content cвойство content позволяет вставлять генерируемое содержание в текст веб-страницы, которое первоначально в тексте отсутствует. Применяется совместно с псевдоэлементами ::after и ::before, они соответственно указывают отображать новое содержимое после или до содержимого элемента, к которому добавляются.

*content: <строка> | attr(<атрибут>) | open-quote | close-quote | no-open-quote | no-close-quote |*

*url | counter | normal | none*

*<строка>*

Текст, который добавляется на веб-страницу, строка при этом должна браться в двойные или одинарные кавычки. Допускается использовать юникод для вставки спецсимволов. Спецсимволы HTML которые начинаются с амперсанда (&sect; например), будут отображаться как есть, т. е. простым текстом (&sect;, а не §).

*attr(<атрибут>)*

Возвращает строку, которая является значением атрибута тега указанного в скобках. Например, a::after {content:attr(href)} добавит после ссылки её адрес, т. е. значение атрибута href. Если указанного атрибута нет, то вернётся пустая строка.

open-quote Вставляет открывающую кавычку, тип которой устанавливается с помощью стилевого свойства quotes.

1. close-quote Вставляет закрывающую кавычку.
2. no-open-quote Отменяет добавление открывающей кавычки.
3. no-close-quote Отменяет добавление закрывающей кавычки.
4. url Абсолютный или относительный адрес вставляемого объекта. Если указанный файл браузер не может отобразить, то значение игнорируется.
5. counter Выводит значение счётчика, заданного свойством counter-reset.
6. none Не добавляет никакого содержимого.
7. normal Задаётся как none для псевдоэлементов ::before и ::after.

counter-increment Стилевое свойство counter-increment предназначено для увеличения значения счётчика приращений, который задается свойством counter-reset. Такой счётчик подсчитывает количество отображений элементов на странице и может выводиться с помощью свойства content и псевдоэлементов ::after и ::before. Это позволяет создавать списки (в том числе многоуровневые), в которых нумерация и вид задаются через стили.

*counter-increment: [<переменная> | <число>?]+ none*

none Запрещает увеличение счётчика для текущего селектора.

<переменная> Задаёт одну или несколько переменных, для которых требуется изменить значение счётчика. Переменные разделяются между собой пробелом.

<число> Определяет значение приращения счётчика. По умолчанию оно равно 1. Допускается использовать только положительные целые числа, отрицательные целые числа и ноль.

*LI { list-style-type: none; }  
OL { counter-reset: list -1; }  
LI:before {counter-increment: list;  
content: counter(list) ". "; }  
LI { list-style-type: none; }  
OL { counter-reset: list; }*

*LI:before {  
counter-increment: list 2;  
content: counter(list) ". ";}  
LI { list-style-type: none; }  
OL { counter-reset: list -1; }  
LI:before {  
counter-increment: list list;  
content: counter(list) ". ";}  
LI { list-style-type: none; }  
OL { counter-reset: list 9; }  
LI:before {  
counter-increment: list;  
content: counter(list) ". ";}  
body { counter-reset: heading; /\* Инициируем счетчик \*/ }  
h2:before { counter-increment: heading; /\* Указываем идентификатор счетчика \*/  
content: "Глава " counter(heading) ". "; /\* Выводим текст перед содержимым тега <h2> \*/ }*

counter-reset Устанавливает переменную, в которой будет храниться счётчик отображений определенного элемента, а также начальное значение счётчика. Такой счётчик может выводиться с помощью свойства content и псевдоэлементов ::after и ::before.

*counter-reset: [<переменная> | <число>?]+ | none*

1. none Запрещает инициацию счётчика для текущего селектора.
2. inherit Наследует значение родителя.
3. <переменная> Задаёт одну или несколько переменных, в которых будет храниться значение счётчика. Значения разделяются между собой пробелом.
4. <число> Начальное значение каждого идентификатора. По умолчанию равно 0.

*li { list-style-type: none; } /\* Убираем исходную нумерацию у списка \*/  
ol { counter-reset: list1; } /\* Инициируем счетчик \*/  
 ol li:before { counter-increment: list1; /\* Увеличиваем значение счетчика \*/  
content: counter(list1) ". "; /\* Выводим число \*/ }*

*ol ol { counter-reset: list2; } /\* Инициируем счетчик вложенного списка \*/  
ol ol li:before {counter-increment: list2; /\* Увеличиваем значение счетчика вложенного списка \*/  
 content: counter(list1) "." counter(list2) ". "; /\* Выводим число \*/ }*

quotes Устанавливает тип кавычек, который применяется в тексте документа. В каждом языке существуют свои традиции для обозначения кавычек, свойство quotes позволяет задать вид их отображения по всему тексту и установить, таким образом, его единообразное оформление. Добавление кавычек происходит автоматически для содержимого контейнера <q>, а также для текста, к которому применяется стилевое свойство content со значением open-quote (открывающая кавычка) или close-quote (закрывающая кавычка).

quotes: [<Строка> <Строка>]+ | none

q {font-family: Times, serif; /\* Шрифт с засечками \*/  
 font-style: italic; /\* Курсивное начертание текста \*/  
 color: navy; /\* Синий цвет текста \*/  
 quotes: "\00ab" "\00bb"; /\* Кавычки в виде двойных угловых скобок \*/ }

псевдоэлемент ::after Псевдоэлемент, который используется для вывода контента после содержимого элемента, к которому он добавляется. Псевдоэлемент ::after работает совместно со свойством content.

*p.new::after {  
 content: "Новьё!"; /\* Добавляемый текст \*/   
 color: #333; /\* Цвет текста \*/   
 background-color: #fc0; /\* Цвет фона \*/   
 font-size: 90%; /\* Размер шрифта \*/   
 padding: 2px; /\* Поля вокруг текста \*/ }*

псевдоэлемент ::before Псевдоэлемент ::before применяется для отображения контента до содержимого элемента, к которому он добавляется. Работает совместно со свойством content.

По умолчанию ::before создаёт строчный элемент.

*li::before { content: "¶ "; /\* Добавляем желаемый символ перед элементом списка \*/ }  
li { list-style: none; /\* Убираем исходные маркеры \*/ }*