

XHTML

Jamel Eddine Jridi

XHTML

- **Contenu** de la page via les balises
- **Apparence** via le Cascading Style Sheet (CSS)
- **Interaction** via Javascript
- Raisons d'utiliser XHTML, **plutôt que** HTML:
 - HTML acquis des balises pour **contrôler les polices**, l'alignement, etc.
 - Syntaxe XHTML est beaucoup **plus stricte**.
 - **Validation syntaxique** des documents XHTML.
- Contenu propre via :
 - XHTML strict
 - Balisage sémantique

Avantages

- Pour le site

Comp
Acc
Lourd

- Plus compréhensible pour les moteurs d'indexation
- Accessibilité accrue
- Moins lourd

- Pour le développeur :

Valid
Maintenance
Combinaison
Moins de balise
plus id

- Validation
- Maintenance et compréhension rapide du code XHTML.
- Combinaison possible avec d'autres vocabulaires (SVG, MathML, SMIL)
- Moins de balises, mais plus d'id

Structure du document (1)

```
<!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>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
    <title>Un page HTML simplissime</title>
</head>
<body>
    <p><b>Bonjour les amis</b></p>
</body>
</html>
```

Structure du document (2)

- Chaque document XHTML doit commencer par l'une des déclarations **DOCTYPE** (DTD) suivantes :

- **Strict**

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

- Balisage vraiment propre, avec aucun attribut de forme (police, couleur, etc.)
- Utilisation du CSS (Cascading Style Sheets)

- **Transitional**

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

- Contient tous les éléments et attributs HTML, y compris les éléments dépréciées (comme la police).

- **Frameset**

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

Structure du document (3)

- `<html>`, `<head>`, `<title>`, et `<body>` sont obligatoires pour chaque document.
- L'ensemble du document doit avoir `<html>` comme sa racine.
- La balise `html` doit avoir l'attribut `xmlns` :

```
<html xmlns = "http://www.w3.org/1999/xhtml">
```
- Un document est défini par son en-tête `<head>` et son corps `<body>`
- La balise `<title>` définit le titre d'un document qui sera affiché dans la barre de titre de la fenêtre du navigateur (en haut de l'écran)

Catégorie de balises

○ Structurelles

- regroupement : `<html>`, `<head>`, `<body>`, ``, `<div>`
- tableau : `<table>`, `<thead>`, `<tbody>`, `<tfoot>`, `<th>`, `<td>`, `<tr>`

○ Sémantiques

- lien : `<a>`, `<script>`
- entête : `<h1>...<h6>`, `<p>`, `<title>`
- listes : ``, ``, ``, `<dl>`, `<dt>`, `<dd>`
- image : ``

○ Présentation

- `<hr>`, `
`
- `<tt>`, `<pre>`
- ``, `<i>`

○ Formulaires

- `<form>`, `<input>`, `<label>`, `<option>`, `<select>`

Syntaxe de base (1)

- Les éléments XHTML doivent être bien imbriqués. Les balises doivent être fermées dans l'ordre inverse d'ouverture.
 - **Invalide** : `<p>ceci est un paragraphe</p>`
 - **Valide** : `<p>ceci est un paragraphe</p>`
- Les noms de balises et des attributs doivent être en **miniscule**.
- **Tout** les éléments XHTML doivent être **fermés**.
 - Pas mal de balises n'ont pas de contenu, par exemple :
 - Le saut de ligne : `
`
 - Les images : ``
 - Les champs de formulaire : `<input type="text" />`
- Commentaires : `<!-- comment -->`

Syntaxe de base (2)

- ★ ○ Les attributs abrégés ne sont désormais plus valides en XHTML (attributs **selected**, **disabled**, **checked** et **readonly**).
 - Invalide : `<input type="text" checked />`
 - Valide : `<input type="text" checked="checked" />`
- ★ ○ Toujours utiliser `&` pour `&` dans les valeurs d'attributs, y compris les URL: ``
 - Invalide : ``
 - Valide : ``
- ★ ○ Toutes les valeurs des attributs doivent être entourées de guillemets et ce même pour les valeurs numériques.
 - Invalide : `<td rowspan=3 >texte</td>`
 - Valide : `<td rowspan="3" >texte</td>`

Text Markup (1)

- Les éléments de description des paragraphes.
 - La balise `<p>`
- Saut de ligne
 - L'effet de la balise `
` est le même que celui de `<p>`, à l'exception de la ligne blanche
- Exemple :

```
On the plains of hesitation <p> bleach the
bones of countless millions </p> <br />
who, at the dawn of victory <br /> sat down
to wait, and waiting, died.
```
- Exécution:

```
On the plains of hesitation

bleach the bones of countless millions
who, at the dawn of victory
sat down to wait, and waiting, died.
```

Text Markup (2)

○ Headings.

- 6 tailles : <h1> ... <h6>
- 1, 2 et 3 proposent une police de font supérieur à celle par défaut.
- 4 : police par défaut.
- 5 et 6 : inférieur à la police par défaut.

```
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head> <title> Headings </title>
</head>
<body>
  <h1> Aidan's Airplanes (h1) </h1>
  <h2> The best in used airplanes (h2) </h2>
  <h3> "We've got them by the hangarful" (h3)</h3>
  <h4> We're the guys to see for a good used airplane (h4) </h4>
  <h5> We offer great prices on great planes(h5) </h5>
  <h6> No returns, no guarantees, no refunds,
    all sales are final (h6) </h6>
</body>
</html>
```

Aidan's Airplanes (h1)

The best in used airplanes (h2)

"We've got them by the hangarful" (h3)

We're the guys to see for a good used airplane (h4)

We offer great prices on great planes (h5)

No returns, no guarantees, no refunds, all sales are final! (h6)

Text Markup (3)

○ Style de font et taille .

- Gras ``
- Italique `<i>`
- Large `<big>`
- Petit `<small>`
- Monospace `<tt>`

The `<big>` sleet `<big>` in `<big>` `<i>` Crete
`</i>``
` lies `</big>` completely `</big>`
in `</big>` the street

The sleet in *Crete*

lies completely in the street

○ Blockquotes `<blockquote>`.

- Définir les citations
- Les navigateurs les indentent souvent, et parfois en italique.

○ Superscripts `<sup>` et subscripts `<sub>` .

- Example: `x₂³`
- Display: x_2^3

○ Block éléments ne peuvent pas être imbriquées dans les inline éléments. `<i>``<p>`ceci est un paragraphe`</p>``</i>`

Image

- Les images sont insérées dans un document avec la balise `img` dont l'attribut `src` définit le path vers l'image.
 - L'attribut `alt` est nécessaire par XHTML.

```
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head> <title> Images </title>
  </head>
  <body>
    <h1> Aidan's Airplanes </h1>
    <h2> The best in used airplanes </h2>
    <h3> "We've got them by the hangarful"
    </h3>
    <h2> Special of the month </h2>
    <p>
      1960 Cessna 210 <br />
      577 hours since major engine overhaul
      <br />
      1022 hours since prop overhaul
      <br /><br />
      <img src = "c210new.jpg"
            alt = "Picture of a Cessna 210"/>
      <br />
      Buy this fine airplane today at a
      remarkably low price <br />
      Call 999-555-1111 today!
    </p>
  </body>
</html>
```

Aidan's Airplanes

The best in used airplanes

"We've got them by the hangarful"

Special of the month

1960 Cessna 210

577 hours since major engine overhaul

1022 hours since prop overhaul

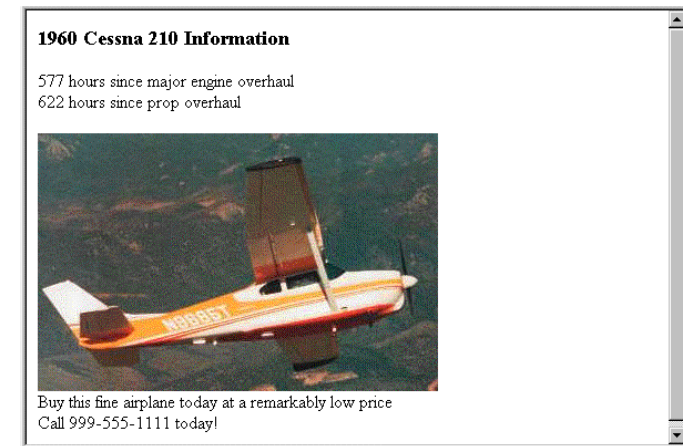


Buy this fine airplane today at a remarkably low price
Call 999-555-1111 today!

Hypertext Link

- Hypertext est la base du web.
- Un lien est spécifié par l'attribut `href` de la balise `<a>`
 - Le contenu de `<a>` est le lien visuel dans le document

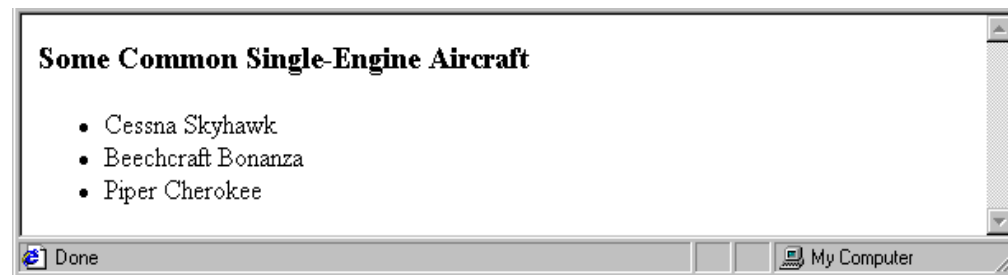
```
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head> <title> Links </title>
</head>
<body>
  <h1> Aidan's Airplanes </h1>
  <h2> The best in used airplanes </h2>
  <h3> "We've got them by the hangarful"
</h3>
  <h2> Special of the month </h2>
  <p>
    1960 Cessna 210 <br />
    <a href = "C210data.html">
      Information on the Cessna 210 </a>
  </p>
</body>
</html>
```



Liste (1)

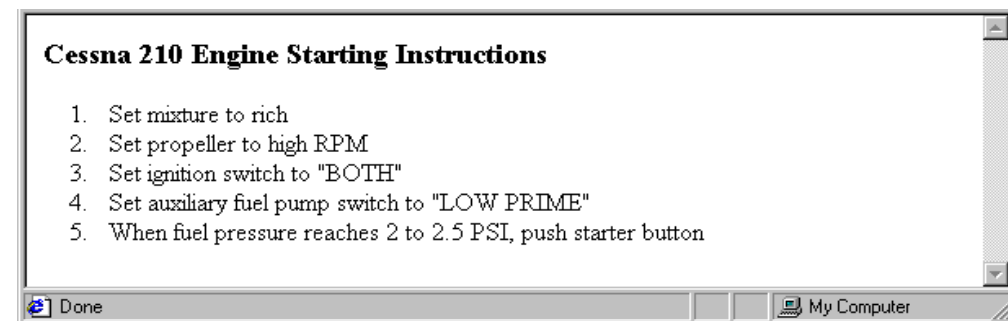
○ Liste non ordonnée.

```
<h3> Some Common Single-Engine Aircraft </h3>
<ul>
  <li> Cessna Skyhawk </li>
  <li> Beechcraft Bonanza </li>
  <li> Piper Cherokee </li>
</ul>
```



○ Liste ordonnée.

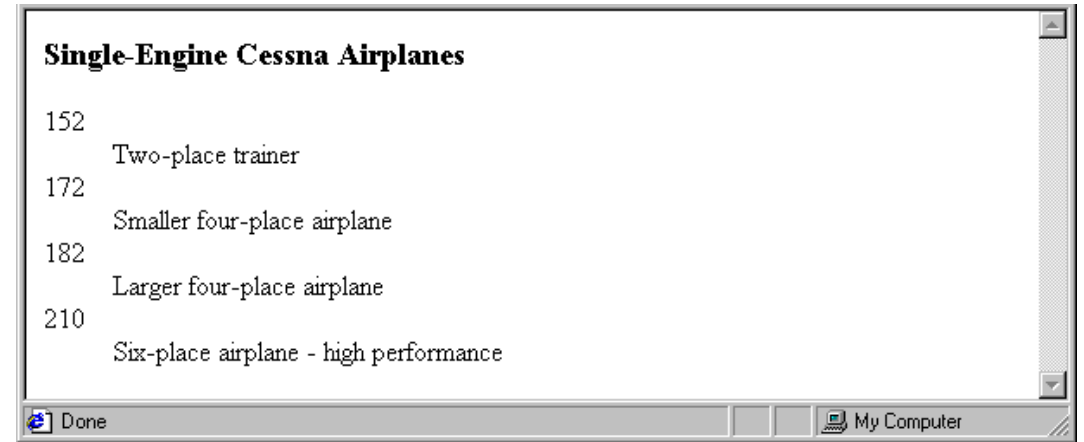
```
<h3> Cessna 210 Engine Starting Instructions
</h3>
<ol>
  <li> Set mixture to rich </li>
  <li> Set propeller to high RPM </li>
  <li> Set ignition switch to "BOTH" </li>
  <li> Set auxiliary fuel pump switch to
    "LOW PRIME" </li>
  <li> When fuel pressure reaches 2 to 2.5
    PSI, push starter button </li>
</ol>
```



Liste (2)

- Liste de description.

<h3> Single-Engine Cessna Airplanes </h3>
<dl >
 <dt> 152 </dt>
 <dd> Two-place trainer </dd>
 <dt> 172 </dt>
 <dd> Smaller four-place airplane </dd>
 <dt> 182 </dt>
 <dd> Larger four-place airplane </dd>
 <dt> 210 </dt>
 <dd> Six-place airplane - high performance
 </dd>
</dl>



- Les listes peuvent être imbriquées.
- La liste imbriquée doit être un élément liste.

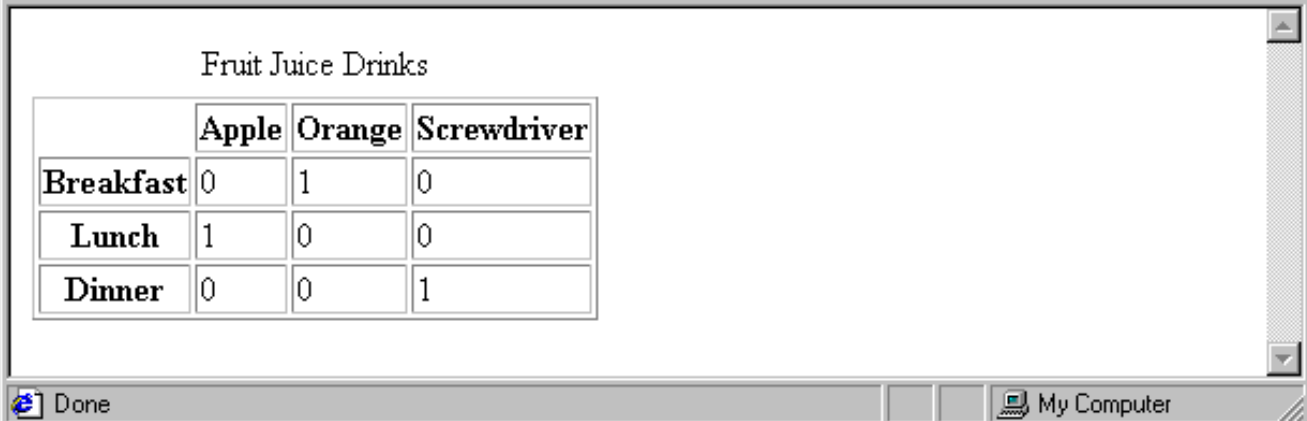
Table (1)

- Un tableau est une matrice de cellules, ayant chacune un contenu
- Les cellules peuvent inclure presque tous les éléments
- Une table est définie comme le contenu d'une balise `<table>`
- Un attribut `border` dans la balise `<table>` spécifie une frontière entre les cellules
- L'attribut `border` peut être réglé à un certain nombre, qui sera la largeur de la bordure
- Sans l'attribut `border`, le tableau aura pas de lignes!
- La balise `<caption>` définit le titre d'un tableau qui suit immédiatement `<table>`

Table (2)

- Les tableaux sont divisés en lignes définies avec la balise `<tr>`.
- Les lignes de tableau sont divisées en colonnes avec la balise `<td>`.
- Une rangée de table peut être divisé en rubriques avec la balise `<th>`.

```
<table border = "border">
  <caption> Fruit Juice Drinks </caption>
  <tr>
    <th> </th>
    <th> Apple </th>
    <th> Orange </th>
    <th> Screwdriver </th>
  </tr>
  <tr>
    <th> Breakfast </th>
    <td> 0 </td>
    <td> 1 </td>
    <td> 0 </td>
  </tr>
  <tr>
    <th> Lunch </th>
    <td> 1 </td>
    <td> 0 </td>
    <td> 0 </td>
  </tr>
  <tr>
    <th> Dinner </th>
    <td> 0 </td>
    <td> 0 </td>
    <td> 1 </td>
  </tr>
</table>
```

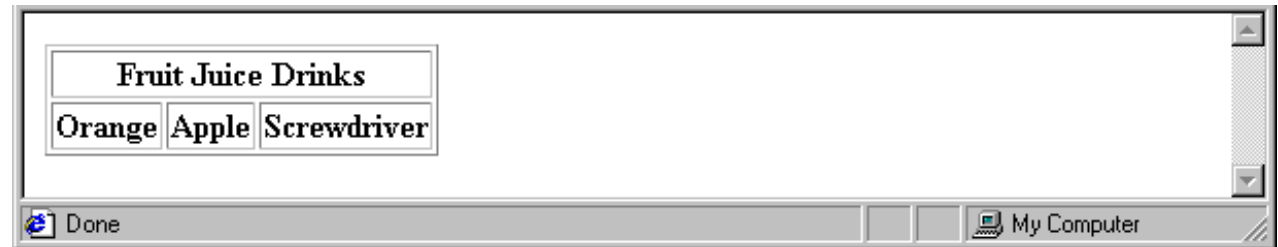


	Apple	Orange	Screwdriver
Breakfast	0	1	0
Lunch	1	0	0
Dinner	0	0	1

Table (3)

- Colspan : attribut pour fusionner les colonnes.
- Rowspan : attribut pour fusionner les lignes.

```
<table border = "border">  
  <tr>  
    <th colspan = "3"> Fruit Juice Drinks </th>  
  </tr>  
  <tr>  
    <th> Orange </th>  
    <th> Apple </th>  
    <th> Screwdriver </th>  
  </tr>  
</table>
```



```
<table border = "border">  
  <tr>  
    <td> item 1 </td>  
    <td rowspan= "2"> item 2 </td>  
    <td> item 3 </th>  
  </tr>  
  <tr>  
    <td> item 4 </td>  
    <td> item 5 </td>  
  </tr>  
</table>
```

Item 1	Item 2	Item 3
Item 4		Item 5

Formulaire (1)

- Un formulaire permet de saisir et de transmettre des informations du navigateur au serveur.
- HTML offre une collection d'objets, appelés widgets, (boutons radio et cases à cocher), pour définir les informations d'un formulaire.
- Un formulaire est déclaré grâce à la balise `<form>` dont on précisera généralement 3 attributs :
 - `name` : le nom que l'on souhaite donner au formulaire
 - `action` : le chemin vers la page où les données du formulaire seront envoyées. Interaction avec le serveur pour sauvegarder les données dans une base, les envoyer par mail, ou réaliser tout autre traitement.
 - `method` : la méthode d'envoi des données (POST ou GET)

Formulaire (2)

- **Get** : les données du formulaire seront encodées dans une URL.
 - Celle-ci est composée du nom de la page ou du script à charger avec les données de formulaire empaquetée dans une chaîne.
- **Post** : les données du formulaire n'apparaissent pas dans l'URL.
 - En conséquence, il n'est pas possible de récupérer directement les données en JavaScript.
 - Il faut ajouter du code PHP dans ce cas.

Formulaire – Widgets (3)

- `<input>` utilisé pour créer des contrôles interactifs dans les formulaires web.
 - L'attribut `type` spécifie le type de l'objet à créer.
- checkbox

```
Grocery Checklist
<form action = "">
  <p>
    <input type = "checkbox"  name ="groceries"
      value = "milk"  checked = "checked" />
    Milk
    <input type = "checkbox"  name ="groceries"
      value = "bread"/>
    Bread
    <input type = "checkbox"  name = "groceries"
      value= "eggs"/>
    Eggs
  </p>
</form>
```



Grocery Checklist

☒ Milk ☐ Bread ☐ Eggs

Formulaire – Widgets (4)

o Radio button

Age Category

```
<form action = "">
```

```
<p>
```

```
<input type = "radio" name = "age" value = "under20" checked = "checked"/> 0-19
```

```
<input type = "radio" name = "age" value = "20-35"/> 20-35
```

```
<input type = "radio" name = "age" value = "36-50"/> 36-50
```

```
<input type = "radio" name = "age" value = "over50"/> Over 50
```

```
</p>
```

```
</form>
```

A screenshot of a web browser window displaying a form titled "Age Category". Below the title, there are four radio buttons arranged horizontally. The first radio button, labeled "0-19", is selected, indicated by a small green dot inside the circle. The other three radio buttons, labeled "20-35", "36-50", and "Over 50", are unselected, shown as empty circles.

o Liste déroulante

Grocery Menu - milk, bread, eggs, cheese

```
<form action = "">
```

```
<p>
```

```
With size = 1 (the default)
```

```
<select name = "groceries">
```

```
<option> milk </option>
```

```
<option> bread </option>
```

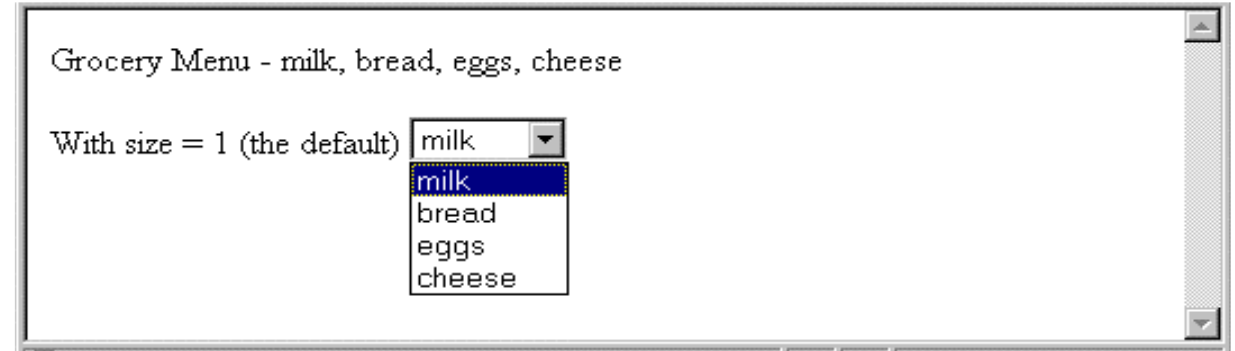
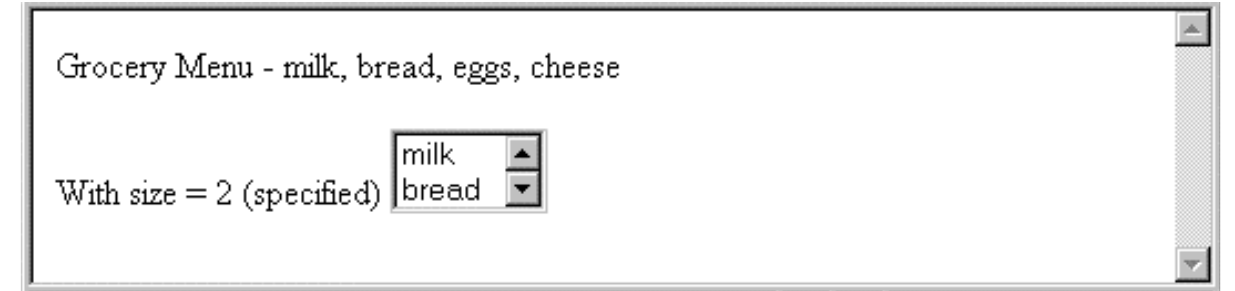
```
<option> eggs </option>
```

```
<option> cheese </option>
```

```
</select>
```

```
</p>
```

```
</form>
```

A screenshot of a web browser window displaying a form titled "Grocery Menu - milk, bread, eggs, cheese". Below the title, there is a text label "With size = 1 (the default)" followed by a dropdown menu. The dropdown menu is open, showing a list of four options: "milk", "bread", "eggs", and "cheese". The "milk" option is currently selected and highlighted with a blue background.A screenshot of a web browser window displaying a form titled "Grocery Menu - milk, bread, eggs, cheese". Below the title, there is a text label "With size = 2 (specified)" followed by a dropdown menu. The dropdown menu is open, showing a list of two options: "milk" and "bread". The "milk" option is currently selected and highlighted with a blue background.