**Floats:**

The float CSS property places an element on the left or right side of its container, allowing text and inline elements to wrap around it. The element is removed from the normal flow of the page, though still remaining part of the flow (in contrast to absolute positioning).

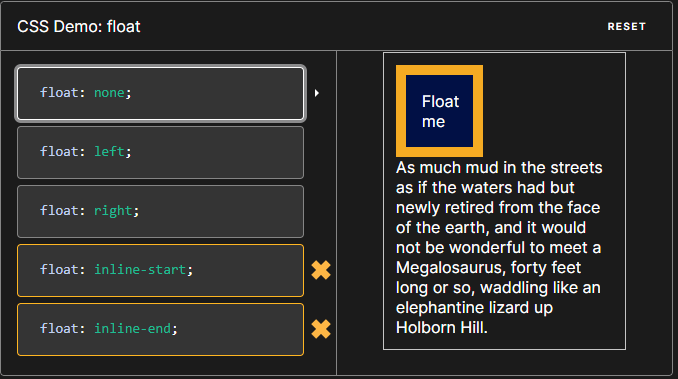
* The CSS float property specifies how an element should float.
* The CSS clear property specifies what elements can float beside the cleared element and on which side.

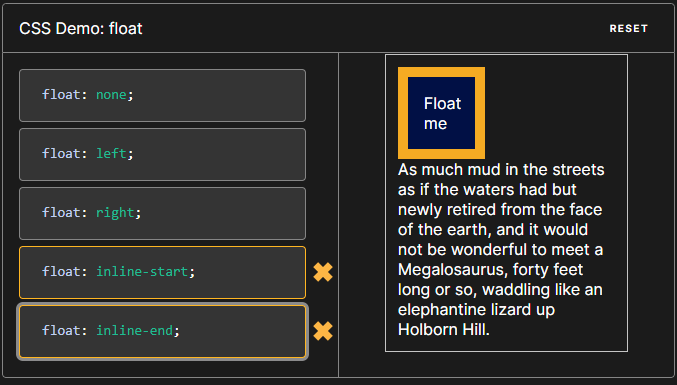
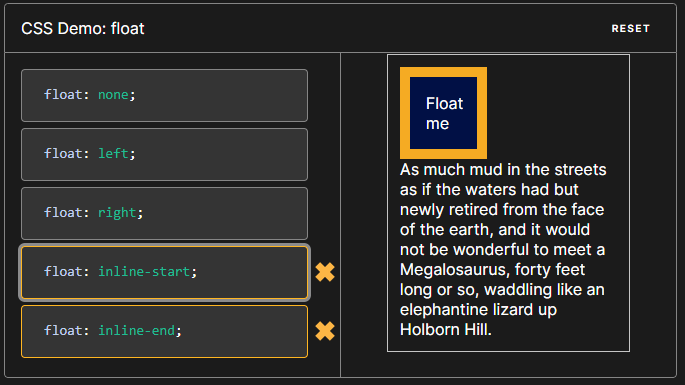
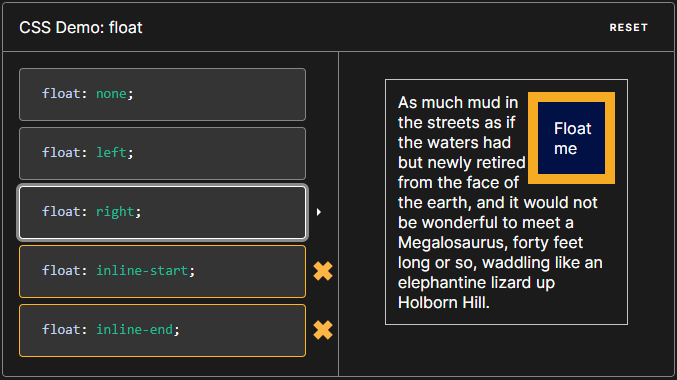
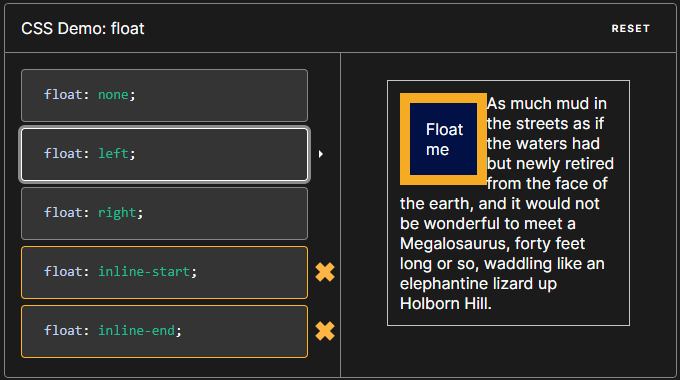
The float Property

The float property is used for positioning and formatting content e.g. let an image float left to the text in a container.

The float property can have one of the following values:

* left - The element floats to the left of its container
* right - The element floats to the right of its container
* none - The element does not float (will be displayed just where it occurs in the text). This is default
* inherit - The element inherits the float value of its parent

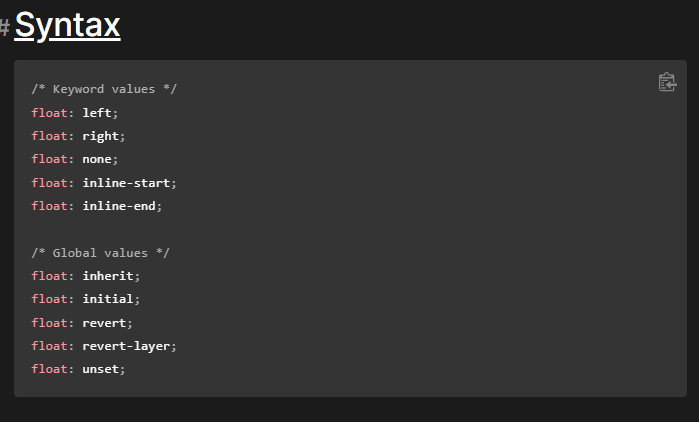




A *floating element* is one where the computed value of floats is not none.

As float implies the use of the block layout, it modifies the computed value of the display values, in some cases:

| **Specified value** | **Computed value** |
| --- | --- |
| Inline | Block |
| inline-block | Block |
| inline-table | Table |
| table-row | Block |
| table-row-group | Block |
| table-column | Block |
| table-column-group | Block |
| table-cell | Block |
| table-caption | Block |
| table-header-group | Block |
| table-footer-group | Block |
| inline-flex | Flex |
| inline-grid | Grid |
| *Other* | *Unchanged* |



The float property is specified as a single keyword, chosen from the list of values below.

Value

1. left - The element must float on the left side of its containing block.
2. Right - The element must float on the right side of its containing block.
3. None - The element must not float.
4. inline-start - The element must float on the start side of its containing block. That is the left side with ltr scripts, and the right side with rtl scripts.
5. inline-end - The element must float on the end side of its containing block. That is the right side with ltr scripts, and the left side with rtl scripts.

**Examples**

How floated elements are positioned

As mentioned above, when an element is floated, it is taken out of the normal flow of the document (though still remaining part of it). It is shifted to the left, or right, until it touches the edge of its containing box, or another floated element.

In this example, there are three coloured squares. Two are floated left, and one is floated right. Note that the second "left" square is placed to the right of the first. Additional squares would continue to stack to the right, until they filled the containing box, after which they would wrap to the next line.

A floated element is at least as tall as its tallest nested floated children. We gave the parent width: 100% and floated it to ensure it is tall enough to encompass its floated children, and to make sure it takes up the width of the parent so we don't have to clear its adjacent sibling.

HTML

<section>

<div class="left">1</div>

<div class="left">2</div>

<div class="right">3</div>

<p>

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi tristique

sapien ac erat tincidunt, sit amet dignissim lectus vulputate. Donec id

iaculis velit. Aliquam vel malesuada erat. Praesent non magna ac massa

aliquet tincidunt vel in massa. Phasellus feugiat est vel leo finibus

congue.

</p>

</section>

CSS

section {

box-sizing: border-box;

border: 1px solid blue;

width: 100%;

float: left;

}

div {

margin: 5px;

width: 50px;

height: 150px;

}

.left {

float: left;

background: pink;

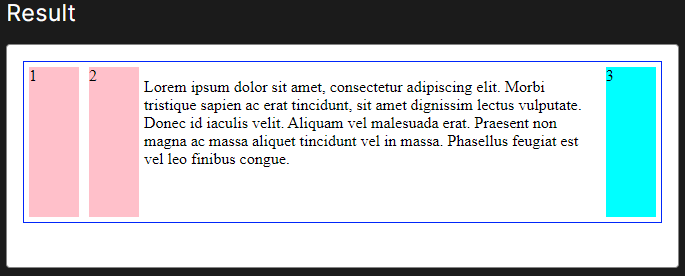
}

.right {

float: right;

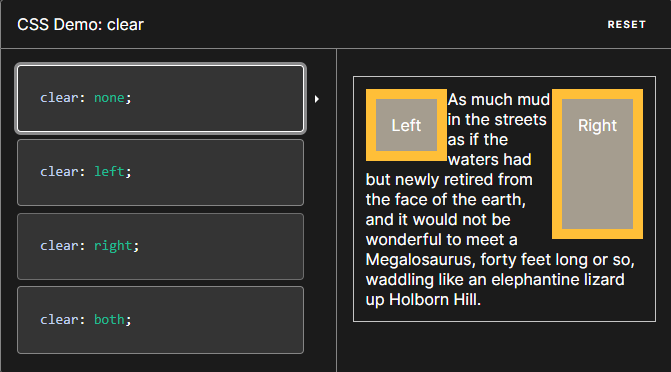
background: cyan;

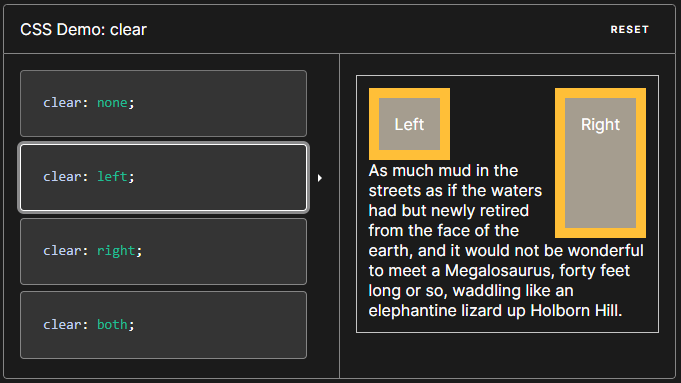
}

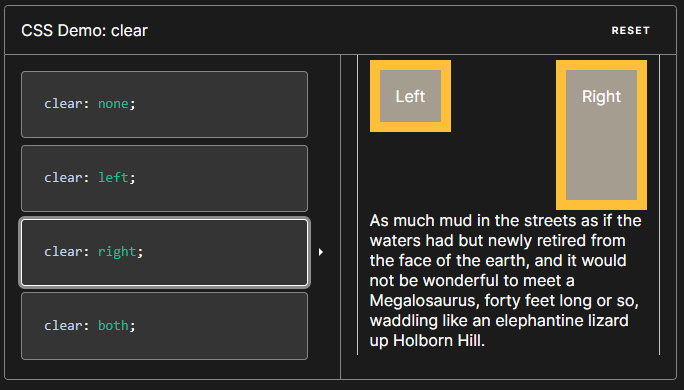


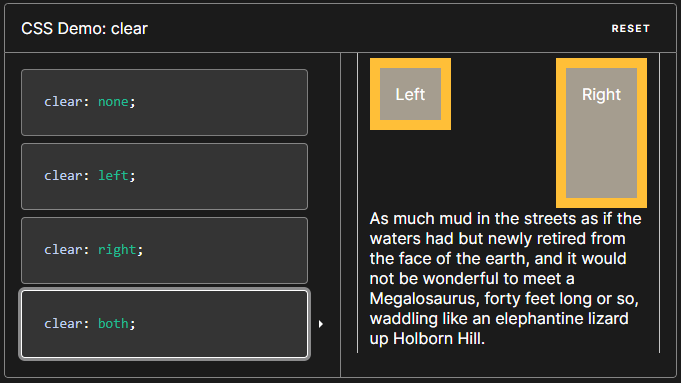
**Clear in Floats CSS layouts**

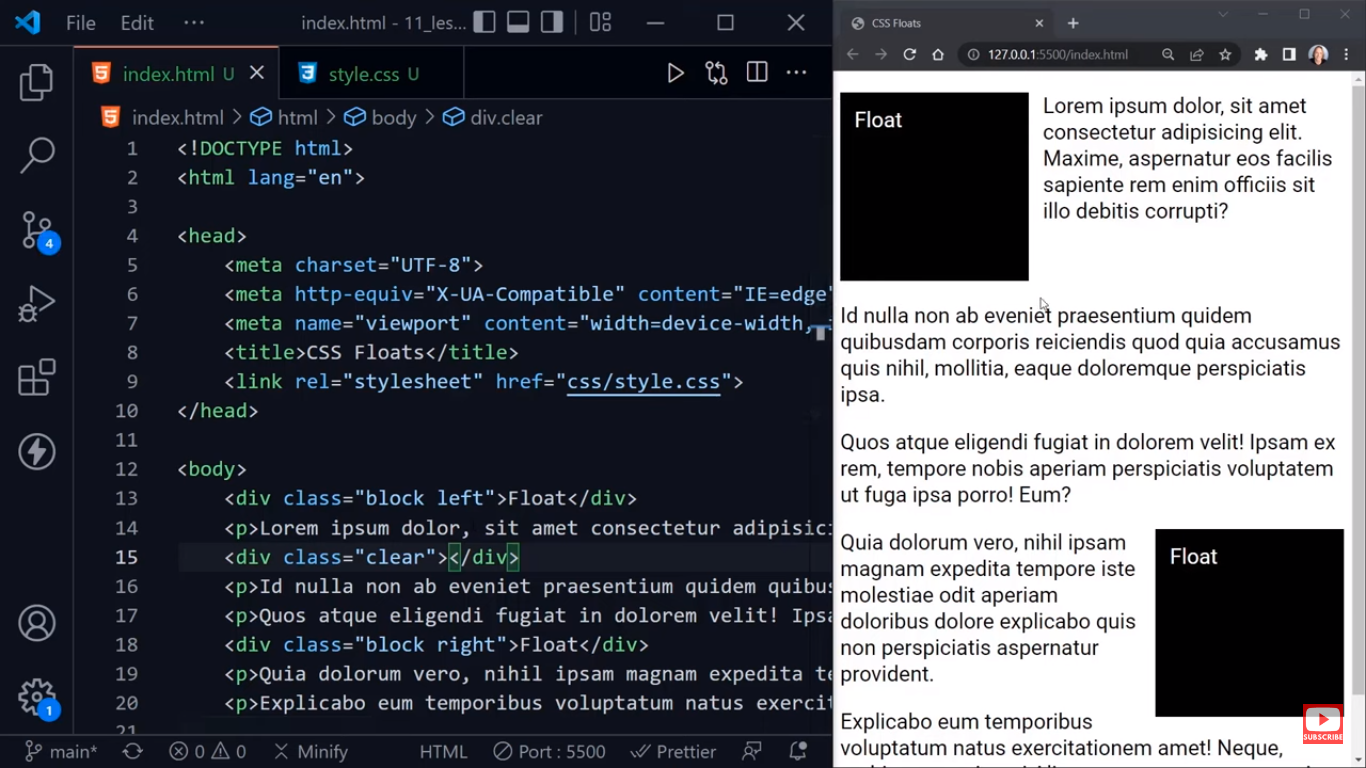
The clear CSS property sets whether an element must be moved below (cleared) floating elements that precede it. The clear property applies to floating and non-floating elements.

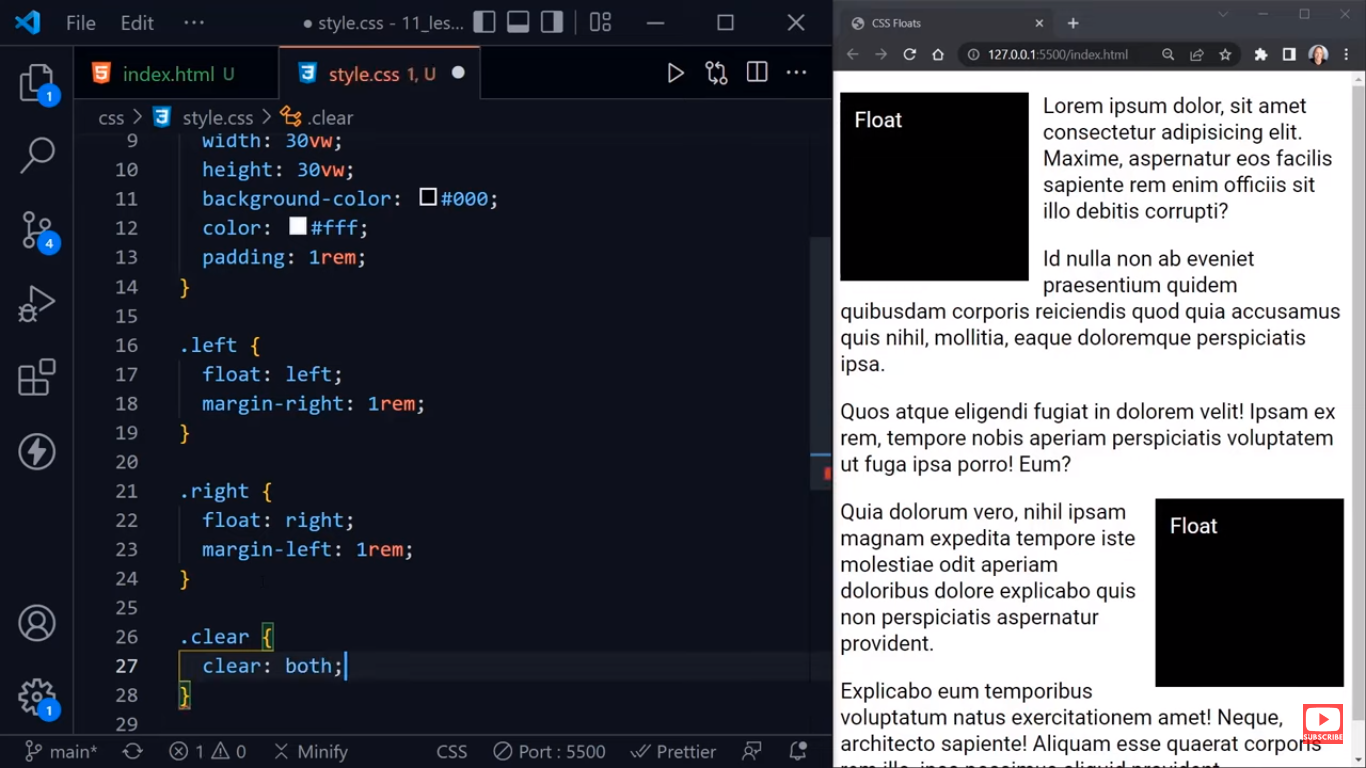












Reference for deep-down Content: <https://developer.mozilla.org/en-US/docs/Web/CSS/float>

<https://www.w3schools.com/css/css_float.asp>

**Overflow Used with Float and Clear**

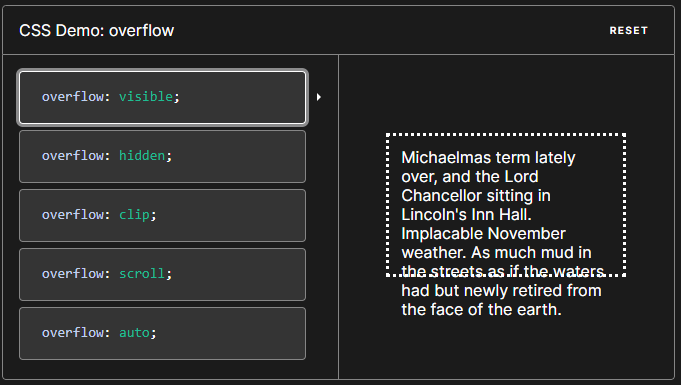
The overflow property specifies whether to clip the content or to add scrollbars when the content of an element is too big to fit in the specified area.

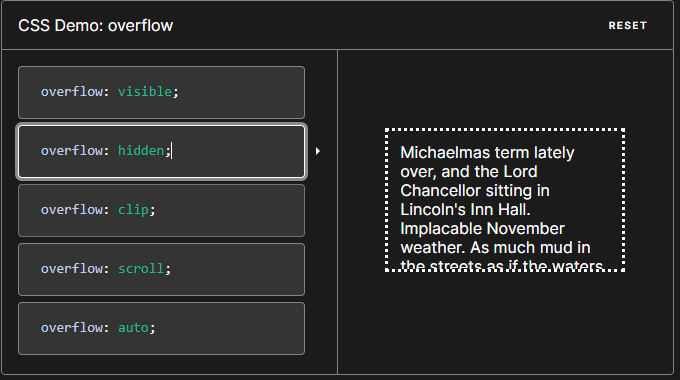
The overflow property has the following values:

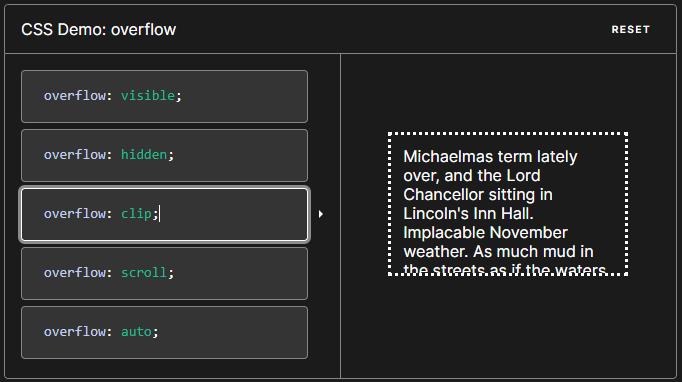
* visible - Default. The overflow is not clipped. The content renders outside the element's box
* hidden - The overflow is clipped, and the rest of the content will be invisible
* scroll - The overflow is clipped, and a scrollbar is added to see the rest of the content
* auto - Similar to scroll, but it adds scrollbars only when necessary

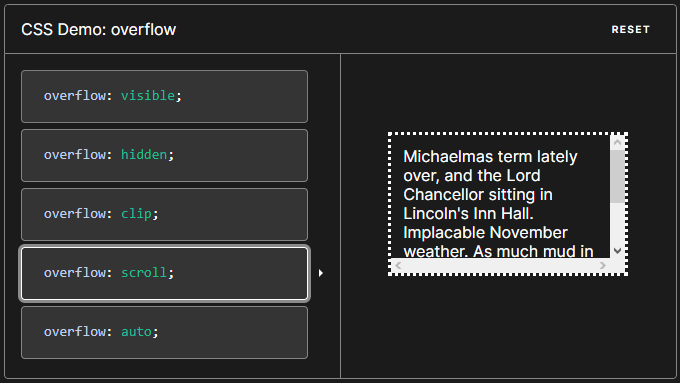
Note – overflow only works with block elements with a specified height

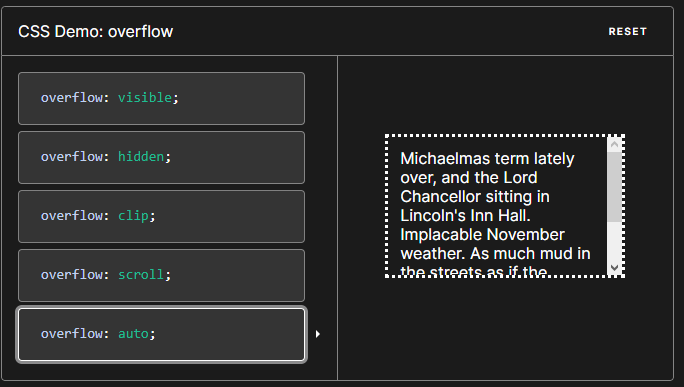
The overflow CSS shorthand property sets the desired behavior when content does not fit in the parent element box (overflows) in the horizontal and/or vertical direction

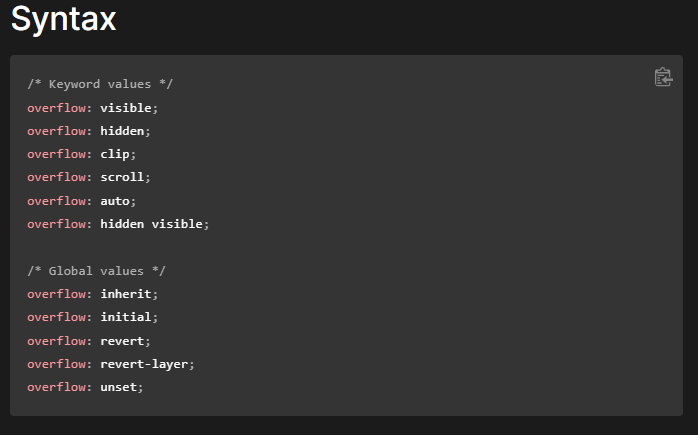












Values

* visible - Overflow content is not clipped and may be visible outside the element's padding box. The element box is not a scroll container. This is the default value of the overflow property.
* Hidden - Overflow content is clipped at the element's padding box. There are no scroll bars, and the clipped content is not visible (i.e., clipped content is hidden), but the content still exists. User agents do not add scroll bars and also do not allow users to view the content outside the clipped region by actions such as dragging on a touch screen or using the scroll wheel on a mouse. The content can be scrolled programmatically (for example, by setting the value of the scrollLeft property or the scrollTo() method), in which case. The element box is a scroll container.
* Clip - Overflow content is clipped at the element's overflow clip edge that is defined using the overflow-clip-margin property. As a result, content overflows the element's padding box by the <length> value of overflow-clip-margin or by 0px if not set. Overflow content outside the clipped region is not visible, user agents do not add a scroll bar, and programmatic scrolling is also not supported. No new formatting context is created. To establish a formatting context, use overflow: clip along with display: flow-root. The element box is not a scroll container.
* Scroll - Overflow content is clipped at the element's padding box, and overflow content can be scrolled into view using scroll bars. User agents display scroll bars in both horizontal and vertical directions if only one value is set, whether or not any content is overflowing or clipped. The use of this keyword, therefore, can prevent scroll bars from appearing and disappearing as content changes. Printers may still print overflowing content. The element box is a scroll container.
* Auto - Overflow content is clipped at the element's padding box, and overflow content can be scrolled into view. Unlike scroll, user agents display scroll bars only if the content is overflowing and hide scroll bars by default. If content fits inside the element's padding box, it looks the same as with visible but still establishes a new formatting context. The element box is a scroll container.

Code and example

<!DOCTYPE html>

<html>

<head>

<style>

.Square{

width:30vh;

height:30vh;

background-color:black;

color:white;

float:left;

margin:1em;

}

section{

background-color:yellow;

padding:1em;

margin:1em;

overflow:auto; // current way to flow overflow is to use display: flow-root ;

}

</style>

</head>

<body>

<h2>Overflow: visible</h2>

<p>By default, the overflow is visible, meaning that it is not clipped and it renders outside the element's box:</p>

<section>

<div class="Square">

Square

</div>

<div class="Text">

<p>.You can use the overflow property when you want to have better control of the layout. The overflow property specifies what happens if content overflows an element's box.

</div>

</section>

</body>

</html>

