CSS Layout - Float Examples

w3schools.com/css/css_float_examples.asp

This page contains common float examples.

Grid of Boxes / Equal Width Boxes

Box 1

Box 2

Box 1

Box 2

Box 3

With the `float` property, it is easy to float boxes of content side by side:

Example

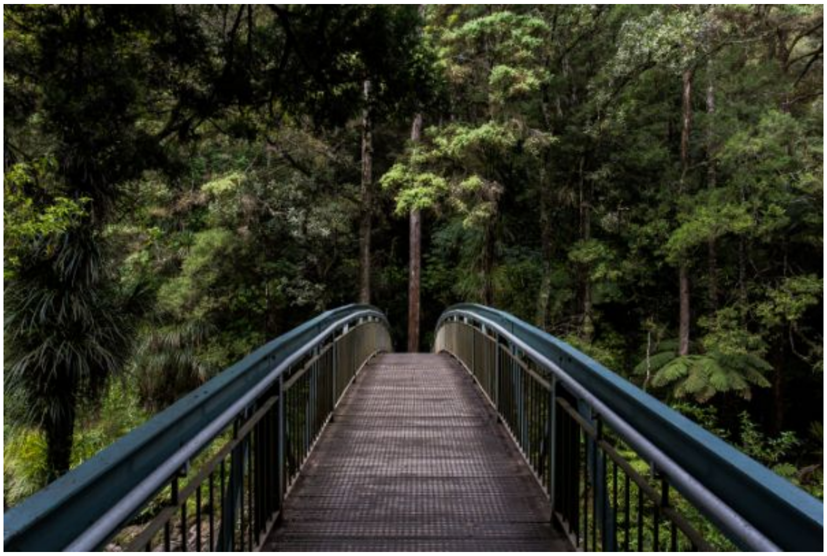
```
* {
  box-sizing: border-box;
}

.box {
  float: left;
  width: 33.33%; /* three boxes (use 25% for four, and 50% for two, etc) */
  padding: 50px; /* if you want space between the images */
}
```

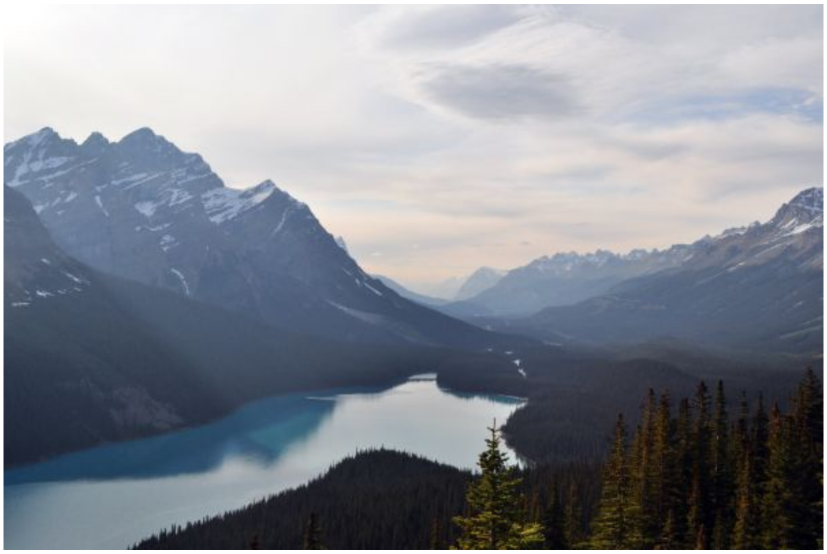
What is box-sizing?

You can easily create three floating boxes side by side. However, when you add something that enlarges the width of each box (e.g. padding or borders), the box will break. The `box-sizing` property allows us to include the padding and border in the box's total width (and height), making sure that the padding stays inside of the box and that it does not break.

You can read more about the box-sizing property in our CSS Box Sizing Chapter.

Images Side By Side





The grid of boxes can also be used to display images side by side:

Example

```
.img-container {
  float: left;
  width: 33.33%; /* three containers (use 25% for four, and 50% for two, etc) */
  padding: 5px; /* if you want space between the images */
}
```

Equal Height Boxes

In the previous example, you learned how to float boxes side by side with an equal width. However, it is not easy to create floating boxes with equal heights. A quick fix however, is to set a fixed height, like in the example below:

Box 1

Some content, some content, some content

Box 2

Some content, some content, some content

Some content, some content, some content

Some content, some content, some content

Example

```
.box {
  height: 500px;
}
```

However, this is not very flexible. It is ok if you can guarantee that the boxes will always have the same amount of content in them. But many times, the content is not the same. If you try the example above on a mobile phone, you will see that the second box's content will be displayed outside of the box. This is where CSS3 Flexbox comes in handy - as it can automatically stretch boxes to be as long as the longest box:

Example

Using **Flexbox** to create flexible boxes:

Box 1 - This is some text to make sure that the content gets really tall. This is some text to make sure that the content gets really tall. This is some text to make sure that the content gets really tall.

Box 2 - My height will follow Box 1.

Tip: You can read more about the Flexbox Layout Module in our CSS Flexbox Chapter.

Navigation Menu

You can also use `float` with a list of hyperlinks to create a horizontal menu:

Web Layout Example

It is also common to do entire web layouts using the `float` property:

Example

```
.header, .footer {
  background-color: grey;
  color: white;
  padding: 15px;
}

.column {
  float: left;
  padding: 15px;
}

.clearfix::after {
  content: "";
  clear: both;
  display: table;
}

.menu {
  width: 25%;
}

.content {
  width: 75%;
}
```

More Examples

An image with border and margins that floats to the right in a paragraph
Let an image float to the right in a paragraph. Add border and margins to the image.

An image with a caption that floats to the right
Let an image with a caption float to the right.

Let the first letter of a paragraph float to the left
Let the first letter of a paragraph float to the left and style the letter.

Creating a website with float
Use float to create a homepage with a navbar, header, footer, left content and main content.

All CSS Float Properties

Property	Description
box-sizing	Defines how the width and height of an element are calculated: should they include padding and borders, or not
clear	Specifies what should happen with the element that is next to a floating element
float	Specifies whether an element should float to the left, right, or not at all
overflow	Specifies what happens if content overflows an element's box
overflow-x	Specifies what to do with the left/right edges of the content if it overflows the element's content area
overflow-y	Specifies what to do with the top/bottom edges of the content if it overflows the element's content area