



CSS Variables - The var() Function





CSS Variables

The var () function is used to insert the value of a CSS variable.

CSS variables have access to the DOM, which means that you can create variables with local or global scope, change the variables with JavaScript, and change the variables based on media queries.

A good way to use CSS variables is when it comes to the colors of your design. Instead of copy and paste the same colors over and over again, you can place them in variables.

The Traditional Way

The following example shows the traditional way of defining some colors in a style sheet (by defining the colors to use, for each specific element):

Example

```
body { background-color: #1e90ff; }
h2 { border-bottom: 2px solid #1e90ff; }
.container {
  color: #1e90ff;
```

```
background-color: #ffffff;
padding: 15px;
}

button {
  background-color: #ffffff;
  color: #1e90ff;
  border: 1px solid #1e90ff;
  padding: 5px;
}
```

Try it Yourself »

Syntax of the var() Function

The var () function is used to insert the value of a CSS variable.

The syntax of the var() function is as follows:

```
var(--name, value)
```

| Value | Description | |
|-------|------------------------------------------------------------------|--|
| name | Required. The variable name (must start with two dashes) | |
| value | Optional. The fallback value (used if the variable is not found) | |

Note: The variable name must begin with two dashes (--) and it is case sensitive!

How var() Works

First of all: CSS variables can have a global or local scope.

Global variables can be accessed/used through the entire document, while local variables can be used only inside the selector where it is declared.

To create a variable with global scope, declare it inside the :root selector. The :root selector matches the document's root element.

To create a variable with local scope, declare it inside the selector that is going to use it.

The following example is equal to the example above, but here we use the var() function.

First, we declare two global variables (--blue and --white). Then, we use the var () function to insert the value of the variables later in the style sheet:

Example

```
:root {
    --blue: #1e90ff;
    --white: #ffffff;
}

body { background-color: var(--blue); }

h2 { border-bottom: 2px solid var(--blue); }

.container {
    color: var(--blue);
    background-color: var(--white);
    padding: 15px;
}

button {
    background-color: var(--white);
    color: var(--blue);
    border: 1px solid var(--blue);
    padding: 5px;
}
```

Try it Yourself »

Advantages of using var() are:

- makes the code easier to read (more understandable)
- makes it much easier to change the color values

To change the blue and white color to a softer blue and white, you just need to change the two variable values:

Example

```
:root {
    --blue: #6495ed;
    --white: #faf0e6;
}

body { background-color: var(--blue); }

h2 { border-bottom: 2px solid var(--blue); }

.container {
    color: var(--blue);
    background-color: var(--white);
    padding: 15px;
}

button {
    background-color: var(--white);
    color: var(--blue);
    border: 1px solid var(--blue);
    padding: 5px;
}
```

Try it Yourself »

Browser Support

The numbers in the table specify the first browser version that fully supports the var () function.

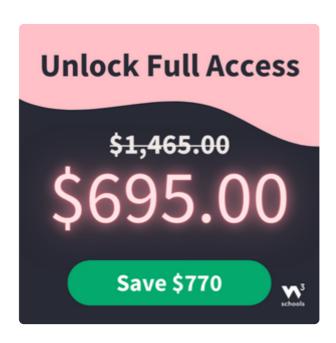
| Function | | | | | |
|----------|------|------|------|-----|------|
| var() | 49.0 | 15.0 | 31.0 | 9.1 | 36.0 |

CSS var() Function

| Property | Description |
|--------------|-------------------------------------|
| <u>var()</u> | Inserts the value of a CSS variable |

Previous

Next >



COLOR PICKER

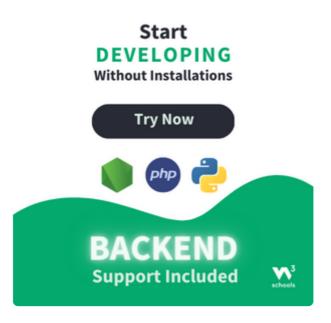




Get certified by completing a CSS course today!



Get started



Report Error

Spaces

Upgrade

Newsletter

Get Certified

Top Tutorials

HTML Tutorial

CSS Tutorial

JavaScript Tutorial

How To Tutorial

SQL Tutorial

Python Tutorial

W3.CSS Tutorial

Bootstrap Tutorial

PHP Tutorial

Java Tutorial

C++ Tutorial

jQuery Tutorial

Top References

HTML Reference

CSS Reference

JavaScript Reference

SQL Reference

Python Reference

W3.CSS Reference

Bootstrap Reference

PHP Reference

HTML Colors

Java Reference

Angular Reference

jQuery Reference

Top Examples

HTML Examples

CSS Examples

JavaScript Examples

How To Examples

SQL Examples

Python Examples

W3.CSS Examples

Bootstrap Examples

PHP Examples

Java Examples

XML Examples

jQuery Examples

Get Certified

HTML Certificate

CSS Certificate

JavaScript Certificate

Front End Certificate

SQL Certificate

Python Certificate

PHP Certificate

jQuery Certificate

Java Certificate

C++ Certificate

C# Certificate

XML Certificate

W3Schools is optimized for learning and training. Examples might be simplified to improve reading and learning. Tutorials, references, and examples are constantly reviewed to avoid errors, but we cannot warrant full correctness of all content. While using W3Schools, you agree to have read and accepted our terms of use, cookie and privacy policy.

Copyright 1999-2023 by Refsnes Data. All Rights Reserved. W3Schools is Powered by W3.CSS.

