

Programmez avec les langages XML / DTD et XHTML / CSS

Eric SULTAN

0212v3

Contenu du cours

- Introduction
- XML
 - Qu'est-ce que le standard XML ?
 - La structure XML
 - Définition d'un langage XML avec une DTD
- XHTML
 - les bases du XHTML
 - les balises du XHTML
 - Créer une feuille de style avec CSS

Programmez avec les langages XML / DTD et XHTML /
CSS

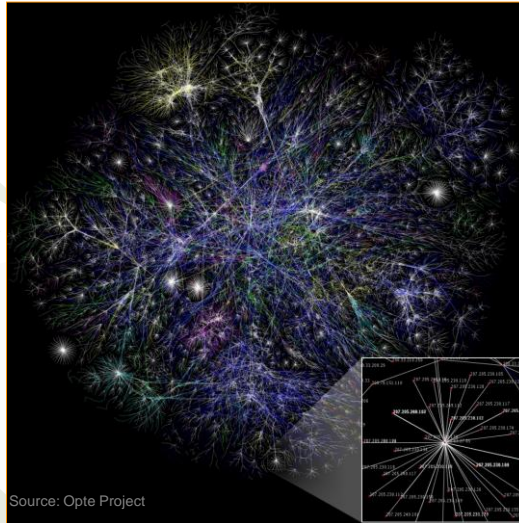
AVANT CELA... QU'EST-CE QU'INTERNET ET LE WEB ?

Internet

- **Internet** est un réseau informatique mondiale utilisant la suite de protocole TCP/IP
- Une connexion sur Internet est identifié par une **adresse IP** (Internet Protocol)
- Sur Internet, on retrouve tous types d'échanges d'informations sous différents protocoles, dans différents réseaux:
Internet est "le réseau des réseaux"
- Le réseau le plus important contenu au sien d'Internet est le **World Wide Web**, **géré par le** World Wide Web Consortium (**W3C**)

Internet

Carte de 30% des adresses IP sur Internet en 2005



Le Web

- Le Web est un réseau de pages web (fichiers **hypertextes**)
- Chaque document web à pour adresse web une **URL** (Uniform Ressource Locator)
- Un document hypertexte est un document contenant des **hyperliens** (des liens vers d'autres documents)
- Les pages web sont essentiellement transférés via le protocole **HTTP** (HyperText Transfert Protocol)

Code source d'une page web

Programmez avec les langages XML / DTD et XHTML / CSS

5

Introduction

- Le HTML est un langage à balises basé sur le standard SGML (Standard Generalized Markup Language)
- Le **XHTML** est le même langage que le HTML, mais celui-ci est basé sur le standard **XML** (eXtensible Markup Language)
- Les standards XML et SGML sont des normes établissant comment doit être défini un langage à balises
- Le standard XML est plus restreint mais aussi plus “convenable” que le SGML

Introduction

- En résumé:
 - Le HTML et le XHTML sont le même langage, seul la norme d’utilisation du langage change
 - le XML est une norme permettant de définir un langage à balises
 - le XHTML est **une application XML (ou « langage XML »)**, c’est-à-dire un langage à balises défini suivant le standard XML.
- Apprenons donc d’abord ce qu’est le XML, puis apprenons à créer des pages web avec le XHTML

Programmez avec les langages XML / DTD et XHTML /
CSS

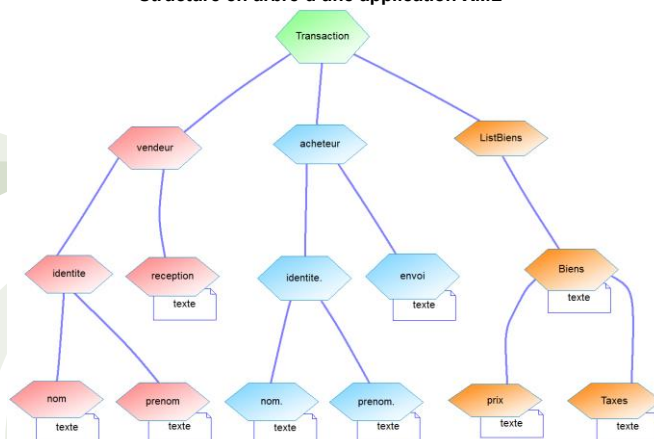
XML: QU'EST-CE QUE LE STANDARD XML ?

Qu'est-ce que le standard XML ?

- Le standard XML est une norme définissant des langages à balises
- Un langage à balises permet de **structurer** des informations sous la forme d'un **arbre**
- **Définir un langage XML** revient à **définir un jeu de balise**

Qu'est-ce que le standard XML ?

Structure en arbre d'une application XML



Qu'est-ce que le standard XML ?

- Un document XML a toujours une seule **racine** (root): une balise « racine » encadre toute les autres
- La définition d'un langage XML (de sa grammaire, son jeu de balises) se fait dans un fichier **DTD (Document Type Definition)**
(il existe d'autres types: XSD, Relax NG...)
- Les langages XML les plus connus sont le XHTML, le SVG, le RSS et bien d'autres

Qu'est-ce que le standard XML ?

Code d'un fichier XML ayant pour langage « hello-page »

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE page SYSTEM "hello-page.dtd">
<page>
  <title>Mon titre</title>
  <content>
    entre ces 2 balises, je met mon contenu !
  </content>
  <comment>
    et pour celles-ci, je met mes commentaires.
  </comment>
</page>

<!--
Ici, je suis dans un bloc de commentaire de code.
Je suis du texte qui ne sera jamais analysé !
-->
```

Qu'est-ce que le standard XML ?

- Les programmes destinés à **analyser des documents XML** sont des **parseurs XML**:
les navigateurs web ont un parseur XML destiné à lire du XHTML
- Il faut toujours lier les documents XML à ses définitions de jeux de balises (fichiers DTD)

Programmez avec les langages XML / DTD et XHTML /
CSS

XML: LA STRUCTURE XML

La structure XML

- Les caractères propres au XML sont:
 - **<** : caractère de début de balise
 - **>** : caractère de fin de balise
 - **"** : caractère (début et fin) de valeur d'attribut
 - **'** : caractère (début et fin) de valeur d'attribut
 - **&** : caractère d'échappement (permettant l'utilisation de ces caractères **dans le contenu**)
- Utilisez **<**, **&**, **>**, **"**, **'** à la place de: **<**, **&**, **>**, **"**, **'**
 - C'est également valable pour les URLs

La structure XML

- Une balise (**tag**) **ouvrante** et une balise **fermante** encadrent **un contenu texte** (et/ou d'autres balises) et définissent le type d'**élément** (**nom de la balise**)
- Des informations liées à une balise (autres que le contenu) peuvent être définies dans des **attributs**
- Les attributs sont définis à **l'intérieur de la balise ouvrante**

```
<produit categorie="fruit et légumes">
  banane</produit>
```

La structure XML

- Balise ouvrante

```
<element>
```

- Balise ouvrante avec des attributs

```
<element attribut1="valeur d'attribut1" attribut2="valeur
d'attribut2">
```

- Balise fermante (toujours sous cette forme !)

```
</element>
```

La structure XML

- La notion « **well-formed** » est un document XML **correctement formé**
- Le document commence par **une déclaration XML** (l'attribut **version** est obligatoire)

- Un **encodage** peut être ajouté en attribut (d'après la norme, le défaut est **utf-8**):

La structure XML

- begin-tags (balises d'ouverture) et end-tags (balises de fermeture) **doivent correspondre**, un peu comme des poupées russes.
- **PAS DE CROISEMENT** de type:
`<i>......</i> `
- Balise de type "EMPTY" (balises sans balises de fermeture):
 Ces balises sans contenu utilisent la syntaxe XML **auto-fermante** (cf HTML: `
`)
 Ils aussi peuvent avoir **des attributs**.

La structure XML

- Un document **valide** doit être:
 - "well-formed"
 - être associé à une DTD (ou un autre type de définition)
 - être conforme à cette DTD (ou un autre type de définition)
- Un document XML doit le plus souvent être valide

La structure XML

- Pour associer un document à une DTD, on utilise la balise **Doctype**
 - Syntaxe:

```
<!Doctype RACINE TYPE "URL">
```

- **RACINE**: le nom de la balise racine
- **TYPE**:
 - **PUBLIC**: La DTD est officiellement publiée (DTD connue: XHTML, RSS, SVG...)
 - **SYSTEM**: La DTD est sur votre système ou internet (pour les DTD « faites maison »)
- **URL**: adresse de la DTD (adresse **relative** ou **absolu**)

La structure XML

- Exemple

```
<!Doctype page SYSTEM "hello-page.dtd">
```

```
<!Doctype page SYSTEM "http://www.foo.bar/hello-page.dtd">
```

- les déclaration de DTD **PUBLIC** ajoutent souvent une information sur **la version**

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

```
<!DOCTYPE rss PUBLIC  
"-//Netscape Communications//DTD RSS 0.91//EN"  
"http://my.netscape.com/publish/formats/rss-0.91.dtd">
```

Programmez avec les langages XML / DTD et XHTML /
CSS

**XML:
LA DÉFINITION
D'UN LANGAGE XML
AVEC UNE DTD**

La définition d'un langage XML avec une DTD

Une DTD est une grammaire qui définit:

1. les balises (tags) possibles et leurs attributs
2. L'imbrication des balises à l'intérieur d'autres balises
3. Quels balises et attributs sont à option et lesquels sont obligatoires

- Exemple de DTD

```
<!ELEMENT name (family,given)>
<!ELEMENT family (#PCDATA)>
<!ELEMENT given (#PCDATA)>
```

- l'élément "name" a deux éléments enfants: **family** et **given**
- Les éléments "family" et "given" ne contiennent que **du texte (pas d'autres éléments)**

La définition d'un langage XML avec une DTD

- Déclaration d'un élément

- On doit définir *chaque élément* que l'on pense utiliser dans les documents
- On doit indiquer comment les éléments s'imbriquent
- Chaque élément ne se définit qu'une seule fois !!

- Syntaxe

```
<!ELEMENT nom_balise spécification_contenu>
```

La spécification du contenu d'un élément contient:

- soit une combinaison d'autres éléments,
- soit une combinaison d'autres élément plus des éléments spéciaux #PCDATA, ANY (contenu mixte)
- soit l'élément #PCDATA, ou ANY
- soit EMPTY

La définition d'un langage XML avec une DTD

Règles de combinaison: On peut combiner selon les règles ci-dessous:

A et B = tags	Explication specification_contenu	Exemples
A?	A (un seul) est une option, (donc: A ou rien)	<code><!ELEMENT person (name,email?)></code>
A+	Il faut un ou plusieurs A	<code><!ELEMENT person (name,email+)></code>
A*	A est une option, il faut zéro, un ou plusieurs A	<code><!ELEMENT person (name,email*)></code>
A B	Il faut A ou B, mais pas les deux	<code><!ELEMENT person (email fax)></code>
A , B	Il faut A, suivi de B (dans l'ordre)	<code><!ELEMENT person (name ,email)></code>
(A, B) +	Les parenthèses regroupent. Ici: un ou plusieurs (A suivi de B)	<code><!ELEMENT liste (name,email)+></code>

La définition d'un langage XML avec une DTD

Éléments spéciaux

Élément spéciaux	Explication specification_contenu	Exemples
#PCDATA	"Parsed Character Data" Données (non-interprétées par XML) dans le langage d'encodage courant	<code><!ELEMENT email (#PCDATA)></code>
ANY	Mot clé qui indique que tous les éléments sont autorisés (déconseillé)	<code><!ELEMENT person ANY></code>
EMPTY	Balise auto-fermante comme dans 	<code><!ELEMENT br EMPTY></code>

- **Restriction sur les identificateurs (noms)**
 - chaque identificateur doit commencer par une lettre ou '_'
 - ensuite lettres, chiffres, '_', '-', '.', ':', '!',

La définition d'un langage XML avec une DTD

- Exemple pour un Address Book

```
<!ELEMENT addressBook (person)+>
<!ELEMENT person (name,email*)>
<!ELEMENT name (family,given)>
<!ELEMENT family (#PCDATA)>
<!ELEMENT given (#PCDATA)>
<!ELEMENT email (#PCDATA)>
```

La définition d'un langage XML avec une DTD

- Exemple de document XML pour la DTD addressBook

```
<addressBook>
  <person>
    <name>
      <family>Wallace</family>
      <given>Bob</given>
    </name>
    <email>bwallace@megacorp.com</email>
  </person>
  <person>
    <name>
      <family>Tuttle</family>
      <given>Magalie</given>
    </name>
    <email>mtuttle@megacorp.com</email>
    <email>ma@megacorp.com</email>
  </person>
</addressBook>
```

La définition d'un langage XML avec une DTD

- Déclaration d'un attribut
 - Syntaxe

```
<!ATTLIST target_tag attr_nom TypeAttribut TypeDef Defaut>
```

- **target_tag**: nom de l'élément contenant cet attribut
- **attr_nom**: nom de l'attribut
- **TypeAttribut**: type d'attribut
- **TypeDef**: définit l'optionnalité de l'attribut ou une valeur par défaut (FIXED)
- **Default**: valeur par défaut (si TypeDef est à FIXED)

La définition d'un langage XML avec une DTD

- Types d'attributs

Types d'attributs (TypeAttribut)	
ID	Permet de définir un identificateur unique pour un élément du document. Donc chaque id doit être différent ! Exemple d'usage: faire des tables de matière.
IDREF	Doit correspondre à un attribut "ID" dans un des éléments du document. Voir ci-dessus.
IDREFS	Doit correspondre à 1 ou plusieurs ID attributs (séparés par des blancs)
(A B C ...)	Liste énumérée de valeurs d'attributs à choix. Vous permet de contrôler un petit peu le contenu que les utilisateurs peuvent rentrer.
CDATA	"Character Data" - Contenu arbitraire, mais normalisé: espaces et fin de lignes convertis en un seul espace !
NMTOKEN	L'utilisateur peut rentrer un seul mot

- Types de TypeDef

Explication de TypeDef	
#IMPLIED	Attribut à option (l'utilisateur peut l'utiliser)
#REQUIRED	Attribut obligatoire (l'utilisateur doit rentrer une valeur)
#FIXED Value	Attribut avec valeur fixe (la valeur est déjà fixée dans la DTD)

La définition d'un langage XML avec une DTD

- Exemple d'addressBook plus complexe

```
<!ELEMENT addressBook (person)+>

<!ELEMENT person (name,email*,link?)>
<!ATTLIST person id ID #REQUIRED>
<!ATTLIST person gender (male|female) #IMPLIED>

<!ELEMENT name (#PCDATA|family|given)*>

<!ELEMENT family (#PCDATA)>

<!ELEMENT given (#PCDATA)>

<!ELEMENT email (#PCDATA)>

<!ELEMENT link EMPTY>
<!ATTLIST link manager IDREF #REQUIRED>
<!ATTLIST link subordinates IDREFS #IMPLIED>
```

Programmez avec les langages XML / DTD et XHTML / CSS

XML: EXERCICES

Le Langage XML et les DTD:

EXERCICES

- Exercice 1

— Analysons ce document XML:

```
<?xml version="1.0"?>
<!DOCTYPE list SYSTEM "list.dtd">
<list>
  <recipe>
    <author>Carol Schmidt</author>
    <recipe_name>Chocolate Chip Bars</recipe_name>
    <meal>Dinner
      <course>Dessert</course>
    </meal>
    <ingredients>
      <item quantity="2/3 C" type="butter" />
      <item quantity="2 C" type="brown sugar" />
      <item quantity="1 tsp" type="vanilla" />
      <item quantity="1 3/4 C" type="unsifted all-purpose flour" />
      <item quantity="1 1/2 tsp" type="baking powder" />
      <item type="salt" />
      <item quantity="3" type="eggs"/>
      <item quantity="1/2 C" type="chopped nuts" />
      <item quantity="2 cups (12-oz pkg.)" type="semi-sweet choc. chips" />
    </ingredients>
    <directions totaltime="50min">
      Preheat oven to 350 degrees. Melt butter; combine with brown sugar and vanilla in large mixing bowl. Set aside to cool. Combine flour, baking powder, and salt; set aside. Add eggs to cooled sugar mixture; beat well. Stir in reserved dry ingredients, nuts, and chips. Spread in greased 13-by-9-inch pan. Bake for 25 to 30 minutes until golden brown; cool. Cut into squares.
    </directions>
  </recipe>
</list>
```

Le Langage XML et les DTD:

EXERCICES

- Exercice 1 Bis

— Créer un langage XML pour créer des documents contenant des liste de recettes de cuisine !

- L'élément **list** (liste de recettes) contient **1 ou plusieurs recipe** (recette)
- Les éléments **recipe** contiennent:
 - un élément **author**,
 - un élément **recipe_name**,
 - un élément **meal** (type de repas)
 - un élément **ingredients** (liste d'ingrédient)
 - un élément **directions** (instructions)
- l'élément **ingredients** contient **1 ou plusieurs item** (ingrédient)
- Les éléments **author**, **recipe_name** et **directions** contiennent que **du texte** (contenu)

Le Langage XML et les DTD: EXERCICES

- L'élément **Item** est composé des attributs **quantity** et **type**
- **Quantity** est optionnel
- **Direction** a un attribut **totaltime**
- **Meal** contient **du texte** et un élément **course**

Programmez avec les langages XML / DTD et XHTML /
CSS

XHTML: LES BASES DU XHTML

les bases du XHTML

- Rappel:
 - Le HTML est le langage d'affichage des pages web
 - Le XHTML et le HTML sont le même langage, seul la structure change (et très peu)
 - XHTML est un langage XML
 - Il est destiné à être analysé par les navigateurs Web
- Si vous avez eu du mal avec XML, ne vous inquiétez pas, ça n'est pas indispensable pour faire du XHTML

les bases du XHTML

- Squelette d'une page XHTML

```
<?xml version="1.0"?>
<!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="fr" >
  <head>
    ...
    ... information d'entête ...
    ...
  </head>
  <body>
    ...
    ... corps de page ...
    ...
  </body>
</html>
```

- Le **corps de page** est le contenu réel de la page, ce qui s'affiche dans l'espace blanc
- Les **informations d'entête** sont des méta-informations, des liens techniques et le titre de la page

les bases du XHTML

- Principales balises
 - **html**: Balise racine d'une page HTML
 - **head**: Balise d'entête
 - **body**: Balise du corps de page
- Au sein de **html**, on ne retrouve que **head** et **body**

les bases du XHTML

- Le XHTML va contenir **la structure de la page**
- Pour le style (couleur, position, police...), il est possible de déterminer cela dans le XHTML mais il est préférable de définir tout cela dans **une feuille de style CSS** (Cascading Style Sheets)

Programmez avec les langages XML / DTD et XHTML /
CSS

XHTML: LES BALISES DU XHTML

Les balise du XHTML

- Les principales balises d'entête
 - **Title**: titre de la page
 - **meta**: balise contenant des méta-informations de différents types

```
<!-- Auteur de la page -->
<meta name="author" content="Jean Dupont" />
<!-- Description de la page -->
<meta name="description" content="La page personnelle de Jean Dupont" />
<!-- Mots-clés de la page -->
<meta name="keywords" content="expériences, recherche, laboratoire, chimie" />
<!-- Adresse de contact -->
<meta name="reply-to" content="monadresse@email.com" />
<!-- Empêcher la mise en cache de la page par le navigateur -->
<meta http-equiv="pragma" content="no-cache" />
<!-- Table de caractères -->
<meta http-equiv="content-type" content="text/html ; charset=iso-8859-1" />
<!-- Rafraîchissement automatique au bout de 10 secondes -->
<meta http-equiv="refresh" content="10 ; URL=http://www.monsite.com" />
```


Les balise du XHTML

- **script:** balise permettant d'insérer un script

```
<script type="text/javascript">
/* Votre script ici */
</script>
```

- **style:** balise permettant d'insérer du CSS

```
<style type="text/css">
/* Votre code CSS ici */
</style>
```

- **link:** balise permettant de lier la page à d'autres fichiers (généralement des fichiers CSS) et d'autres fonctionnalités sont possibles

Les balise du XHTML

- Les principales balises du corps de page

- **a:** permet de créer un lien hypertexte

```
<a href="URL du lien">Cliquez ici (texte du lien)</a>
```

```
 saut de ligne)
```

- **h1 à h6:** balise de titre de paragraphe (1 à 6 pour 6 tailles différentes, du plus grand au plus petit)
- **br (auto-fermante):** saut de ligne
- **hr:** insertion d'une ligne horizontale de séparation

Les balise du XHTML

- **div**: division d'un **bloc**. Ce concept est très important en web 2.0 car un site web doit être conçu en **bloc**
- **span**: division d'un texte
- **ul, ol, li**: Balises permettant la création de liste
 - **ul**: liste
 - **ol**: liste ordonnée
 - **li**: élément de liste

```
<ul>
  <li>article 1</li>
  <li>article 2</li>
</ul>
```

Les balise du XHTML

- **table, tr, td**: balises permettant de créer des tableaux
 - **table**: balise de tableau
 - **tr**: balise de ligne de tableau
 - **td**: balise de case de tableau

```
<table>
  <tr>
    <td>case 1-1</td>
    <td>case 1-2</td>
    <td>case 1-3</td>
  </tr>
  <tr>
    <td>case 2-1</td>
    <td>case 2-2</td>
    <td>case 2-3</td>
  </tr>
</table>
```

- Vous devez toujours avoir le même nombre de case dans chaque lignes (sinon il existe l'attribut **colspan**)

Programmez avec les langages XML / DTD et XHTML /
CSS

XHTML: CRÉER UNE FEUILLE DE STYLE AVEC CSS

Créer une feuille de style avec CSS

- Le CSS est un langage qui permet de définir les paramètres de style d'une ou plusieurs pages web
- En général, nous avons 1 fichier CSS pour tout un site web
- Dans un fichier CSS, on définit des règles sur des identifiants présent dans le code XHTML
- Dans l'en-tête, pour lier un fichier XHTML à un fichier CSS:

```
<link rel="stylesheet" type="text/css" href="URL du fichier CSS" />
```

Créer une feuille de style avec CSS

- Exemple de fichier CSS

```
body{
  text-align:center;
  font-family:Calibri;
}

#content {
  text-align:left;
  margin-left:auto;
  margin-right:auto;
  width:800px;
  border: 1px solid #000;
}

#header {
  text-align:center;
  height:100px;
  border: 1px solid #000;
  font-size:42px;
  text-decoration:bold;
}

#topMenu {
  border: 1px solid #000;
}

#mainbox {
  border: 1px solid #000;
  min-height:50px;
}

#leftMenu {
  float : left;
  width:140px;
  border: 1px solid #000;
  height:100%;
}
```

Créer une feuille de style avec CSS

- Règles

- Il existe 3 type d'identifiants

- **Nom_de_balise**: l'identifiants d'une règle peut être un nom de balise, à ce moment là, la règle s'applique sur toutes les balises de ce nom
- **#id**: Sur toute les balises du XHTML, on peut définir un attribut 'id'; il doit être unique pour chaque balise. La valeur définie peut alors être un identifiant dans le CSS
- **.class**: Sur toutes les balises du XHTML, on peut définir un attribut 'class', qui peut être réutilisé sur plusieurs balises.

Créer une feuille de style avec CSS

- Exemple
 - Nom de balise

```
body{
  text-align:center;
  font-family:Calibri;
}
```

- Id

```
#footer {
  clear:both;
  text-align:center;
}
```

- class

```
.thumbs {
  margin-left:15px;
}
```

Créer une feuille de style avec CSS

- Utilisation des règles dans le XHTML
 - Nom de balise

<body>

```
body{
  text-align:center;
  font-family:Century Gothic;
  background-color:"#ffffff";
}
```

- Id

<div id="mainbox">

```
#mainbox {
  border: 4px solid blue;
  min-height:50px;
  overflow:visible;
}
```

- Classe

(déconnexion)

```
.deco
{
  color:#486AC6;
}
```

Créer une feuille de style avec CSS

- Identifiants spéciaux

- **a:link**: le suffixe :link permet de définir une règle pour les liens non visités
- **a:visited**: le suffixe :visited permet de définir une règle pour les liens visités
- **a:active**: le suffixe :active permet de définir une règle pour les liens actifs (lorsqu'on clique)
- **a:hover**: le suffixe :hover permet de définir une règle pour les liens survolés par la souris

Créer une feuille de style avec CSS

- Les propriétés

- Il existe une centaine de propriétés CSS, elles sont disponibles pour:
 - La **police**
 - Les **bordures**
 - Le **fond**
 - Les **listes**
 - Les **marges**
 - Les **marge interne**
 - Le **positionnement**
 - Les **dimensions**
 - Les **tableaux**
 - ...

Créer une feuille de style avec CSS

- Découvrons ensemble les différentes possibilités des propriétés CSS

Programmez avec les langages XML / DTD et XHTML / CSS

XHTML: EXERCICES

XHTML: Exercices

- Exercice 1

Créer une page web XHTML / CSS avec:

- Un menu latéral gauche
- Un header avec image
- Un footer centré
- Un contenu centrale sur fond gris

XHTML: Exercices

- Exercice 2

- Créer un site web vitrine sur le modèle de MeublyPlus

Sources

- W3C : www.w3.org
- W3schools : www.w3schools.com
- Introduction technique à XML (TECFA - Daniel Schneider, Vivian Synteta, Stephane Lattion)
tecfa.unige.ch/guides/tie/tie.html