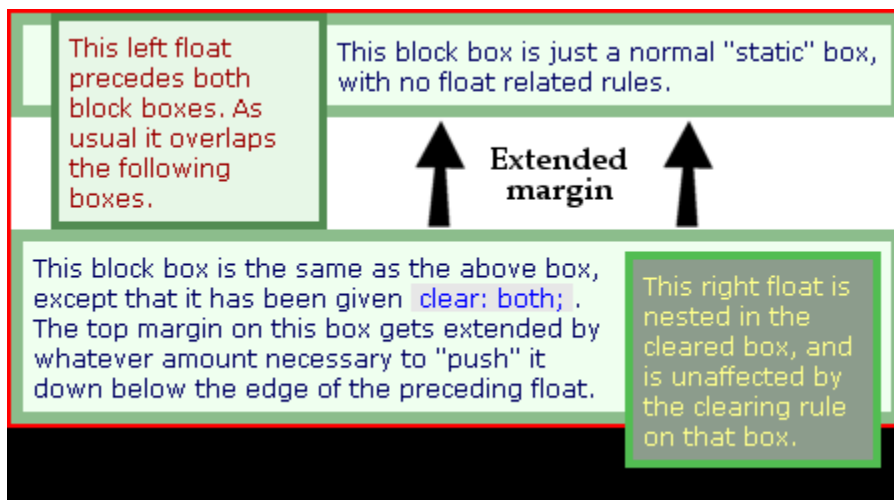


1. What is CSS clearfix, how is it used, and why is it helpful?

The CSS clearfix is a CSS technique that resolves a recurring problem that arises when two floated elements are placed adjacent to each other. This alignment causes the parent container to have a height of zero, which can disrupt the layout. Even a simple task, like positioning a sidebar to the left of the main content block, can result in overlapping and collapsing elements. Moreover, the issue varies across browsers, making it more complicated. The clearfix was introduced to address all of these problems.

Sample picture taken from the source: <https://css-tricks.com/clearfix-a-lesson-in-web-development-evolution/>



2. What are SaSS and LESS?

Currently, there are three primary CSS preprocessors available, which include Sass, Less, and Stylus. However, Sass and Less are the two most popularly utilized preprocessors. The advantage of using CSS processor is to avoid repeating ourselves. Also, preprocessors offer a distinct advantage over conventional CSS, as they enable the use of variables. This feature allows for the storage of values such as colors, fonts, or any other values that can be reused later on.

Comparing SaSS and LESS:

Sass and Less are robust CSS extensions that function more like programming languages that enhance the maintainability, theming, and extensibility of CSS. By renaming a .css file to .less or .scss, Sass and Less can effortlessly convert existing CSS files, as both preprocessors are backward compatible. It's worth noting that Less is built using JavaScript, while Sass is built using Ruby. Examples in SaSS and LESS taken from the [website](#) are as follows:

Sass font example#

```
$font-stack: Helvetica, sans-serif;
$primary-color: #333;

body {
  font: 100% $font-stack;
  color: $primary-color;
}
```

CSS output#

```
body {
  font: 100% Helvetica, sans-serif;
  color: #333;
}
```

Less color example#

```
@nice-blue: #5B83AD;
@light-blue: @nice-blue + #111;

#header {
```

```
    color: @light-blue;
}
```

CSS output#

```
#header {
    color: #6c94be;
}
```

Also, regarding the operations, we can do all types of operations in SaSS and LESS.

Another powerful feature of SaSS and LESS is that they can share properties from one selector to another.

For example: (examples taken from the website <https://www.keycdn.com/blog/sass-vs-less>)

In Sass, you use @extend

```
.message {
    border: 1px solid #ccc;
    padding: 10px;
    color: #333;
}

.success {
    @extend .message;
    border-color: green;
}

.error {
    @extend .message;
    border-color: red;
}

.warning {
    @extend .message;
    border-color: yellow;
}
```

CSS output#

```
.message, .success, .error, .warning {
    border: 1px solid #cccccc;
    padding: 10px;
    color: #333;
}

.success {
    border-color: green;
}
```

```
.error {
  border-color: red;
}

.warning {
  border-color: yellow;
}
```

Less#

In Less you use `:extend`

```
nav ul {
  &:extend(.inline);
  background: blue;
}

.inline {
  color: red;
}
```

CSS output#

```
nav ul {
  background: blue;
}

.inline, nav ul {
  color: red;
}
```

References:

<https://css-tricks.com/clearfix-a-lesson-in-web-development-evolution/>

<https://www.keycdn.com/blog/sass-vs-less>