 6)</title>
  </head>
  <body>
    <big><strong>Java SE 6</strong></big><br/>
    <strong>Liste des packages</strong><br/>
    <ul>
      <li><a href="applet.html" target="PackageFrame">java.applet</a></li>
      <li><a href="awt.html" target="PackageFrame">java.awt</a></li>
    </ul>
  </body>
</html>
```


PackageListFrame

PackageFrame

ClassFrame

allclasses-frame.html

```
<!DOCTYPE html
  PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
  "http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Toutes les classes (Java SE 6)</title>
  </head>
  <body>
    <big><strong>Toutes les classes</strong></big><br/>
    <ul>
      <li><a href="abstractaction.html" target="ClassFrame">AbstractAction</a></li>
      <li><a href="visitor.html" target="ClassFrame">ValueVisitor</a></li>
    </ul>
  </body>
</html>
```


overview-summary.html

```
<!DOCTYPE html
  PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
  "http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Liste des packages (Java SE 6)</title>
  </head>
  <body>
    <center><big><strong>Java SE 6</strong></big><br/></center>
    <strong>Liste des packages</strong><br/>
    <ul>
      <li><a href="applet.html">java.applet</a></li>
      <li><a href="awt.html">java.awt</a></li>
    </ul>
  </body>
</html>
```

Notez l'absence d'attribut target

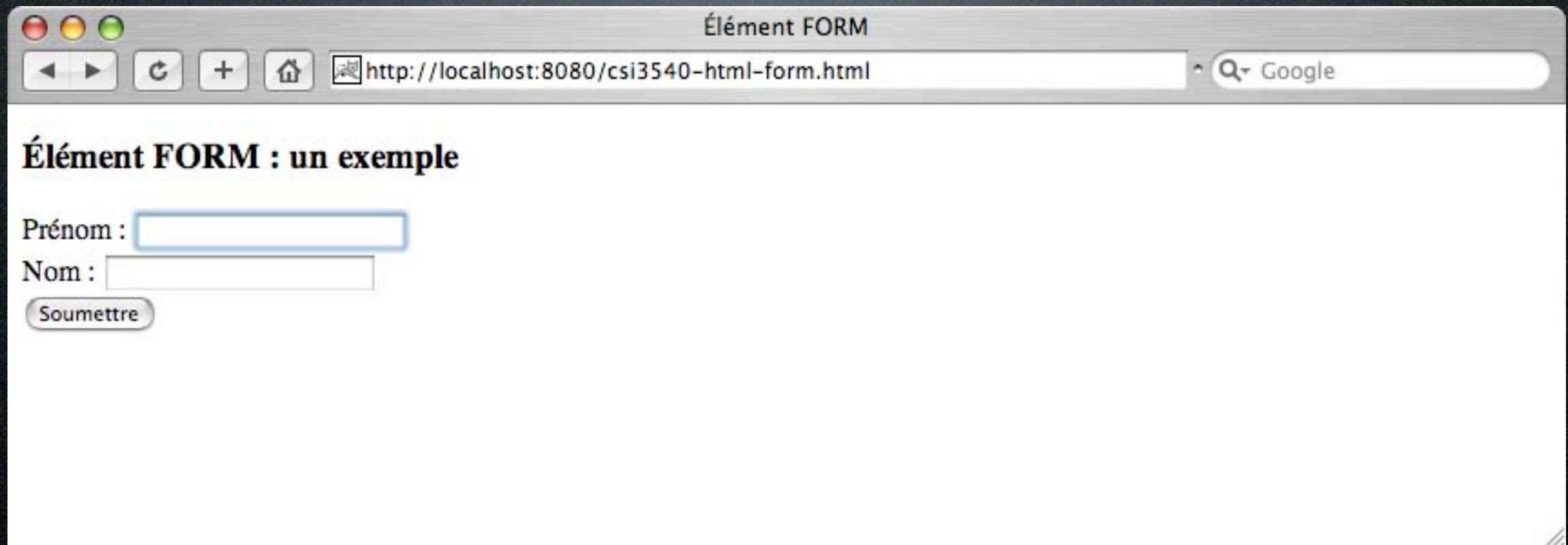
Les noms de cible suivants sont réservés et ont des significations particulières :

- **_blank** : L'agent utilisateur devrait charger le document désigné dans une nouvelle fenêtre sans nom.
- **_self** : L'agent utilisateur devrait charger le document dans le même cadre que l'élément qui désigne cette cible.
- **_parent** : L'agent utilisateur devrait charger le document dans l'élément FRAMESET parent immédiat du cadre courant. Cette valeur est équivalente à «_self» si le cadre courant n'a pas de parent.
- **_top** : L'agent utilisateur devrait charger le document dans la fenêtre originale entière (de ce fait, annulant tous les autres cadres). Cette valeur est équivalente à «_self» si le cadre courant n'a pas de parent.

Problèmes associés aux cadres (framesets)

- Confusion chez les novices
 - Surtout du côté de l'impression des documents
- Problèmes d'affichage sur les petits écrans (téléphone portable, par exemple)
- Souvent mal programmé
 - Nombre anormal de copies imbriquées, ou encore, le cadre du site «A» demeure lorsqu'on suit un lien vers le site «B»

Élément FORM



Élément FORM

http://localhost:8080/csi3540-html-form.html

Google

Élément FORM : un exemple

Prénom :

Nom :

Élément FORM



Request Parameters Example

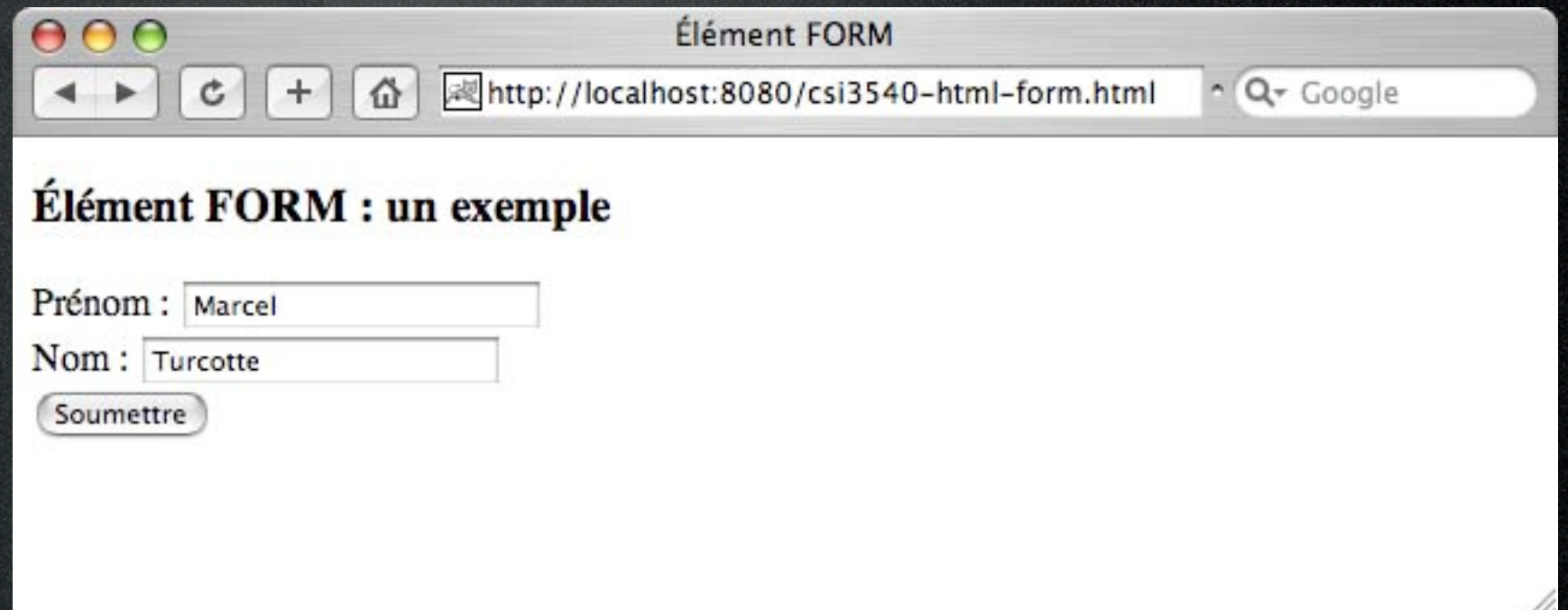
Parameters in this request:
First Name: = Marcel
Last Name: = Turcotte

First Name:

Last Name:

<http://localhost:8080/csi3540-html-form.html>

```
<!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" lang="fr-CA">
  <head>
    <title>Élément FORM</title>
  </head>
  <body>
    <h1>Élément FORM : un exemple</h1>
    <p>
      <form action="http://localhost:8080/examples/servlets/servlet/RequestParamExample" method="post">
        <div>
          <label>Prénom : <input type="text" size="20" name="firstname" /></label><br />
          <label>Nom : <input type="text" size="20" name="lastname" /></label> <br />
          <input type="submit" value="Soumettre" />
        </div>
      </form>
    </p>
  </body>
</html>
```



The screenshot shows a web browser window titled "Élément FORM". The address bar contains the URL "http://localhost:8080/csi3540-html-form.html". The page content displays the rendered form with the following elements:

- Prénom :
- Nom :
-

Élément FORM

- L'attribut **method** de l'élément FORM prend soit la valeur **"get"** ou la valeur **"post"** (discussion à suivre)

Élément INPUT

- L'attribut **type** de l'élément INPUT peut prendre l'une des valeurs suivantes :
text, password, checkbox, radio, submit, reset, file, hidden, image ou **button**
- Valeur par défaut
 - `<input type="text" size="20" name="firstname" value="Marie ou Joseph" />`
- La valeur de l'attribut **name** servira au programme associé au formulaire
- En plus de l'élément INPUT, il y a aussi les éléments TEXTAREA et SELECT

L'attribut SUBMIT de l'élément INPUT

1. Lorsque l'utilisateur appuie sur le bouton submit (associé à l'élément INPUT dont la valeur de l'attribut TYPE est SUBMIT)
2. Le navigateur produit une requête HTTP Post dont le corps contient la liste des paramètres et valeurs du formulaire
3. Qu'il soumet au serveur
(les interactions client/serveur seront traitées en détail au cours du semestre)

Élément FORM



Request Parameters Example

Parameters in this request:
First Name: = Marcel
Last Name: = Turcotte

First Name:

Last Name:


```
⌘ telnet localhost 8080
```

```
Trying ::1...
```

```
Connected to localhost.
```

```
Escape character is '^['.
```

```
POST /examples/servlets/servlet/RequestParamExample HTTP/1.1
```

```
Host: localhost:8080
```

```
User-Agent: Mozilla/5.0
```

```
Content-Length: 34
```

```
Content-Type: application/x-www-form-urlencoded
```

```
firstname=Marcel&lastname=Turcotte
```

```
HTTP/1.1 200 OK
```

```
Server: Apache-Coyote/1.1
```

```
Content-Type: text/html
```

```
Content-Length: 684
```

```
Date: Thu, 14 Jun 2007 21:18:46 GMT
```

```
...
```



```
...
<html>
<body>
<head>
<title>Request Parameters Example</title>
</head>
<body bgcolor="white">
<a href="../reqparams.html">
</a>
<a href="../index.html">
</a>
<h3>Request Parameters Example</h3>
Parameters in this request:<br>
First Name:
= marcel<br>
Last Name:
= turcotte
<p>
<form action="RequestParamExample" method=POST>
First Name:
<input type=text size=20 name=firstname>
<br>
Last Name:
<input type=text size=20 name=lastname>
<br>
<input type=submit>
</form>
</body>
</html>
```


Association explicite et implicite des étiquettes

```
FORM action="..." method="post">
  <P>
  <LABEL for="label_prenom">Prénom : </LABEL>
    <INPUT type="text" id="label_prenom"><BR>
  <LABEL for="label_nom">Nom : </LABEL>
    <INPUT type="text" id="label_nom"><BR>
  <LABEL for="label_email">E-mail : </LABEL>
    <INPUT type="text" id="label_email"><BR>
  <INPUT type="radio" name="genre" value="homme"> Homme<BR>
  <INPUT type="radio" name="genre" value="femme"> Femme<BR>
  <INPUT type="submit" value="Envoyer"> <INPUT type="reset">
  </P>
</FORM>
```

LABEL sert à associer une étiquette à une commande (élément INPUT)
Cette étiquette peut-être lue par un synthétiseur de la parole

Association explicite et implicite des étiquettes

```
<FORM action="..." method="post">
<P>
<LABEL>
  Prénom;nom
  <INPUT type="text" name="prenom">
</LABEL>
<LABEL>
  <INPUT type="text" name="nom">
  Nom
</LABEL>
</P>
</FORM>
```

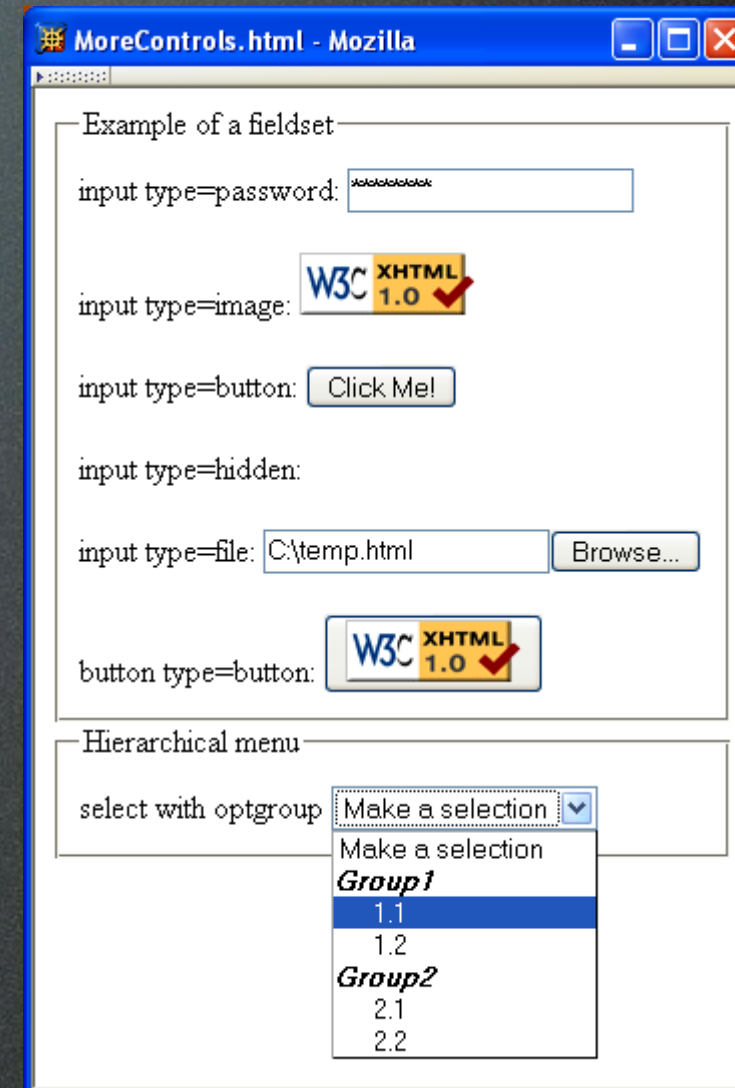

Association explicite et implicite des étiquettes

```
<form action="..." method="post">
<table>
  <tr>
    <td><label for="label_prenom">Prénom</label></td>
    <td><input type="text" name="prenom" id="label_prenom"></td>
  </tr>
  <tr>
    <td><label for="label_nom">Nom</label></td>
    <td><input type="text" name="nom" id="label_nom"></td>
  </tr>
</table>
</form>
```


Autres types d'éléments INPUT

TABLE 2.5: HTML 4.01/XHTML 1.0 non-deprecated form controls.

Element	type Attribute	Control
input	text	Text input
input	password	Password input
input	checkbox	Checkbox
input	radio	Radio button
input	submit	Submit button
input	image	Graphical submit button
input	reset	Reset button (form clear)
input	button	Push button (for use with scripts)
input	hidden	Non-displayed control (stores server-supplied information)
input	file	File select
button	submit	Submit button with content (not an empty element)
button	reset	Cancel button with content (not an empty element)
button	button	Button with content but no predefined action
select	N/A	Menu
option	N/A	Menu item
optgroup	N/A	Heading in a hierarchical menu
textarea	N/A	Multi-line text input
label	N/A	Associate label with control(s)
fieldset	N/A	Groups controls
legend	N/A	Add caption to a fieldset



HTML5

- 2007: Groupe de travail WHATWG du W3C
- Réduire (éliminer) l'utilisation d'environnements d'exécution tels qu'Adobe Flash, Microsoft Silverlight, Sun JavaFX
- <http://www.w3.org/TR/html5/>
(25 août 2009)
- <http://www.whatwg.org/specs/web-apps/current-work/multipage/>
(19 janvier 2010)

HTML5

- Nouveaux éléments : NAV, HEADER, FOOTER, SECTION, ARTICLE, TIME
- Certains éléments sont éliminés : FONT, CENTER, STRIKE, TT, BIG, U, etc.
- Plusieurs attributs éliminés : BGCOLOR, BORDER, ALIGN, SIZE, etc.
- Meilleure séparation contenu/présentation (CSS)

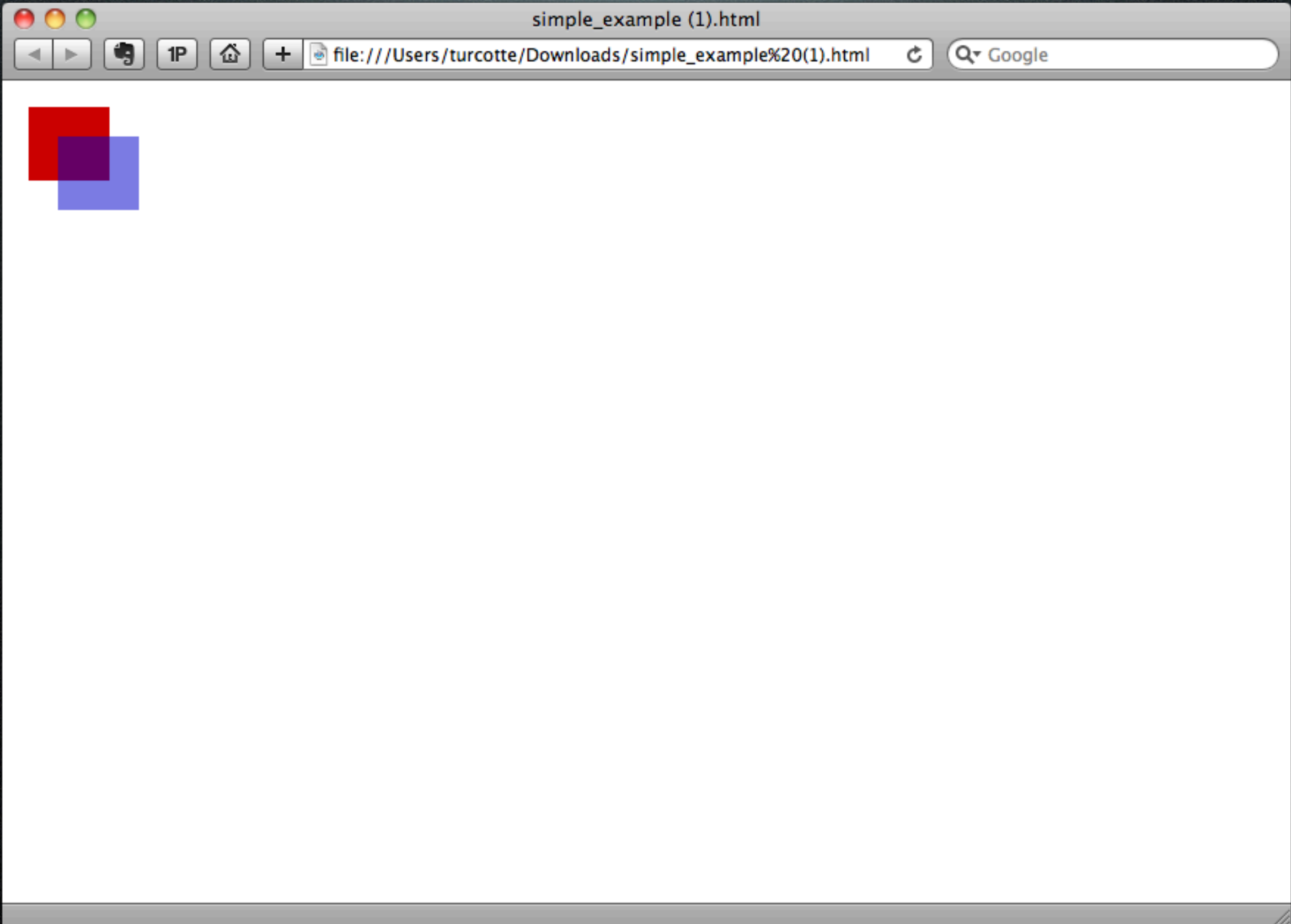
HTML5

- Nouveaux éléments : VIDEO, AUDIO, CANVAS (bitmap)
- Nouvelles interfaces de programmation (API) : vidéo, audio, canvas


```
<html>
<head>
<script type="application/x-javascript">
function draw() {
  var canvas = document.getElementById("canvas");
  if (canvas.getContext) {
    var ctx = canvas.getContext("2d");

    ctx.fillStyle = "rgb(200,0,0)";
    ctx.fillRect (10, 10, 55, 50);

    ctx.fillStyle = "rgba(0, 0, 200, 0.5)";
    ctx.fillRect (30, 30, 55, 50);
  }
}
</script>
</head>
<body onload="draw();">
  <canvas id="canvas" width="150" height="150"></canvas>
</body>
</html>
```


```
<video src=sample.mov autoplay></video>
```

```
<script>
```

```
function playPause() {
```

```
  var myVideo = document.getElementsByTagName('video')[0];
```

```
  if (myVideo.paused)
```

```
    myVideo.play();
```

```
  else
```

```
    myVideo.pause();
```

```
}
```

```
</script>
```

```
<input type=button onclick="playPause()" value="Play/Pause">
```


HTML5

- Nouveaux éléments pour favoriser l'accessibilité
- Ajout du concept de cache pour les applications hors ligne (« offline »)
- Glisser-déposer (« drag-and-drop »)
- Envoi de messages entre documents
- Bac à sable (« sandbox »)
- MathML et SVG

HTML5

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>SVG in text/html</title>
  </head>
  <body>
    <p>
      A green circle:
      <svg> <circle r="50" cx="50" cy="50" fill="green"/> </svg>
    </p>
  </body>
</html>
```


HTML5

- Possède une syntaxe HTML et une syntaxe XML
- Ressemble plus à HTML 4.0 qu'à XHTML 2.0
- http://wiki.whatwg.org/wiki/Implementations_in_Web_browsers

Conclusion et remarques

- Éviter l'utilisation d'éléments de présentation (nous utiliserons CSS pour cela)
- Favoriser l'utilisation d'éléments à caractère sémantique

Resources

- Liste des traductions françaises disponibles des documents W3C [<http://www.w3.org/2005/11/Translations/Lists/ListLang-fr.html>] 2007
- **La spécification HTML 4.01** [<http://www.la-grange.net/w3c/html4.01/cover.html>] 2007
- Modularisation de XHTML - Termes et définitions [<http://www.la-grange.net/w3c/xhtml-modularization/terms.html>] 2007

Resources (suite)

- Uniform Resource Identifiers (URI) :
Generic Syntax [<ftp://ftp.rfc-editor.org/in-notes/std/std-66.txt>]
2007