

Styling lists

← Previous

↑ Overview: Styling text

Next →

[Lists](#) behave like any other text for the most part, but there are some CSS properties specific to lists that you need to know about, and some best practices to consider. This article explains all.

Prerequisites: Basic computer literacy, HTML basics (study [Introduction to HTML](#)), CSS basics (study [Introduction to CSS](#)), [CSS text and font fundamentals](#).

Objective: To become familiar with the best practices and properties related to styling lists.

A simple list example

To begin with, let's look at a simple list example. Throughout this article, we'll look at unordered, ordered, and description lists — all have styling features that are similar, and some that are particular to their type of list. The unstyled example is [available on Github](#) (check out the [source code](#) too.)

The HTML for our list example looks like so:

```
1 <h2>Shopping (unordered) list</h2>
2
3 <p>Paragraph for reference, paragraph for reference, paragraph for reference,
4 paragraph for reference, paragraph for reference, paragraph for reference.</p>
5
6 <ul>
7   <li>Hummus</li>
8   <li>Pita</li>
9   <li>Green salad</li>
10  <li>Halloumi</li>
11 </ul>
12
13 <h2>Recipe (ordered) list</h2>
14
15 <p>Paragraph for reference, paragraph for reference, paragraph for reference,
16 paragraph for reference, paragraph for reference, paragraph for reference.</p>
17
18 <ol>
19   <li>Toast pita, leave to cool, then slice down the edge.</li>
20   <li>Fry the halloumi in a shallow, non-stick pan, until browned on both sides.</li>
```

```

21 <li>Wash and chop the salad.</li>
22 <li>Fill pita with salad, hummus, and fried halloumi.</li>
23 </ol>
24
25 <h2>Ingredient description list</h2>
26
27 <p>Paragraph for reference, paragraph for reference, paragraph for reference,
28 paragraph for reference, paragraph for reference, paragraph for reference.</p>
29
30 <dl>
31 <dt>Hummus</dt>
32 <dd>A thick dip/sauce generally made from chick peas blended with tahini, lemon juice, salt, garlic, and olive oil.</dd>
33 <dt>Pita</dt>
34 <dd>A soft, slightly leavened flatbread.</dd>
35 <dt>Halloumi</dt>
36 <dd>A semi-hard, unripened, brined cheese with a higher-than-usual melting point, usually made from goat or sheep's milk.</dd>
37 <dt>Green salad</dt>
38 <dd>That green healthy stuff that many of us just use to garnish kebabs.</dd>
39 </dl>

```

If you go to the live example now and investigate the list elements using [browser developer tools](#), you'll notice a couple of styling defaults:

- The `` and `` elements have a top and bottom `margin` of 16px (1em) and a `padding-left` of 40px (2.5em.)
- The list items (`` elements) have no set defaults for spacing.
- The `<dl>` element has a top and bottom `margin` of 16px (1em), but no padding set.
- The `<dd>` elements have `margin-left` of 40px (2.5em.)

- The `<p>` elements we've included for reference have a top and bottom `margin` of `16px` (`1em`), the same as the different list types.
-

Handling list spacing [↗](#)

When styling lists, you need to adjust their styles so they keep the same vertical spacing as their surrounding elements (such as paragraphs and images; sometimes called vertical rhythm), and the same horizontal spacing as each other (you can see the [↗ finished styled example](#) on Github, and [↗ find the source code](#) too.)

The CSS used for the text styling and spacing is as follows:

```
1  /* General styles */
2
3  html {
4      font-family: Helvetica, Arial, sans-serif;
5      font-size: 10px;
6  }
7
8  h2 {
9      font-size: 2rem;
10 }
11
12 ul,ol,dl,p {
13     font-size: 1.5rem;
14 }
```

```
15
16 li, p {
17     line-height: 1.5;
18 }
19
20 /* Description list styles */
21
22
23 dd, dt {
24     line-height: 1.5;
25 }
26
27 dt {
28     font-weight: bold;
29 }
30
31 dd {
32     margin-bottom: 1.5rem;
33 }
```

- The first rule sets a sitewide font and a baseline font size of 10px. These are inherited by everything on the page.
- Rules 2 and 3 set relative font sizes for the headings, different list types (the children of the list elements inherit these), and paragraphs. This means that each paragraph and list will have the same font size and top and bottom spacing, helping to keep the vertical rhythm consistent.
- Rule 4 sets the same `line-height` on the paragraphs and list items — so the paragraphs and each individual list item will have the same spacing between lines. This will also help to keep the vertical rhythm consistent.

- Rules 5 and 6 apply to the description list — we set the same `line-height` on the description list terms and descriptions as we did with the paragraphs and list items. Again, consistency is good! We also make the description terms have bold font, so they visually stand out easier.
-

List-specific styles [↗](#)

Now we've looked at general spacing techniques for lists, let's explore some list-specific properties. There are three properties you should know about to start with, which can be set on `` or `` elements:

- `list-style-type`: Sets the type of bullets to use for the list, for example, square or circle bullets for an unordered list, or numbers, letters or roman numerals for an ordered list.
- `list-style-position`: Sets whether the bullets appear inside the list items, or outside them before the start of each item.
- `list-style-image`: Allows you to use a custom image for the bullet, rather than a simple square or circle.

Bullet styles [↗](#)

As mentioned above, the `list-style-type` property allows you to set what type of bullet to use for the bullet points. In our example, we've set the ordered list to use uppercase roman numerals, with:

```
1 ol {  
2   list-style-type: upper-roman;  
3 }
```

This gives us the following look:

- I. Toast pitta, leave to cool, then slice down the edge.
- II. Fry the halloumi in a shallow, non-stick pan, until browned on both sides.
- III. Wash and chop the salad
- IV. Fill Pitta with salad, humous, and fried halloumi.

You can find a lot more options by checking out the [list-style-type](#) reference page.

Bullet position

The [list-style-position](#) property sets whether the bullets appear inside the list items, or outside them before the start of each item. The default value is `outside`, which causes the bullets to sit outside the list items, as seen above.

If you set the value to `inside`, the bullets will sit inside the lines:

```
1 ol {  
2   list-style-type: upper-roman;  
3   list-style-position: inside;  
4 }
```

- I. Toast pitta, leave to cool, then slice down the edge.
- II. Fry the halloumi in a shallow, non-stick pan, until browned on both sides.
- III. Wash and chop the salad
- IV. Fill Pitta with salad, humous, and fried halloumi.

Using a custom bullet image [↗](#)

The `list-style-image` property allows you to use a custom image for your bullet. The syntax is pretty simple:

```
1  ul {  
2    list-style-image: url(star.svg);  
3  }
```

However, this property is a bit limited in terms of controlling the position, size, etc. of the bullets. You are better off using the `background` family of properties, which you'll learn a lot more about in the [Styling boxes](#) module. For now, here's a taster!

In our finished example, we have styled the unordered list like so (on top of what you've already seen above):

```
1  ul {  
2    padding-left: 2rem;  
3    list-style-type: none;  
4  }  
5  
6  ul li {
```



```
7 padding-left: 2rem;
8 background-image: url(star.svg);
9 background-position: 0 0;
10 background-size: 1.6rem 1.6rem;
11 background-repeat: no-repeat;
12 }
```

Here we've done the following:

- Set the `padding-left` of the `` down from the default 40px to 20px, then set the same amount on the list items. This is so that overall the list items are still lined up with the order list items and the description list descriptions, but the list items have some padding for the background images to sit inside. If we didn't do this, the background images would overlap with the list item text, which would look messy.
- Set the `list-style-type` to `none`, so that no bullet appears by default. We're going to use `background` properties to handle the bullets instead.
- Inserted a bullet onto each unordered list item. The relevant properties are as follows:
 - `background-image`: This references the path to the image file you want to use as the bullet.
 - `background-position`: This defines where in the background of the selected element the image will appear — in this case we are saying `0 0`, which means the bullet will appear in the very top left of each list item.
 - `background-size`: This sets the size of the background image. We ideally want the bullets to be the same size as the list items (or very slightly smaller or larger). We are using a size of 1.6rem (16px), which fits very nicely with the 20px padding we've allowed for the bullet to sit inside — 16px plus 4px of space between the bullet and the list item text works well.

- `background-repeat`: By default, background images repeat until they fill up the available background space. We only want one copy of the image inserted in each case, so we set this to a value of `no-repeat`.

This gives us the following result:



list-style shorthand [↗](#)

The three properties mentioned above can all be set using a single shorthand property, `list-style`. For example, the following CSS:

```
1 ul {  
2   list-style-type: square;  
3   list-style-image: url(example.png);  
4   list-style-position: inside;  
5 }
```

Could be replaced by this:

```
1 ul {  
2   list-style: square url(example.png) inside;  
3 }
```

The values can be listed in any order, and you can use one, two or all three (the default values used for the properties that are not included are `disc`, `none`, and `outside`). If both a `type` and an `image` are specified, the type is used as a fallback if the image can't be loaded for some reason.

Controlling list counting [🔗](#)

Sometimes you might want to count differently on an ordered list — e.g. starting from a number other than 1, or counting backwards, or counting in steps of more than 1. HTML and CSS have some tools to help you here.


start [🔗](#)

The `start` attribute allows you to start the list counting from a number other than 1. The following example:

```
1 <ol start="4">
2   <li>Toast pita, leave to cool, then slice down the edge.</li>
3   <li>Fry the halloumi in a shallow, non-stick pan, until browned on both sides.</li>
4   <li>Wash and chop the salad.</li>
5   <li>Fill pita with salad, hummus, and fried halloumi.</li>
6 </ol>
```

Gives you this output:

4. Toast pita, leave to cool, then slice down the edge.
5. Fry the halloumi in a shallow, non-stick pan, until browned on both sides.
6. Wash and chop the salad.
7. Fill pita with salad, hummus, and fried halloumi.

reversed 

The `reversed` attribute will start the list counting down instead of up. The following example:

```
1 <ol start="4" reversed>
2   <li>Toast pita, leave to cool, then slice down the edge.</li>
3   <li>Fry the halloumi in a shallow, non-stick pan, until browned on both sides.</li>
4   <li>Wash and chop the salad.</li>
5   <li>Fill pita with salad, hummus, and fried halloumi.</li>
6 </ol>
```

Gives you this output:

4. Toast pita, leave to cool, then slice down the edge.
3. Fry the halloumi in a shallow, non-stick pan, until browned on both sides.
2. Wash and chop the salad.
1. Fill pita with salad, hummus, and fried halloumi.

Note: If there are more list items in a reversed list than the value of the `start` attribute, the count will continue to zero and then into negative values.


value [🔗](#)

The `value` attribute allows you to set your list items to specific numerical values. The following example:

```
1 <ol>
2   <li value="2">Toast pita, leave to cool, then slice down the edge.</li>
3   <li value="4">Fry the halloumi in a shallow, non-stick pan, until browned on both sides.</li>
4   <li value="6">Wash and chop the salad.</li>
5   <li value="8">Fill pita with salad, hummus, and fried halloumi.</li>
6 </ol>
```

Gives you this output:

2. Toast pita, leave to cool, then slice down the edge.
4. Fry the halloumi in a shallow, non-stick pan, until browned on both sides.
6. Wash and chop the salad.
8. Fill pita with salad, hummus, and fried halloumi.

 **Note:** Even if you are using a non-number `list-style-type`, you still need to use the equivalent numerical values in the `value` attribute.

Active learning: Styling a nested list

In this active learning session, we want you to take what you've learned above and have a go at styling a nested list. We've provided you with the HTML, and we want you to:

1. Give the unordered list square bullets.
2. Give the unordered list items and the ordered list items a line height of 1.5 of their font-size.
3. Give the ordered list lower alphabetical bullets.
4. Feel free to play with the list example as much as you like, experimenting with bullet types, spacing, or whatever else you can find.

If you make a mistake, you can always reset it using the *Reset* button. If you get really stuck, press the *Show solution* button to see a potential answer.

HTML Input

```
<ul>
  <li>First, light the candle.</li>
  <li>Next, open the box.</li>
  <li>Finally, place the three magic items in the box, in this exact
order, to complete the spell:
  <ol>
    <li>The book of spells</li>
    <li>The shiny rod</li>
    <li>The goblin statue</li>
  </ol>
```

CSS Input

Output

- First, light the candle.
- Next, open the box.
- Finally, place the three magic items in the box, in this exact order, to complete the spell:
 1. The book of spells
 2. The shiny rod
 3. The goblin statue

2. The sniny rod
3. The goblin statue

Reset

Show solution

See also [↗](#)

CSS counters provide advanced tools for customizing list counting and styling, but they are quite complex. We recommend looking into these if you want to stretch yourself. See:

- [@counter-style](#)
- [counter-increment](#)
- [counter-reset](#)

Summary [↗](#)

Lists are relatively easy to get the hang of styling once you know a few associated basic principles and specific properties. In the next article we'll get on to link styling techniques.

← Previous

↑ Overview: Styling text

Next →

In this module [↗](#)

- [Fundamental text and font styling](#)
 - [Styling lists](#)
 - [Styling links](#)
 - [Web fonts](#)
 - [Typesetting a community school homepage](#)
-