CSS Layout - float and clear

The float property specifies whether or not an element should float.

The clear property is used to control the behavior of floating elements.

Elements after a floating element will flow around it. To avoid this, use the clear property.

The clear property specifies on which sides of an element floating elements are not allowed to float:

If an element is taller than the element containing it, and it is floated, it will overflow outside of its container.

Then we can add overflow: auto; to the containing element to fix this problem:

.clearfix {  
    overflow: auto;  
}

<div class="clearfix"><img class="img2" src="w3css.gif" alt="W3Schools.com" width="100" height="140">

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum...</div>

# CSS3 Box Sizing

The CSS3 box-sizing property allows us to include the padding and border in an element's total width and height.

box-sizing: border-box;

.div2 {  
    width: 300px;  
    height: 100px;  
    padding: 50px;  
    border: 1px solid red;  
    box-sizing: border-box;  
}

# CSS3 Flexible Box

Flexbox consists of flex containers and flex items.

A flex container is declared by setting the display property of an element to either flex (rendered as a block) or inline-flex (rendered as inline).

Inside a flex container there is one or more flex items.

**Note:** Everything outside a flex container and inside a flex item is rendered as usual. Flexbox defines how flex items are laid out inside a flex container.

Flex items are positioned inside a flex container along a flex line. By default there is only one flex line per flex container.

It is also possible to change the direction of the flex line.

If we set the direction property to rtl (right-to-left), the text is drawn right to left, and also the flex line changes direction, which will change the page layout:

## Flex Direction

The flex-direction property specifies the direction of the flexible items inside the flex container. The default value of flex-direction is row(left-to-right, top-to-bottom).

The other values are as follows:

* row-reverse - If the writing-mode (direction) is left to right, the flex items will be laid out right to left
* column - If the writing system is horizontal, the flex items will be laid out vertically
* column-reverse - Same as column, but reversed

## The justify-content Property 水平方向

The justify-content property horizontally aligns the flexible container's items when the items do not use all available space on the main-axis. （如果item没有占满空间，则用这个属性规划

The possible values are as follows:

* flex-start - Default value. Items are positioned at the beginning of the container
* flex-end - Items are positioned at the end of the container
* center - Items are positioned at the center of the container
* space-between - Items are positioned with space between the lines
* space-around - Items are positioned with space before, between, and after the lines

## The align-items Property （垂直方向）

The align-items property vertically aligns the flexible container's items when the items do not use all available space on the cross-axis.

The possible values are as follows:

* stretch - Default value. Items are stretched to fit the container
* flex-start - Items are positioned at the top of the container
* flex-end - Items are positioned at the bottom of the container
* center - Items are positioned at the center of the container (vertically)
* baseline - Items are positioned at the baseline of the container

## The flex-wrap Property 里面的item是否wrap

The flex-wrap property specifies whether the flex items should wrap or not, if there is not enough room for them on one flex line.

The possible values are as follows:

* nowrap - Default value. The flexible items will not wrap
* wrap - The flexible items will wrap if necessary
* wrap-reverse - The flexible items will wrap, if necessary, in reverse order

## The align-content Property （针对多行，因为wrap有多行了）

The align-content property modifies the behavior of the flex-wrap property. It is similar to align-items, but instead of aligning flex items, it aligns flex lines.

The possible values are as follows:

* stretch - Default value. Lines stretch to take up the remaining space
* flex-start - Lines are packed toward the start of the flex container
* flex-end - Lines are packed toward the end of the flex container
* center - Lines are packed toward the center of the flex container
* space-between - Lines are evenly distributed in the flex container
* space-around - Lines are evenly distributed in the flex container, with half-size spaces on either end

## Flex Item Properties

### Ordering

The order property specifies the order of a flexible item relative to the rest of the flexible items inside the same container:

### Margin

Setting margin: auto; will absorb extra space. It can be used to push flex items into different positions.

### Perfect Centering

In the following example we will solve an almost daily problem: perfect centering.

It is very easy with flexbox. Setting margin: auto; will make the item perfectly centered in both axis:

### align-self

The align-self property of flex items overrides the flex container's align-items property for that item. It has the same possible values as thealign-items property.

### flex

The flex property specifies the length of the flex item, relative to the rest of the flex items inside the same container.

In the following example, the first flex item will consume 2/4 of the free space, and the other two flex items will consume 1/4 of the free space each:

CSS3 中新增了一组相对于可视区域百分比的长度单位 vw, vh, vmin, vmax。其中 vw 是相对于视口宽度百分比的单位，1vw = 1% viewport width， vh 是相对于视口高度百分比的单位，1vh = 1% viewport height；vmin 是相对当前视口宽高中 较小 的一个的百分比单位，同理 vmax 是相对当前视口宽高中 较大 的一个的百分比单位。该单位浏览器兼容性如下：