

---

Technologies ▼

---

References & Guides ▼

---

Feedback ▼

---

Sign in 

---

 Search

---

# Floats

Previous

Overview: CSS layout

Next

Originally for floating images inside blocks of text, the `float` property became one of the most commonly used tools for creating multiple column layouts on webpages. With the advent of Flexbox and Grid it has now returned to its original purpose, as this article explains.

**Prerequisites:** HTML basics (study [Introduction to HTML](#)), and an idea of How CSS works (study [Introduction to CSS](#).)

**Objective:** To learn how to create floated features on webpages, and to use the `clear` property and other methods of clearing floats.

---

## The background of floats

The `float` property was introduced to allow web developers to implement simple layouts involving an image floating inside a column of text, with the text wrapping around the left or right of it. The kind of thing you might get in a newspaper layout.

But web developers quickly realised that you can float anything, not just images, so the use of float broadened. Our fancy paragraph example from earlier in the course shows how you can use float in the creation of a fun drop-cap effect.

Floats have commonly been used to create entire web site layouts featuring multiple columns of information floated so they sit alongside one another (the default behaviour would be for the columns to sit below one another, in the same order as they appear in the source). There are newer, better layout techniques available and so use of floats in this way should be regarded as a legacy technique.

In this article we'll just concentrate on the proper uses of floats.

---

## A simple float example [🔗](#)

Let's explore how to use floats. We'll start with a really simple example involving floating a block of text around an element. You can follow along by creating a new `index.html` file on your computer, filling it with a simple HTML template, and inserting the below code into it at the appropriate places. At the bottom of the section you can see a live example of what the final code should look like.

First, we'll start off with some simple HTML — add the following to your HTML body, removing anything that was inside there before:

```
1 <h1>Simple float example</h1>
2
3 <div class="box">Float</div>
4
5 <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla luctus aliquam
6
7 <p>Sed auctor cursus massa at porta. Integer ligula ipsum, tristique sit amet
8
9 <p>Nam vulputate diam nec tempor bibendum. Donec luctus augue eget malesuada
```

Now apply the following CSS to your HTML (using a `<style>` element or a `<link>` to a separate `.css` file — your choice):

```
1  body {
2    width: 90%;
3    max-width: 900px;
4    margin: 0 auto;
5    font: .9em/1.2 Arial, Helvetica, sans-serif
6  }
7
8  .box {
9    width: 150px;
10   height: 100px;
11   border-radius: 5px;
12   background-color: rgb(207,232,220);
13   padding: 1em;
14 }
```

If you save and refresh now, you'll see something much like what you'd expect — the box is sitting above the text, in normal flow. To float the text around it add two properties to the `.box` rule:

```
1  .box {
2    float: left;
3    margin-right: 15px;
4    width: 150px;
5    height: 100px;
6    border-radius: 5px;
7    background-color: rgb(207,232,220);
8    padding: 1em;
9  }
```

Now if you save and refresh you'll see something like the following:

# Simple float example

Float

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla luctus aliquam dolor, eu lacinia lorem placerat vulputate.

Duis felis orci, pulvinar id metus ut, rutrum luctus orci. Cras porttitor imperdiet nunc, at ultricies tellus laoreet sit amet. Sed auctor cursus massa at porta. Integer ligula ipsum, tristique sit amet orci vel, viverra egestas ligula. Curabitur vehicula tellus neque, ac ornare ex malesuada et. In vitae convallis lacus. Aliquam erat volutpat.

Suspendisse ac imperdiet turpis. Aenean finibus sollicitudin eros pharetra congue. Duis ornare egestas augue ut luctus. Proin blandit quam nec lacus varius commodo et a urna. Ut id ornare felis, eget fermentum sapien.

Nam vulputate diam nec tempor bibendum. Donec luctus augue eget malesuada ultrices. Phasellus turpis est, posuere sit amet dapibus ut, facilisis sed est. Nam id risus quis ante semper consectetur eget aliquam lorem. Vivamus tristique elit dolor, sed pretium metus suscipit vel. Mauris ultricies lectus sed lobortis finibus. Vivamus eu urna eget velit cursus viverra quis vestibulum sem. Aliquam tincidunt eget purus in interdum. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

So let's think about how the float works — the element with the float set on it (the `<div>` element in this case) is taken out of the normal layout flow of the document and stuck to the left hand side of its parent container ( `<body>` , in this case). Any content that comes below the floated element in the normal layout flow will now wrap around it, filling up the space to the right-hand side of it as far up as the top of the floated element. There, it will stop.

Floating the content to the right has exactly the same effect, but in reverse — the floated element will stick to the right, and the content will wrap around it to the left. Try changing the float value to `right` and replace `margin-right` with `margin-left` in the last ruleset to see what the result is.

While we can add a margin to the float to push the text away, we can't add a margin to the text to move it away from the float. This is because a floated element is taken out of normal flow, and the boxes of following items actually run behind the float. You can demonstrate this by making some changes to your example.

Add a class of `special` to the first paragraph of text, the one immediately following the floated box, then in your CSS add the following rules. These will give our following paragraph a background color.

```
1 .special {  
2   background-color: rgb(79,185,227);  
3   padding: 10px;  
4   color: #fff;  
5 }
```

To make the effect easier to see, change the `margin-right` on your float to `margin`, so you get space all around the float. You will be able to see the background on the paragraph running right underneath the floated box, as in the example below.

## Simple float example

Float

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla luctus aliquam dolor, eu lacinia lorem placerat vulputate.

Duis felis orci, pulvinar id metus ut, rutrum luctus orci. Cras porttitor imperdiet nunc, at ultricies tellus laoreet sit amet. Sed auctor cursus massa at porta. Integer ligula ipsum, tristique sit amet orci vel, viverra egestas ligula. Curabitur vehicula tellus neque, ac ornare ex malesuada et. In vitae convallis lacus. Aliquam erat volutpat. Suspendisse ac imperdiet turpis. Aenean finibus sollicitudin eros pharetra congue. Duis ornare

egestas augue ut luctus. Proin blandit quam nec lacus varius commodo et a urna. Ut id ornare felis, eget fermentum sapien.

Nam vulputate diam nec tempor bibendum. Donec luctus augue eget malesuada ultrices. Phasellus turpis est, posuere sit amet dapibus ut, facilisis sed est. Nam id risus quis ante semper consectetur eget aliquam lorem. Vivamus tristique elit dolor, sed pretium metus suscipit vel. Mauris ultricies lectus sed lobortis finibus. Vivamus eu urna eget velit cursus viverra quis vestibulum sem. Aliquam tincidunt eget purus in interdum. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

The line boxes of our following element have been shortened so the text runs around the float, but due to the float being removed from normal flow the box around the paragraph still remains full width.

---

## Clearing floats

We have seen that the float is removed from normal flow and that other elements will display beside it, therefore if we want to stop a following element from moving up we need to clear it. This is achieved with the `clear` property.

In your HTML from the previous example, add a class of `cleared` to the second paragraph below the floated item. Then add the following to your CSS:

```
1 | .cleared {  
2 |     clear: left;  
3 | }
```

# Simple float example

Float

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla luctus aliquam dolor, eu lacinia lorem placerat vulputate.

Duis felis orci, pulvinar id metus ut, rutrum luctus orci. Cras porttitor imperdiet nunc, at ultricies tellus laoreet sit amet. Sed auctor cursus massa at porta. Integer ligula ipsum, tristique sit amet orci vel, viverra egestas ligula. Curabitur vehicula tellus neque, ac ornare ex malesuada et. In vitae convallis lacus. Aliquam erat volutpat. Suspendisse ac imperdiet turpis. Aenean finibus sollicitudin eros pharetra congue. Duis ornare egestas augue ut luctus. Proin blandit quam nec lacus varius commodo et a urna. Ut id ornare felis, eget fermentum sapien.

Nam vulputate diam nec tempor bibendum. Donec luctus augue eget malesuada ultrices. Phasellus turpis est, posuere sit amet dapibus ut, facilisis sed est. Nam id risus quis ante semper consectetur eget aliquam lorem. Vivamus tristique elit dolor, sed pretium metus suscipit vel. Mauris ultricies lectus sed lobortis finibus. Vivamus eu urna eget velit cursus viverra quis vestibulum sem.

You should see that the following paragraph clears the floated element and no longer comes up alongside it. The `clear` property accepts the following values:

- `left`: Clear items floated to the left.
- `right`: clear items floated to the right.
- `both`: clear any floated items, left or right.

You now know how to clear something following a floated element, but take a look at what happens if you have a tall float and a short paragraph, with a box wrapped around both elements. Change your document so that the first paragraph and our floated box are wrapped with a `<div>` with a class of `wrapper`.

```
1 | <div class="wrapper">
2 |   <div class="box">Float</div>
3 |
4 |   <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla luctus al
5 | </div>
```

In your CSS, add the following rule for the `.wrapper` class and then reload the page:

```
1 | .wrapper {
2 |   background-color: rgb(79,185,227);
3 |   padding: 10px;
4 |   color: #fff;
5 | }
```

In addition to the original `.cleared` class:

```
1 | .cleared {
2 |   clear: left;
3 | }
```

You will see that, just like in the example where we put a background image on the paragraph, the background color runs behind the float.



# Simple float example

Float

Lorem ipsum dolor sit  
amet, consectetur  
adipiscing elit. Nulla  
luctus aliquam dolor,  
eu lacinia lorem  
placerat vulputate.

Duis felis orci, pulvinar  
id metus ut, rutrum  
luctus orci. Cras  
porttitor imperdiet nunc,

at ultricies tellus laoreet sit amet. Sed auctor cursus  
massa at porta. Integer ligula ipsum, tristique sit amet orci  
vel, viverra egestas ligula. Curabitur vehicula tellus neque,  
ac ornare ex malesuada et. In vitae convallis lacus.  
Aliquam erat volutpat. Suspendisse ac imperdiet turpis.  
Aenean finibus sollicitudin eros pharetra congue. Duis  
ornare egestas augue ut luctus. Proin blandit quam nec  
lacus varius commodo et a urna. Ut id ornare felis, eget  
fermentum sapien.

Nam vulputate diam nec tempor bibendum. Donec luctus  
augue eget malesuada ultrices. Phasellus turpis est,  
posuere sit amet dapibus ut, facilisis sed est. Nam id risus  
quis ante semper consectetur eget aliquam lorem.  
Vivamus tristique elit dolor, sed pretium metus suscipit vel.  
Mauris ultricies lectus sed lobortis finibus. Vivamus eu  
urna eget velit cursus viverra quis vestibulum sem.  
Aliquam tincidunt eget cursus in interdum. Cum sociis

Once again, this is because the float has been taken out of normal flow. Clearing the following element doesn't help with this box clearing problem, where you want the bottom of the box to wrap the floated item and wrapping content even if the content is shorter. There are three potential ways to deal with this, two of which work in all browsers — yet are slightly hacky — and a third new way that deals with this situation properly.

## The clearfix hack [↗](#)

The way that this situation has traditionally been dealt with is to use something known as a "clearfix hack". This involves inserting some generated content after the box which contains the float and wrapping content, and setting that to clear both.

Add the following CSS to our example:

```
1 | .wrapper::after {  
2 |     content: "";  
3 |     clear: both;  
4 |     display: block;  
5 | }
```

Now reload the page and the box should clear. This is essentially the same as if you had added an HTML element such as a `<div>` below the items and set it to `clear: both`.

## Simple float example

Float

Lorem ipsum dolor sit  
amet, consectetur  
adipiscing elit. Nulla  
luctus aliquam dolor,  
eu lacinia lorem  
placerat vulputate.

Duis felis orci, pulvinar id metus ut, rutrum luctus orci.  
Cras porttitor imperdiet nunc, at ultricies tellus laoreet sit  
amet. Sed auctor cursus massa at porta. Integer ligula  
ipsum, tristique sit amet orci vel, viverra egestas ligula.  
Curabitur vehicula tellus neque, ac ornare ex malesuada  
et. In vitae convallis lacus. Aliquam erat volutpat.  
Suspendisse ac imperdiet turpis. Aenean finibus  
sollicitudin eros pharetra congue. Duis ornare egestas  
augue ut luctus. Proin blandit quam nec lacus varius  
commodo et a urna. Ut id ornare felis, eget fermentum  
sapien.

Nam vulputate diam nec tempor bibendum. Donec luctus  
augue eget malesuada ultrices. Phasellus turpis est,  
posuere sit amet dapibus ut, facilisis sed est. Nam id risus  
quis ante semper consectetur eget aliquam lorem.  
Vivamus tristique elit dolor, sed pretium metus suscipit vel.

## Using overflow [↗](#)

An alternative method is to set the `overflow` property of the wrapper to a value other than `visible`.

Remove the `clearfix` CSS you added in the last section, and instead add `overflow: auto` to the rules for wrapper. Once again, the box should clear.

```
1  .wrapper {  
2    background-color: rgb(79,185,227);  
3    padding: 10px;  
4    color: #fff;  
5    overflow: auto;  
6  }
```

# Simple float example

Float

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla luctus aliquam dolor, eu lacinia lorem placerat vulputate.

Duis felis orci, pulvinar id metus ut, rutrum luctus orci. Cras porttitor imperdiet nunc, at ultricies tellus laoreet sit amet. Sed auctor cursus massa at porta. Integer ligula ipsum, tristique sit amet orci vel, viverra egestas ligula. Curabitur vehicula tellus neque, ac ornare ex malesuada et. In vitae convallis lacus. Aliquam erat volutpat. Suspendisse ac imperdiet turpis. Aenean finibus sollicitudin eros pharetra congue. Duis ornare egestas augue ut luctus. Proin blandit quam nec lacus varius commodo et a urna. Ut id ornare felis, eget fermentum sapien.

Nam vulputate diam nec tempor bibendum. Donec luctus augue eget malesuada ultrices. Phasellus turpis est, posuere sit amet dapibus ut, facilisis sed est. Nam id risus quis ante semper consectetur eget aliquam lorem. Vivamus tristique elit dolor, sed pretium metus suscipit vel.

This example works by creating what is known as a **block formatting context** (BFC). This is like a mini layout inside your page, inside which everything is contained, therefore our floated element is contained inside the BFC and the background runs behind both items. This will usually work, however in certain cases you might find unwanted scrollbars or clipped shadows due to unintended consequences of using overflow.

`display: flow-root` 

The modern way of solving this problem is to use the value `flow-root` of the `display` property. This exists only to create a BFC without using hacks — there will be no unintended consequences when you use it. Remove `overflow: auto` from your `.wrapper` rule and add `display: flow-root`. Assuming you have a supporting browser, the box will clear.

```
1 | .wrapper {  
2 |     background-color: rgb(79,185,227);  
3 |     padding: 10px;  
4 |     color: #fff;  
5 |     display: flow-root;  
6 | }
```

## Simple float example

Float

Lorem ipsum dolor sit  
amet, consectetur  
adipiscing elit. Nulla  
luctus aliquam dolor,  
eu lacinia lorem  
placerat vulputate.

Duis felis orci, pulvinar id metus ut, rutrum luctus orci.  
Cras porttitor imperdiet nunc, at ultricies tellus laoreet sit  
amet. Sed auctor cursus massa at porta. Integer ligula  
ipsum, tristique sit amet orci vel, viverra egestas ligula.  
Curabitur vehicula tellus neque, ac ornare ex malesuada  
et. In vitae convallis lacus. Aliquam erat volutpat.  
Suspendisse ac imperdiet turpis. Aenean finibus  
sollicitudin eros pharetra congue. Duis ornare egestas  
augue ut luctus. Proin blandit quam nec lacus varius  
commodo et a urna. Ut id ornare felis, eget fermentum  
sapien.

Nam vulputate diam nec tempor bibendum. Donec luctus  
augue eget malesuada ultrices. Phasellus turpis est,  
posuere sit amet dapibus ut, facilisis sed est. Nam id risus  
quis ante semper consectetur eget aliquam lorem.  
Vivamus tristique elit dolor, sed pretium metus suscipit vel.

You now know all there is to know about floats in modern web development. See the article on [legacy layout methods](#) for information on how they used to be used, which may be useful if you find yourself working on older projects.

[Previous](#)[Overview: CSS layout](#)[Next](#)

---

## In this module [↗](#)

- Introduction to CSS layout
  - Normal Flow
  - Flexbox
  - Grid
  - Floats
  - Positioning
  - Multiple-column Layout
  - Legacy Layout Methods
  - Supporting older browsers
  - Fundamental Layout Comprehension Assessment
-