

# **CSS Counters**



Next >



CSS counters are "variables" maintained by CSS whose values can be incremented by CSS rules (to track how many times they are used). Counters let you adjust the appearance of content based on its placement in the document.

## **Automatic Numbering With Counters**

CSS counters are like "variables". The variable values can be incremented by CSS rules (which will track how many times they are used).

To work with CSS counters we will use the following properties:

- counter-reset Creates or resets a counter
- counter-increment Increments a counter value
- content Inserts generated content
- counter() or counters() function Adds the value of a counter to an element

To use a CSS counter, it must first be created with counter-reset.

The following example creates a counter for the page (in the body selector), then increments the counter value for each <h2> element and adds "Section <value of the



## **Nesting Counters**

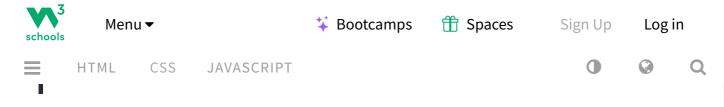
The following example creates one counter for the page (section) and one counter for each <h1> element (subsection). The "section" counter will be counted for each <h1> element with "Section <value of the section counter>.", and the "subsection" counter will be counted for each <h2> element with "<value of the section counter>. <value of the subsection counter>":

### Example

```
body {
  counter-reset: section;
}

h1 {
  counter-reset: subsection;
}

h1::before {
  counter-increment: section;
  content: "Section " counter(section) ". ";
}
```



Try it Yourself »

A counter can also be useful to make outlined lists because a new instance of a counter is automatically created in child elements. Here we use the counters() function to insert a string between different levels of nested counters:

### Example

```
ol {
   counter-reset: section;
   list-style-type: none;
}

li::before {
   counter-increment: section;
   content: counters(section,".") " ";
}
```

Try it Yourself »

# **CSS Counter Properties**

Property	Description
<u>content</u>	Used with the ::before and ::after pseudo-elements, to insert generated content
<u>counter-</u> <u>increment</u>	Increments one or more counter values
counter-reset	Creates or resets one or more counters
<u>counter()</u>	Returns the current value of the named counter



Menu **▼** 

☼ Bootcamps

ff Spaces Sign Up Log in

 $\equiv$ 

HTML CSS JAVASCRIPT







### **COLOR PICKER**





Menu **▼** 

**☼** Bootcamps

**#** Spaces

Sign Up Log in

 $\equiv$ 

HTML CSS JAVASCRIPT









**Spaces** 

Upgrade

Newsletter

**Get Certified** 

**Report Error** 

### **Top Tutorials**



Menu **▼** 

🟅 Bootcamps

**#** Spaces

Sign Up

Log in

 $\equiv$ 

HTML

CSS

**JAVASCRIPT** 







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



Menu **▼** 



HTML CSS

JAVASCRIPT





terms of use, cookie and privacy policy.

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

