

# HTML Chapter 3

## By Dwight Watt

Copyright 2003-2014

Now that we can do basic web pages with text and images and links, let's look at other ways to present our material better. In this chapter we will look at better ways to put lists in our web pages and to make them look more professional

There are three types of lists that you can use in HTML. There are definition lists, ordered lists and unordered lists. We will look at each one and see what the differences are and how to do each type.

Ordered and unordered lists are very similar to each other. They both list the items. Unordered lists put a bullet in front of each item. Ordered lists put a number or letter in front of each item, by default a number. An example of an ordered list is:

1. Ginny
2. Hannah
3. Byte
4. Jock

An example of an unordered list is:

- Ginny
- Hannah
- Byte
- Jock

Definition lists are used to list a value and then list a definition for it, similar to a dictionary. So there are two parts: the term and the definition. However you are not limited to pure definitions. An example is:

Ginny  
    My dog  
Hannah  
    My dog  
Byte  
    My cat  
Jock  
    My dog

Notice the terms were on one line and the definitions on the next line.

Now for the tag codes we need to build lists. The ordered lists and unordered lists will be very similar. The code for definition lists will be different, but work in a similar way.

To do an ordered or unordered list you will use the tag codes `<li>` and `</li>` to mark each entry. It is important that you use the closing `</li>` tags as some browsers can produce

some weird results without it. Internet Explorer is not picky on the closing `</li>` tag. Code validators are picky on always having closing tags or marking self-closing tags as self-closing by using `/` in tags like `image`. However you do not know what browser your users are using so build for both.

Ordered lists will use a tag code of `<ol>` to begin the list and a tag code of `</ol>` to end the list. There are two attributes you can use with the `<ol>` tag code.

The first is that you can use a type attribute. This attribute lets you assign the type of character in front of each of your entries. The values are A for uppercase letters (A, B, C etc), a for lowercase letters (a, b, c, etc.), I for Roman numbers (I, II, III, IV, etc.), i for Lowercase Roman numbers (I, ii, iii, iv, etc.) and l for Arabic (normal) numbers (1, 2, 3, etc.).

The second attribute you can use is start. You will probably seldom use this attribute as normally you want list to start with A, 1 or I. However if we had an Arabic numbered list what we wanted to start at 5 instead of 1 we would use the tag code of `<ol start=5>`.

So the code for a ordered list using defaults would be:

```
<ol>
  <li>Ginny</li>
  <li>Hannah</li>
  <li>Byte</li>
  <li>Jock</li>
</ol>
```

Which would produce output of:

1. Ginny
2. Hannah
3. Byte
4. Jock

If we used the type attribute so the list uses Roman numbers our code would be:

```
<ol type=I>
  <li>Ginny</li>
  <li>Hannah</li>
  <li>Byte</li>
  <li>Jock</li>
</ol>
```

The results will be:

- I. Ginny
- II. Hannah
- III. Byte
- IV. Jock

Unordered lists will use the <ul> and </ul> tag codes to start and end the list. The <ul> tag code can also use a variation of the type attribute. However in the unordered lists you can only control the type of symbol. You can have a circle, square or disc. The circle is the outside line of them while a disc and square are colored in when used in Internet Explorer (however different browsers and different versions of browsers may give different results). You will use the <li and </li> on the unordered lists like on an ordered list. An example of the code for an unordered list is:

```
<ul>
  <li>Ginny</li>
  <li>Hannah</li>
  <li>Byte</li>
  <li>Jock</li>
</ul>
```

The results would be:

- Ginny
- Hannah
- Byte
- Jock

If we were to decide we wanted the list to use squares instead of discs we would use the following code:

```
<ul type=square>
  <li>Ginny</li>
  <li>Hannah</li>
  <li>Byte</li>
  <li>Jock</li>
</ul>
```

The results will be:

- Ginny
- Hannah
- Byte
- Jock

You can use the type attribute also on the <li> tag codes in both ordered and unordered lists. Normally you should use them on the <ol> and <ul> tag codes. The start attribute may also be used on a <li> tag code in an ordered list.

You can also nest lists inside other lists. You are not limited to nested only similar types of lists but also ordered lists can be nested inside unordered lists or vice versa. With nesting of lists doing outlines becomes simple to make them look professional. An example of a nested ordered list would be:

- I. Fruits
  - A. Apple

- B. Oranges
- C. Grapes
- II. Vegetables
  - A. Broccoli
  - B. Beans
  - C. Corn
- III. Meats
  - A. Beef
  - B. Pork

The code for the above nested lists will be:

```

<ol type=I>
  <li>Fruits</li>
    <ol type=A>
      <li>Apple</li>
      <li>Oranges</li>
      <li>Grapes</li>
    </ol>
  <li>Vegetables</li>
    <ol type=A>
      <li>Broccoli</li>
      <li>Beans</li>
      <li>Corn</li>
    </ol>
  <li>Meats</li>
    <ol type=A>
      <li>Beef</li>
      <li>Pork</li>
    </ol>
</ol>

```

Notice how the code was indented to make it more readable and to see the nested lists as being nested in the other list. This is a good coding practice.

Definition lists will be different. They will use the <dl> and </dl> tag codes to mark the beginning and end of the list. Each of the terms on the list will use the <dt> and </dt> codes to mark the term. To mark the definition use the <dd> and </dd> tag codes. Notice the <dd> and <dt> codes are similar to the <li> tag code but give you the ability to be more specific. The code for a definition list will be:

```

<dl>
  <dt>Ginny</dt><dd>My dog</dd>
  <dt>Hannah</dt><dd>my dog</dd>
  <dt>Byte</dt><dd>My cat</dd>
  <dt>Jock</dt><dd>My dog</dd>
</dl>

```

The results of this code will be:

Ginny

My dog  
Hannah  
my dog  
Byte  
My cat  
Jock  
My dog

The tag codes we have covered in this chapter include `<ol>`, `<ul>`, `<dl>`, `<li>`, `<dt>` and `<dl>`. We have also looked at attributes to control the appearance of our lists.