<length>

^ CSS: Cascading Style Sheets

English (US) ▾

Change language

▲ Jump to section

The `<length>` CSS data type represents a distance value. Lengths can be used in numerous CSS properties, such as `width`, `height`, `margin`, `padding`, `border-width`, `font-size`, and `text-shadow`.

> **Note:** Although `<percentage>` values are also CSS dimensions, and are usable in some of the same properties that accept `<length>` values, they are not themselves `<length>` values. See `<length-percentage>`.

Syntax

The `<length>` data type consists of a `<number>` followed by one of the units listed below. As with all CSS dimensions, there is no space between the unit literal and the number. The length unit is optional after the number `0`.

> **Note:** Some properties allow negative `<length>`s, while others do not.

Units

Relative length units

Relative lengths represent a measurement in terms of some other distance. Depending on the unit, this can be the size of a specific character, the line height, or the size of theviewport.

Font-relative lengths

Font-relative lengths define the `<length>` value in terms of the size of a particular character or font attribute in the font currently in effect in an element or its parent.

> **Note:** These units, especially `em` and `rem`, are often used to create scalable layouts, which maintain the vertical rhythm of the page even when the user changes the font size.

`cap` ⚗

Represents the "cap height" (nominal height of capital letters) of the element's `font`.

`ch`

Represents the width, or more precisely the advance measure, of the glyph "0" (zero, the Unicode character U+0030) in the element's `font`.

In the cases where it is impossible or impractical to determine the measure of the “0” glyph, it must be assumed to be 0.5em wide by 1em tall.

`em`

Represents the calculated `font-size` of the element. If used on the `font-size` property itself, it represents the *inherited* font-size of the element.

`ex`

Represents the x-height of the element's `font`. On fonts with the "x" letter, this is generally the height of lowercase letters in the font; `1ex ≈ 0.5em` in many fonts.

`ic` 🧪

Equal to the used advance measure of the "水" glyph (CJK water ideograph, U+6C34), found in the font used to render it.

`lh` 🧪

Equal to the computed value of the `line-height` property of the element on which it is used, converted to an absolute length.

`rem`

Represents the `font-size` of the root element (typically `<html>`). When used within the root element `font-size`, it represents its initial value (a common browser default is `16px`, but user-defined preferences may modify this).

`rlh` 🧪

Equal to the computed value of the `line-height` property on the root element (typically `<html>`), converted to an absolute length. When used on the `font-size` or `line-height` properties of the root element, it refers to the properties' initial value.

Viewport-percentage lengths

Viewport-percentage lengths define the `<length>` value relative to the size of theviewport, i.e., the visible portion of the document. Viewport lengths are invalid in `@page` declaration blocks.

`vh`

Equal to 1% of the height of the viewport's initialcontaining block.

`vw`

Equal to 1% of the width of the viewport's initialcontaining block.

`vi` 🧪

Equal to 1% of the size of the initialcontaining block, in the direction of the root element’s inline axis.

`vb` 🧪

Equal to 1% of the size of the initialcontaining block, in the direction of the root element’s block axis.

`vmin`

Equal to the smaller of `vw` and `vh`.

`vmax`

Equal to the larger of `vw` and `vh`.

Absolute length units

Absolute length units represent a physical measurement when the physical properties of the output medium are known, such as for print layout. This is done by anchoring one of the units to a physical unit, and then defining the others relative to it. The anchor is done differently for low-resolution devices, such as screens, versus high-resolution devices, such as printers.

For low-dpi devices, the unit `px` represents the physical *reference pixel*; other units are defined relative to it. Thus, `1in` is defined as `96px`, which equals `72pt`. The consequence of this definition is that on such devices, dimensions described in inches (`in`), centimeters (`cm`), or millimeters (`mm`) don't necessary match the size of the physical unit with the same name.

For high-dpi devices, inches (`in`), centimeters (`cm`), and millimeters (`mm`) are the same as their physical counterparts. Therefore, the `px` unit is defined relative to them (1/96 of 1 inch).

> **Note:** Many users increase theiruser agent's default font size to make text more legible. Absolute lengths can cause accessibility problems, since they are fixed and do not scale according to user settings. For this reason, prefer relative lengths (such as `em` or `rem`) when setting `font-size`.

`px`

One pixel. For screen displays, it traditionally represents one device pixel (dot). However, for *printers* and *high-resolution screens*, one CSS pixel implies multiple device pixels. `1px` = 1/96th of `1in`.

`cm`

One centimeter. `1cm` = `96px/2.54`.

`mm`

One millimeter. `1mm` = 1/10th of `1cm`.

`Q` 🧪

One quarter of a millimeter. `1Q` = 1/40th of `1cm`.

`in`

One inch. `1in` = `2.54cm` = `96px`.

`pc`

One pica. `1pc` = `12pt` = 1/6th of `1in`.

`pt`

One point. `1pt` = 1/72nd of `1in`.

`mozmm` ⚠ **, removed in Firefox 59**

An experimental unit that attempts to render at exactly one millimeter regardless of the size or resolution of the display. This is rarely actually what you want, but may be useful in particular for mobile devices.

Interpolation

When animated, values of the `<length>` data type are interpolated as real, floating-point numbers. The interpolation happens on the calculated value. The speed of the interpolation is determined by the timing function associated with the animation.

Examples

Length unit comparison

The following demo provides you with an input field in which you can enter a `<length>` value (e.g. `300px`, `50%`, `30vw`) to set the width of a result bar that will appear below it once you've pressed **Return**.

This allows you to compare and contrast the effect of different length units.

HTML

```
<div class="outer">
  <div class="input-container">
    <label>Enter width:</label>
    <input type="text" id="length">
  </div>
  <div class="inner">

  </div>
</div>
<div class="results">
</div>
```

CSS

```css
html {
  font-family: sans-serif;
  font-weight: bold;
  box-sizing: border-box;
}

.outer {
  width: 100%;
  height: 50px;
  background-color: #eee;
  position: relative;
}

.inner {
  height: 50px;
  background-color: #999;
  box-shadow: inset 3px 3px 5px rgba(255,255,255,0.5),
              inset -3px -3px 5px rgba(0,0,0,0.5);
}

.result {
  height: 20px;
  background-color: #999;
  box-shadow: inset 3px 3px 5px rgba(255,255,255,0.5),
              inset -3px -3px 5px rgba(0,0,0,0.5);
  background-color: orange;
  display: flex;
  align-items: center;
  margin-top: 10px;
}

.result code {
  position: absolute;
  margin-left: 20px;
}

.results {
  margin-top: 10px;
}

.input-container {
  position: absolute;
  display: flex;
  justify-content: flex-start;
  align-items: center;
  height: 50px;
}

label {
  margin: 0 10px 0 20px;
}
```

JavaScript

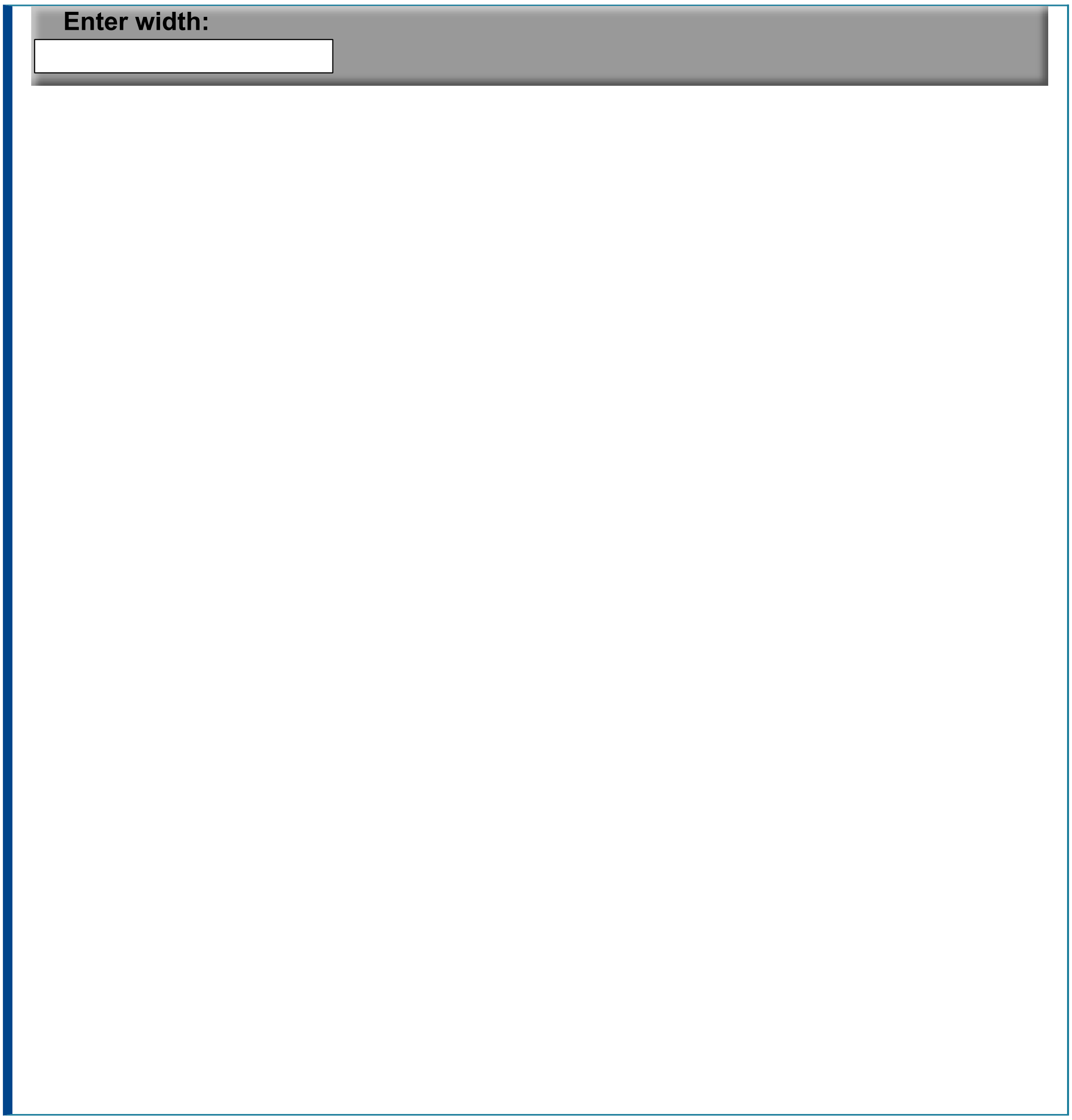
```
const inputDiv = document.querySelector('.inner');
const inputElem = document.querySelector('input');
const resultsDiv = document.querySelector('.results');

inputElem.addEventListener('change', () => {
  inputDiv.style.width = inputElem.value;

  const result = document.createElement('div');
  result.className = 'result';
  result.style.width = inputElem.value;
  result.innerHTML = `<code>width: ${inputElem.value}</code>`;
  resultsDiv.appendChild(result);

  inputElem.value = '';
  inputElem.focus();
})
```

Result



Enter width:

Specifications

Specification	Status	Comment

Specification	Status	Comment
CSS Values and Units Module Level 4 The definition of '<length>' in that specification.	ED Editor's Draft	Adds the `vi`, `vb`, `ic`, `lh`, and `rlh` units.
CSS Values and Units Module Level 3 The definition of '<length>' in that specification.	CR Candidate Recommendation	Adds the `ch`, `rem`, `vw`, `vh`, `vmin`, `vmax`, and `Q` units.
CSS Level 2 (Revision 1) The definition of '<length>' in that specification.	REC Recommendation	Explicit definition of the `em`, `pt`, `pc`, and `px` units.
CSS Level 1 The definition of '<length>' in that specification.	REC Recommendation	Initial definition. Implicit definition of the `em`, `pt`, `pc`, and `px` units.

Browser compatibility

BCD tables only load in the browser

See also

- CSS values & units tutorial
- CSS values & units reference
- Box Model

Related Topics

CSS

CSS Reference

CSS Types

▼ Properties

`<angle-percentage>`

`<angle>`

`<basic-shape>`

`<blend-mode>`

`color`

`<custom-ident>`

`<dimension>`

`flex`

`<frequency-percentage>`

`<frequency>`

`<gradient>`

`ident`

`<image>`

`<integer>`

`<length-percentage>`

`<length>`

`<number>`

`<percentage>`

`position`

`<ratio>`

`<resolution>`

👎 `<shape>`

`<string>`

`<time-percentage>`

`<time>`

`<transform-function>`

Found a problem with this page?

- Source on **GitHub**
- Report a problem with this content on **GitHub**
- Want to fix the problem yourself? See our Contribution guide.

Last modified: Dec 18, 2020, by MDN contributors

MDN Web Docs
moz://a

Web Technologies

Learn Web Development

About MDN

Feedback

About

MDN Web Docs Store

Contact Us

Firefox

MDN

Mozilla

Terms Privacy Cookies