

ISTA 230



HTML Lists

CSS

HTML Lists

- Useful for grouping related information/elements together
- Provide additional means of styling your content

CSS

HTML Lists

- Three types of lists
 - Unordered Lists
 - Ordered Lists
 - Description Lists

CSS

HTML Unordered Lists

- Useful for grouping things whose order doesn't matter
- E.g., a shopping list

CSS

HTML Unordered Lists

HTML

```
<ul>  
  <li>milk</li>  
  <li>bread</li>  
  <li>butter</li>  
  <li>coffee</li>  
</ul>
```

Browser

- milk
- bread
- butter
- coffee

CSS

HTML Ordered Lists

- Useful for grouping things whose order **DOES** matter
- E.g., a recipe

CSS

HTML Ordered Lists

HTML

```
<ol>
  <li>Gather ingredients</li>
  <li>Mix ingredients together</li>
  <li>Place ingredients in a baking dish</li>
  <li>Bake in oven for an hour</li>
</ol>
```

Browser

1. Gather ingredients
2. Mix ingredients together
3. Place ingredients in a baking dish
4. Bake in oven for an hour

Note that the browser automatically adds numeric list item markers for ordered lists.

CSS

HTML Description Lists

- Useful for associating name-value groups
- E.g., metadata
- E.g., glossaries

CSS

HTML Description Lists

HTML

```
<dl>  
  <dt>Title</dt>  
  <dd>Robopocalypse</dd>  
  <dt>Author</dt>  
  <dd>Daniel H. Wilson</dd>  
  <dt>Publisher</dt>  
  <dd>Doubleday</dd>  
</dl>
```

Browser

Title	Robopocalypse
Author	Daniel H. Wilson
Publisher	Doubleday

CSS

HTML Description Lists

When using a description lists for definitions, it is recommended that you also use `<dfn>` tags around the term being defined. While this doesn't change the way the element is displayed in most browsers, it does provide our content with more structure, organization, and meaning (which is the purpose of HTML).

HTML

```
<dl>
  <dt><dfn>milk</dfn></dt>
  <dd>A white, liquid dairy product.</dd>
  <dt><dfn>butter</dfn></dt>
  <dd>A yellow, solid dairy product.</dd>
</dl>
```

Browser

<i>milk</i>	A white, liquid dairy product.
<i>butter</i>	A yellow, solid dairy product.

CSS

HTML Description Lists

If you have a term that has multiple descriptions, you shouldn't use more than one `<dt>` for the term. For example, the word 'coffee' can have many definitions.

HTML

```
<dl>
  <dt><dfn>coffee</dfn></dt>
  <dd> - a beverage made from roasted, ground coffee beans</dd>
  <dd> - a social gathering at which coffee is consumed</dd>
  <dd> - a medium to dark brown color</dd>
</dl>
```

We include a single `<dt>` element for the term, followed by a `<dd>` element for each definition. The result is as follows:

Browser

```
coffee
  - a beverage made from roasted, ground coffee beans
  - a social gathering at which coffee is consumed
  - a medium to dark brown color
```

CSS

HTML Description Lists

Similarly, we may have a set of terms that are synonyms for one another and all share the same definition. For this case, we include multiple <dt> elements for our terms, followed by a single <dd> element for the shared definition.

HTML

```
<dl>
  <dt><dfn>soda</dfn></dt>
  <dt><dfn>pop</dfn></dt>
  <dt><dfn>cola</dfn></dt>
  <dd> - A sweet, carbonated beverage</dd>
</dl>
```

Browser

```
soda
pop
cola
  - A sweet, carbonated beverage
```

CSS

HTML Lists

To determine what type of list your content requires, you can use the following set of questions:

- What list should you use?
 - Are you making a list of associated name-value groups?
 - If yes, use a description list.
 - If no, go to the next question.
 - Is the order of items important?
 - If yes, use an ordered list.
 - If no, use an unordered list.

CSS

HTML Nested Lists

List items can also contain sub-lists. Note that the sublist below, providing a secondary list of fruits, is placed in between the starting and closing `` tags for the parent item.

HTML

```
<ul>
  <li>milk</li>
  <li>fruit
    <ul>
      <li>apples</li>
      <li>bananas</li>
      <li>cherries</li>
    </ul>
  </li>
  <li>bread</li>
</ul>
```

CSS

HTML Nested Lists

Browser

- milk
- fruit
 - apples
 - bananas
 - cherries
- bread

When creating sub-lists, note that the browser automatically provides a different list item marker (i.e., bullet) for the items in the sub-list.

CSS

Why use HTML Lists?

- Provides better semantic information
- Allows for easier re-ordering of ordered items

If we had to specify the numbers for each item in an ordered list, changing the order in which things are displayed would require us to modify each list item. Using ordered lists (``), the browser assigns the number value for us, allowing us to easily move `` items around without worrying about the numbering.

- Allows for easier styling through CSS

CSS

CSS and Lists

We can use CSS to style our lists

When we apply the color property to a list, it applies the color to both the text of each list item as well as the list item marker (bullet, number, etc.) of the item.

CSS

```
ul  
{  
  color:#f00;  
}
```

Browser

- List Item 1
- List Item 2
- List Item 3

CSS

CSS and Lists

We can use CSS to style our lists

Using the list-style-type property, we can change what type of list item marker the list uses. In the example below, we change the list item marker from a filled circle to an unfilled circle.

CSS

```
ul
{
  color:#f00;
  list-style-type: circle;
}
```

Browser

- List Item 1
- List Item 2
- List Item 3

CSS

list-style-type - Options

Below are the list-style-type options that are well supported by all browsers. It is worth noting that 'disc' is the default for unordered lists and 'decimal' is the default for ordered lists.

- disc
- circle
- square
- none

1. decimal
- A. upper-alpha
- a. lower-alpha
- I. **upper-roman**
- i. **lower-roman**

CSS

list-style-type Property

CSS

```
ul li { list-style-type: disc; }
```

```
ol li { list-style-type: decimal; }
```

```
li { list-style-type: none; }
```

```
/*  
 * Using this CSS property, we can specify  
 * what type of list item markers should be  
 * used for a given list item. We can also  
 * use the keyword 'none' to specify that no  
 * list item marker should be displayed.  
 *  
 *  
 * Default Value: disc for <ul>, decimal for <ol>  
 * Inherited: Yes  
 *  
 */
```

CSS




CSS and Lists

We can also define custom list item markers using a combination of an image and the 'list-style-image' property.

CSS

```
ul  
{  
  list-style-image: url('../images/chili.png');  
}
```

Browser

-  List Item 1
-  List Item 2
-  List Item 3

Just because you can doesn't mean you should. Replacing list item markers with images can be a tricky operation, often leading to decreased readability when done poorly.

CSS

list-style-image Property

CSS

```
li { list-style-image: url('../images/bullet.png'); }
```

```
/*  
 * Using this CSS property, we can specify  
 * the path to an image that should be used  
 * as the marker for a given list item.  
 * We can also use the keyword 'none' to  
 * specify that no image should be used  
 * and that the list-style-type property  
 * should be used instead.  
 *  
 *  
 * Default Value: none  
 * Inherited: Yes  
 *  
 */
```

CSS

List Items and the Box Model

CSS

```
li  
{  
  background-color:#999;  
}
```

Browser

- List Item 1
- List Item 2
- List Item 3

It is worth noting that background properties will not extend to markers in a list item.

CSS

List Items and the Box Model

Likewise, markers will appear outside of borders applied to list items.

CSS

```
li  
{  
  border:1px solid #000;  
}
```

Browser

- List Item 1
- List Item 2
- List Item 3

CSS

List Items and the Box Model

We can change the way the browser displays list item markers using the 'list-style-position' property.

CSS

```
li
{
border:1px solid #000;
list-style-position: inside
}
```

Browser

- List Item 1
- List Item 2
- List Item 3

CSS

list-style-position Property

CSS

```
li { list-style-position: inside; }
```

```
li { list-style-position: outside; }
```

```
/*  
 * Using this CSS property, we can specify  
 * whether list item markers should  
 * display inside or outside of  
 * the content area.  
 *  
 * Default Value: outside  
 * Inherited: Yes  
 *  
 */
```

CSS

List Items and the Box Model

By default, browsers apply a margin to both list elements as well as individual list items.

CSS

```
ul, li  
{  
}
```

Browser

A paragraph for reference

- List Item 1
- List Item 2
- List Item 3

A paragraph for reference

CSS

List Items and the Box Model

While we can set the margins to '0', doing so will result in list item markers displaying out of alignment with the rest of the elements, creating a ragged left edge. *Note that this only applies when list-style-position is set to outside.*

CSS

```
ul, li
{
  margin-left: 0;
}
```

Browser

```

|
| A paragraph for reference
|
| • List Item 1
| • List Item 2
| • List Item 3
|
| A paragraph for reference
|
|
```

CSS

List Items and the Box Model

To keep from having a ragged left edge on list items, we can apply a 1em margin to our list element while removing the left margin on the list items.

CSS

```
ul { margin-left: 1em; }  
li { margin-left: 0; }
```

Browser

A paragraph for reference

- List Item 1
- List Item 2
- List Item 3

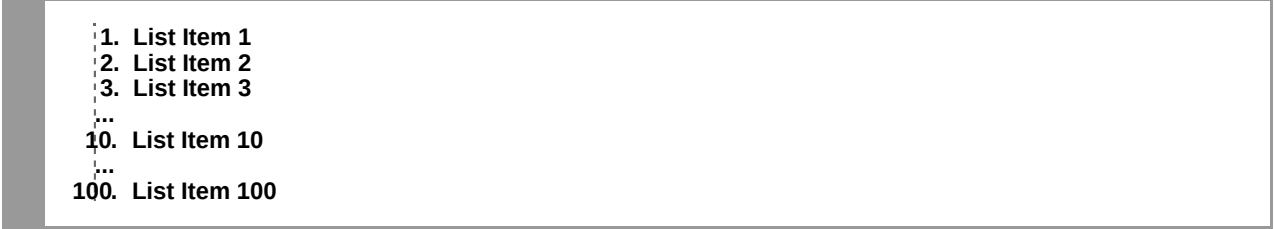
A paragraph for reference

CSS

List Items and the Box Model

While the approach described above works well for creating a smooth left edge for unordered lists, it is worth noting that the technique does not have as consistent an effect on ordered lists, particularly when there is a large number of list items.

Browser

- 
- 1. List Item 1
 - 2. List Item 2
 - 3. List Item 3
 - ...
 - 10. List Item 10
 - ...
 - 100. List Item 100

CSS

HTML Lists

- Useful for grouping related information/elements together
- Provide additional means of styling your content
- Good option for navigation!

CSS

Horizontal Lists

Instead of using a combination of divs and a elements for our navigation, we can use an unordered list to provide a more semantically-appropriate approach.

HTML

```
<ul id="navigation">
  <li><a href='...'>Home</a></li>
  <li><a href='...'>About Me</a></li>
  <li><a href='...'>Projects</a></li>
  <li><a href='...'>Contact Me</a></li>
</ul>
```

Most of the time, you'll want to use an unordered list for your navigation. If you're creating a web application that walks users through the navigation step-by-step, an ordered list might be more appropriate, though that tends to be a rarer case.

CSS

Horizontal Lists

To create a horizontal navigation using an unordered list, we'll set the width of each of our list items, float them to the left, and remove the list item marker using "list-style-type: none". We'll also want to set 'overflow:auto' on the unordered list element to ensure that it doesn't collapse when the list items are floated.

CSS

```
ul
{
  background-color:#333; color:#fff;
  overflow:auto;
}
li
{
  width:21%;
  padding: 0 1%;
  margin: 0 1%;
  list-style-type:none;
  float: left;
}
```

Browser



Home About Me Projects Contact Me

CSS

Horizontal Lists

In order to make sure that our navigational links display the way we want when hovered over, we'll set them to display as block elements and set the width to auto.

CSS

```
a { display: block; width: auto; }  
a:hover  
{  
  background-color: #f00;  
}
```

Browser



Home About Me Projects Contact Me

CSS

Review - Absolute Positioning

CSS

```
div { position: absolute; }
```

- The element is taken out of the document flow
- Position is determined based on the **containing element**
 - The nearest ancestor with a non-static 'position' value
 - The topmost element (html)

CSS

Dropdown Lists

Using nested unordered lists, we can create an extensive set of links for our navigation.

HTML

```
<ul>
  <li>Movies
    <ul>
      <li>Finding Nemo</li>
      <li>The Lion King</li>
    </ul>
  </li>
  <li>Albums
    <ul>
      <li>Purple Rain</li>
      <li>Pet Sounds</li>
    </ul>
  </li>
</ul>
```

CSS

Dropdown Lists

While the HTML for our navigation is semantically appropriate, the display of the lists isn't the most user friendly...

Browser

- **Movies**
 - **Finding Nemo**
 - **The Lion King**
- **Albums**
 - **Purple Rain**
 - **Pet Sounds**

CSS

Dropdown Lists

To help us format our navigation, we'll add class attributes to all of our lists and list items. Note that the class names relate to whether the list is the primary menu or the secondary menu.

HTML

```
<ul class='menu'>
  <li class='item'>Movies
    <ul class='sub_menu'>
      <li class='sub_item'>Finding Nemo</li>
      <li class='sub_item'>The Lion King</li>
    </ul>
  </li>
  <li class='item'>Albums
    <ul class='sub_menu'>
      <li class='sub_item'>Purple Rain</li>
      <li class='sub_item'>Pet Sounds</li>
    </ul>
  </li>
</ul>
```

CSS

Dropdown Lists

Using CSS similar to what we used for our horizontal navigation, we'll format the lists so that the two top-level list items display side by side. Because the sub-lists are children of the top-level list items, they also display side by side.

CSS

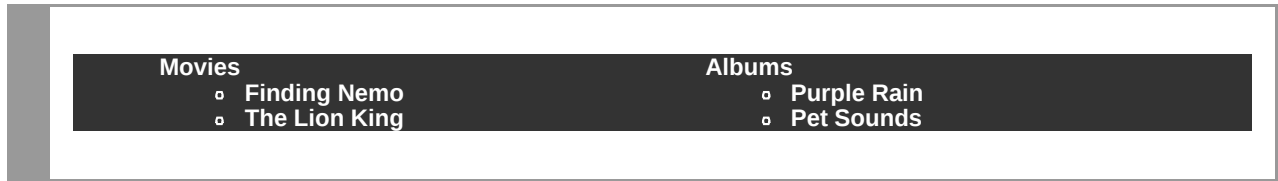
```
ul.menu
{
  background-color:#333; color:#fff;
  overflow:auto;
}

li.item
{
  width:50%;
  list-style-type:none;
  float: left;
}
```

CSS

Dropdown Lists

Browser



A screenshot of a browser's dropdown menu. The menu is dark gray with white text. It is divided into two columns. The left column is titled "Movies" and contains two items: "Finding Nemo" and "The Lion King". The right column is titled "Albums" and contains two items: "Purple Rain" and "Pet Sounds". The menu is displayed over a light gray background.

Movies	Albums
◦ Finding Nemo	◦ Purple Rain
◦ The Lion King	◦ Pet Sounds

CSS

Dropdown Lists

So that the submenu items line up with the top-level items, we'll remove all of the margins, padding, and list item markers on the submenus.

CSS

```
ul.submenu
{
  margin: 0;
  padding: 0;
}

li.sub_item
{
  margin: 0;
  padding: 0;
  list-style-type: none;
}
```

CSS

Dropdown Lists

Browser

<p>Movies Finding Nemo The Lion King</p>	<p>Albums Purple Rain Pet Sounds</p>
--	--

CSS

Dropdown Lists

In order to create a dropdown menu, we'll use absolute positioning. We'll set our submenus to be positioned absolutely. We'll also set the position of our top level list items to relative. This will cause the top-level list item to act as the containing element for our absolutely positioned submenus.

CSS

```
li.item
{
  position: relative;
}

ul.sub_menu
{
  position: absolute;
}
```

CSS

Dropdown Lists

We'll also adjust the background color and the width of the submenus so that they match the top-level list items.

CSS

```
ul.sub_menu
{
  background-color: #333;
  width: 100%;
}
```

CSS

Dropdown Lists

Lastly, we'll use the 'left' property to hide the submenus off the left side of the screen.

CSS

```
ul.sub_menu  
{  
  left: -9999px;  
}
```

CSS

Dropdown Lists

When users view our navigation, the submenus will not be visible. However, we can use the pseudo-class 'hover' to make our submenus reappear when the parent list item is hovered over.

CSS

```
li.item:hover ul.sub_menu  
{  
  left: 0;  
}
```

Because we set the position of the parent list items to 'relative', the submenus will be positioned relative to it (since it's the containing element).

CSS

Dropdown Lists

With a little additional styling, we have a fully functional dropdown menu implemented with CSS.

Browser

