

# Text

Example: [CSS text.html](#)

## !important Rule

- The `!important` rule is used to give the value of a specific property the highest priority.
- The `!important` rule will override ALL previous styling rules for that specific property on that element!

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
    background-color: yellow !important;
}

#myid {
    background-color: blue;
}

.myclass {
    background-color: gray;
}

</style>
</head>
<body>

<h2>Using the !important Rule</h2>

<p style="background-color:orange;">This is a paragraph.</p>

<p class="myclass">This is a paragraph.</p>

<p id="myid">This is a paragraph.</p>

</body>
</html>
```

# Using the !important Rule

This is a paragraph.

This is a paragraph.

This is a paragraph.

# Display: block, inline block

## CSS display block inline

The display property allows you to turn an inline element into a block-level element or vice versa, and can also be used to hide an element from the page. The values this property can take are:

## inline

(like `<span>`, `<a>`, `<button>`, `<input>`, `<img>`)

- sit next to other elements on the same line and only take the space they need.
- Only take up as much space as the text/content inside them.
- **Ignore height and width** - you cannot set those values.

## Block

(like `<div>`, `<p>`, `<h1>`–`<h6>`, `<header>`, `<footer>`)

- always start on a new line and occupy the full available width.
- You can **set width and height**, like `width: 200px; height: 200px`.

# Changing Display Behavior

We can override the default behavior using the `display` property.

## 1. `display: block`

If we set a `<span>` to `display: block`:

- It becomes a block-level element.
- It moves to a new line.
- We **can now apply width and height**.

Setting `display: block` on a `<div>` changes nothing, since it's already block-level.

## 2. `display: inline`

If we change block elements to `display: inline`:

- They behave like inline elements.
- They sit next to other inline content.

- Width and height no longer apply.

### 3. `display: inline-block`

`inline-block` combines the best of both:

- **Sits inline** like a span
- **Can have width and height** like a block element

This is very useful for creating small “card-like” elements aligned side-by-side.

### `display:none`

This **hides** an element from the page. In this case, the element acts as though it is not on the page at all (although a user could still see the content of the box if they used the view source option in their browser). Other elements shift to fill the gap.

### `visibility: hidden`

The element becomes invisible. But it still takes up space in the layout. The layout does not change.

## Overflow

[CSS Overflow.html](#)

The overflow property tells the browser what to do if the content contained within a box is larger than the box itself.

**visible** ( Default ) - content spills outside

**Hidden** - This property simply hides any extra content that does not fit in the box.

**Scroll** - This property adds a scrollbar to the box so that users can scroll to see the missing content.

**Auto** - Smart → shows scrollbars ONLY when needed. **Most Used**

### Axis-Specific Control:

```
overflow-x: auto; /* Horizontal overflow only */
overflow-y: hidden; /* Vertical overflow only */
```

# Float

## CSS Float

The float property allows you to take an element in normal flow and place it as far to the left or right of the containing element as possible. Anything else that sits inside the containing element will flow around the element that is floated.

When you use the float property, you should also use the width property to indicate how wide the floated element should be. If you do not, results can be inconsistent but the box is likely to take up the full width of the containing element (just like it would in normal flow).

## Using Float to Place Elements Side-by-Side

A lot of layout place boxes next to each other. The float property is commonly used to achieve this. When elements are floated, the height of the boxes can affect where the following elements sit.

The fourth paragraph does not go across to the left hand edge of the page as one might expect. Rather it sits right under the third paragraph.

The reason for this is that the fourth paragraph has space to start under the third paragraph, but it cannot go any further to the left because the second paragraph is in the way.

In 1817 Baron von Drais invented a walking machine that would help him get around. Lorem ipsum dolor sit amet consectetur adipisicing elit. Ex praesentium ducimus, vitae qui autem molestias totam. Illo tenetur mollitia omnis.

In 1839, Kirkpatrick Macmillan, a Scottish blacksmith, added pedals to the front wheel of a bicycle... Lorem, ipsum dolor sit amet consectetur adipisicing elit. Et nulla illo obcaecati commodi repellat voluptate, officia placeat, quis impedit laudantium tempora, necessitatibus quos. Minima aperiam modi quibusdam, soluta doloribus illo!

By the 1860s, bicycles had become a popular mode of transportation... Lorem ipsum dolor sit amet consectetur adipisicing elit. Quisquam, quidem.

In the 1880s, the safety bicycle was introduced... Lorem ipsum dolor sit amet consectetur adipisicing elit. Quisquam, quidem. Lorem ipsum dolor sit amet consectetur adipisicing elit. Necessitatibus quo placeat commodi voluptatibus, quas numquam cum nemo est perferendis. Temporibus unde architecto laboriosam repellat, laudantium dolores mollitia corrupti quasi dolor.

Today, bicycles are used worldwide for transportation, recreation, and sport... Lorem ipsum dolor sit amet consectetur adipisicing elit. Quisquam, quidem. Lorem ipsum dolor sit amet consectetur adipisicing elit. Necessitatibus quo placeat commodi voluptatibus, quas numquam cum nemo est perferendis. Temporibus unde architecto laboriosam repellat, laudantium dolores mollitia corrupti quasi dolor.

From the early designs of Baron von Drais to the modern bicycles we see today, the evolution of the bicycle has been marked by innovation and a commitment to improving human mobility... Lorem ipsum dolor sit amet consectetur adipisicing elit. Quisquam, quidem. Lorem ipsum dolor sit amet consectetur adipisicing elit. Necessitatibus quo placeat commodi voluptatibus, quas numquam cum nemo est perferendis. Temporibus unde architecto laboriosam repellat, laudantium dolores mollitia corrupti quasi dolor.

# Clearing Floats

Start the next element below all previous floats

The clear property allows you to say that no element (within the same containing element) should touch the left or right-hand sides of a box. It can take the following values:

```
clear: left;      /* clear floats floated left */  
clear: right;     /* clear floats floated right */  
clear: both;      /* clear all floats */
```

In 1817 Baron von Drais invented a walking machine that would help him get around. Lorem ipsum dolor sit amet consectetur adipisicing elit. Ex praesentium ducimus, vitae qui autem molestias totam. Illo tenetur mollitia omnis.

In 1839, Kirkpatrick Macmillan, a Scottish blacksmith, added pedals to the front wheel of a bicycle... Lorem, ipsum dolor sit amet consectetur adipisicing elit. Et nulla illo obcaecati commodi repellat voluptate, officia placeat, quis impedit laudantium tempora, necessitatibus quos. Minima aperiam modi quibusdam, soluta doloribus illo!

By the 1860s, bicycles had become a popular mode of transportation... Lorem ipsum dolor sit amet consectetur adipisicing elit. Quisquam, quidem.

In the 1880s, the safety bicycle was introduced... Lorem ipsum dolor sit amet consectetur adipisicing elit. Quisquam, quidem. Lorem ipsum dolor sit amet consectetur adipisicing elit. Necessitatibus quo placeat commodi voluptatibus, quas numquam cum nemo est perferendis. Temporibus unde architecto laboriosam repellat, laudantium dolores mollitia corrupti quasi dolor.

Today, bicycles are used worldwide for transportation, recreation, and sport... Lorem ipsum dolor sit amet consectetur adipisicing elit. Quisquam, quidem. Lorem ipsum dolor sit amet consectetur adipisicing elit. Necessitatibus quo placeat commodi voluptatibus, quas numquam cum nemo est perferendis. Temporibus unde architecto laboriosam repellat, laudantium dolores mollitia corrupti quasi dolor.

From the early designs of Baron von Drais to the modern bicycles we see today, the evolution of the bicycle has been marked by innovation and a commitment to improving human mobility... Lorem ipsum dolor sit amet consectetur adipisicing elit. Quisquam, quidem. Lorem ipsum dolor sit amet consectetur adipisicing elit. Necessitatibus quo placeat commodi voluptatibus, quas numquam cum nemo est perferendis. Temporibus unde architecto laboriosam repellat, laudantium dolores mollitia corrupti quasi dolor.

In this example, the fourth paragraph has a class called clear. The CSS rule for this class uses the clear property to indicate that nothing should touch the left-hand side of it.

## Parents of Floated Elements: Problem

```
div {  
border: 1px solid #665544;  
overflow: auto;}
```

## Characteristics of `float` in CSS

### 1. Floated elements are removed from normal document flow

- They are taken out of the regular layout flow.
- Other block elements behave as if the floated element is not there.
- This is why the `body height does not expand` around floated elements unless you clear them.

### 2. Inline content (text, inline elements) wraps around floated elements

- Text will naturally flow around the float.
- This is the *original purpose* of floats — wrapping text around images (like newspapers).

### 3. Floated elements stay “anchored” to the left or right

- `float: left;` sticks the element to the left, moving inline text to its right.
- `float: right;` sticks the element to the right, moving text to its left.

### 4. Floated elements shrink to their content width (unless width is set)

### 5. Floats allow multiple floated elements to sit side-by-side

## Flexbox

[CSS Flexbox Ultimate Guide](#)