CSS values and units

Every CSS declaration includes a property / value pair. Depending on the property, the value can include a single integer or keyword, to a series of keywords and values with or without units. There are a common set of data types — values and units — that CSS properties accept. Below is an overview of most of these data types. Refer to the page for each value type for more detailed information.

Textual data types

- <custom-ident>
- Pre-defined keywords as an <ident>
- <string>
- url()

Text data types are either <string> , a quoted series of characters, or an <ident> , a "CSS Identifier" which is an unquoted string. A <string> must be quoted with either single or double quotes. CSS Identifiers, listed in the specifications as <ident> or <custom-ident> , must be unquoted.

In the CSS specifications, values that can be defined by the web developer, like keyframe animations, font-family names, or grid areas are listed as a <custom-ident> , <string> , or both.

When both quoted and unquoted user defined text values are permitted, the specification will list <custom-ident> | <string> , meaning quotes are optional, such as is the case with animation names:

```
@keyframe validIdent {
  /* keyframes go here */
}
@keyframe 'validString' {
  /* keyframes go here */
}
```

Some text values are not valid if encompassed in quotes. For example, the value of grid-area can be a <custom-ident> , so if we had a grid area named content we would use it without quotes:

```
.item {
  grid-area: content;
}
```

In comparison, a data type that is a <string> , such as a string value of the content property, must be quoted:

```
.item::after {
  content: "This is my content.";
}
```

While you can generally create any name you want, including using emojis, the identifier can't be none , unset , initial , or inherit , start with a digit or two dashes, and generally you don't want it to be any other pre-defined CSS keyword. See the <custom-ident> and <string> reference pages for more details.

Pre-defined keyword values

Pre-defined keywords are text values defined by the specification for that property. These keywords are also CSS Identifiers and are therefore used without quotes.

When viewing CSS property value syntax in a CSS specification or the MDN property page, allowable keywords will be listed in the following form. The following values are the pre-defined keyword values allowed for float .

```
left | right | none | inline-start | inline-end
```

Such values are used without quotes:

```
.box {
  float: left;
}
```

CSS-wide values

In addition to the pre-defined keywords that are part of the specification for a property, all CSS properties accept the CSS-wide property values initial , inherit , and unset , which explicitly specify defaulting behaviors.

The initial keyword represents the value specified as the property's initial value.
The inherit keyword represents the computed value of the property on the element's parent, provided it is inherited.

The unset keyword acts as either inherit or initial , depending on whether the property is inherited or not.

A fourth value of revert was added in the Cascade Level 4 specification, but it does not currently have good browser support.

URLs

A url() type uses functional notation, which accepts a <string> that is a URL. This may be an absolute URL or a relative URL. For example, if you wanted to include a background image, you might use either of the following.

```
.box {
  background-image: url("images/my-background.png");
}

.box {
  background-image: url("https://www.exammple.com/images/my-background.png");
}
```

The parameter for `url()` can be either quoted or unquoted. If unquoted, it is parsed as a `<url-token>`, which has extra requirements including the escaping of certain characters. See `url()` for more information.

Numeric data types

- `<integer>`
- `<number>`
- `<dimension>`
- `<percentage>`

Integers

An integer is one or more decimal digits, `0` through `9`, such as `1024` or `-55`. An integer may be preceded by a `+` or `-` symbol, with no space between the symbol and the integer.

Numbers

A `<number>` represents a real number, which may or may not have a decimal point with a fractional component, for example `0.255`, `128` or `-1.2`. Numbers may also be preceded by a `+` or `-` symbol.

Dimensions

A `<dimension>` is a `<number>` with a unit attached to it, for example `45deg`, `100ms`, or `10px`. The attached unit identifier is case insensitive. There is never a space or any other characters between a the number and the unit identifier: i.e. `1 cm` is not valid.

CSS uses dimensions to specify:

- `<length>` (Distance units)
- `<angle>`
- `<time>`
- `<frequency>`
- `<flex>`
- `<resolution>`

These are all covered in subsections below.

Distance units

Where a distance unit, also known as a length, is allowed as a value for a property, this is described as the <length> type. There are two types of lengths in CSS: relative and absolute.

Relative length units specify a length in relation to something else. For example, em is relative to the font size on the element and vh is relative to the viewport height.

Unit	Relative to
em	Font size of the element.
ex	x-height of the element's font.
cap	Cap height (the nominal height of capital letters) of the element's font.
ch	Average character advance of a narrow glyph in the element's font, as represented by the "0" (ZERO, U+0030) glyph.
ic	Average character advance of a full width glyph in the element's font, as represented by the "水" (CJK water ideograph, U+6C34) glyph.
rem	Font size of the root element.
lh	Line height of the element.
rlh	Line height of the root element.
vw	1% of viewport's width.
vh	1% of viewport's height.
vi	1% of viewport's size in the root element's inline axis.
vb	1% of viewport's size in the root element's block axis.
vmin	1% of viewport's smaller dimension.
vmax	1% of viewport's larger dimension.

Container query length units specify a length relative to the dimensions of a query container. For example, cqw is relative to the width of the query container and cqh is relative to the height of the query container.

Unit	Relative to
cqw	1% of a query container's width
cqh	1% of a query container's height
cqi	1% of a query container's inline size
cqb	1% of a query container's block size
cqmin	The smaller value of cqi or cqb

Unit	Relative to
cqmax	The larger value of cqi or cqb

Absolute length units are fixed to a physical length: either an inch or a centimeter. Many of these units are therefore more useful when the output is a fixed size media, such as print. For example, mm is a physical millimeter, 1/10th of a centimeter.

Unit	Name	Equivalent to
cm	Centimeters	1cm = 96px/2.54
mm	Millimeters	1mm = 1/10th of 1cm
Q	Quarter-millimeters	1Q = 1/40th of 1cm
in	Inches	1in = 2.54cm = 96px
pc	Picas	1pc = 1/6th of 1in
pt	Points	1pt = 1/72th of 1in
px	Pixels	1px = 1/96th of 1in

When including a length value, if the length is 0 , the unit identifier is not required. Otherwise, the unit identifier is required, is case insensitive, and must come immediately after the numeric part of the value, with no space in-between.

Angle units

Angle values are represented by the type <angle> and accept the following values:

Unit	Name	Description
deg	Degrees	There are 360 degrees in a full circle.
grad	Gradians	There are 400 gradians in a full circle.
rad	Radians	There are 2π radians in a full circle.
turn	Turns	There is 1 turn in a full circle.

Time units

Time values are represented by the type <time> . When including a time value, the unit identifier — the s or ms — is required. It accepts the following values.

Unit	Name	Description
s	Seconds	

Unit	Name	Description
ms	Milliseconds	There are 1,000 milliseconds in a second.

Frequency units

Frequency values are represented by the type <frequency> . It accepts the following values.

Unit	Name	Description
Hz	Hertz	Represents the number of occurrences per second.
kHz	KiloHertz	A kiloHertz is 1000 Hertz.

1Hz , which can also be written as 1hz or 1HZ , is one cycle per second.

Flex units

Flex units are represented by the type <flex> . It accepts the following value.

Unit	Name	Description
fr	Flex	Represents a flexible length within a grid container

Resolution units

Resolution units are represented by the type <resolution> . They represent the size of a single dot in a graphical representation, such as a screen, by indicating how many of these dots fit in a CSS inch, centimeter, or pixel. It accepts the following values:

Unit	Description
dpi	Dots per inch.
dpcm	Dots per centimeter.
dppx , x	Dots per px unit.

Percentages

A <percentage> is a type that represents a fraction of some other value.

Percentage values are always relative to another quantity, for example a length. Each property that allows percentages also defines the quantity to which the percentage refers. This quantity can be a value of another property of the same element, the value of a property of an ancestor element, a measurement of a containing block, or something else.

As an example, if you specify the width of a box as a percentage, it refers to the percentage of the box's parent's computed width:

```
.box {
  width: 50%;
}
```

Mixing percentages and dimensions

Some properties accept a dimension that could be either one of two types, for example a <length> or a <percentage> . In this case the allowed value is detailed in the specification as a combination unit, e.g. <length-percentage> . Other possible combinations are as follows:

- <frequency-percentage>
- <angle-percentage>
- <time-percentage>

Special data types (defined in other specs)

- <color>
- <image>
- <position>

Color

The <color> value specifies the color of an element feature (e.g. it's background color), and is defined in the CSS Color Module.

Image

The <image> value specifies all the different types of image that can be used in CSS, and is defined in the CSS Image Values and Replaced Content Module.

Position

The <position> type defines 2D positioning of an object inside a positioning area, for example a background image inside a container. This type is interpreted as a background-position and therefore specified in the CSS Backgrounds and Borders specification.

Functional notation

- calc()
- min()
- max()
- minmax()
- clamp()
- toggle()

- attr()

Functional notation is a type of value that can represent more complex types or invoke special processing by CSS. The syntax starts with the name of the function immediately followed by a left parenthesis (followed by the argument(s) to the notation followed by a right parenthesis) . Functions can take multiple arguments, which are formatted similarly to a CSS property value.

White space is allowed, but optional inside the parentheses. (But see notes regarding whitespace within pages for min() , max() , minmax() , and clamp() functions.)

Some legacy functional notations such as rgba() use commas, but generally commas are only used to separate items in a list. If a comma is used to separate arguments, white space is optional before and after the comma.

Specifications

Specification
CSS Values and Units Module Level 4
CSS Color Module Level 4
CSS Images Module Level 3

See also

- CSS Basic Data Types
- Introduction to CSS: Values and Units

Last modified: Nov 15, 2022, by MDN contributors