

CSS Regions Specification

Revision:

Revision: 0.6

Authors:

Alexandru Chiculita, (Adobe Systems, Inc.)

Andrei Bucur, (Adobe Systems, Inc.)

Mihnea Ovidenie (Adobe Systems, Inc.)

Virgil Palanciuc, (Adobe Systems, Inc.)

Peter Sorotokin (Adobe Systems, Inc.)

Copyright © 2011 Adobe Systems Inc.

Abstract

This specification defines a set of features that go beyond CSS Multi-column Layout. The central concept of this specification is a *region*, which is similar to a column, but without restrictions on its geometry and position. It also defines features for cross-region floats, text fitting and region-based styling.

Table of Contents

[CSS Regions Specification](#)

[Table of Contents](#)

[Introduction](#)

[Defining the flow and the regions](#)

[Defining the threads: 'flow-thread' property](#)

[Defining the threads: 'region-thread' property](#)

[Defining the threads: 'region-index' property](#)

[Region shape: 'region-shape' property](#)

[Region-based styling: 'region-styling' property](#)

[Fitting content in regions: 'region-fitting' property](#)

[Exclusions list: 'exclusions' property](#)

[Exclusion shape: 'exclusion-shape' property](#)

Introduction

CSS Multi-column Layout specification has pushed the limit of what is possible to achieve with

CSS. However it still falls far short of the goal of representing typical magazine, newspaper, or textbook layouts in the digital space. This specification aims to close the remaining gap by giving content creators basic building blocks to express complex layouts. It does not aim to cover higher-level layout issues (e.g. allocating areas to fit all the content completely or placing areas on the page). These issues can be addressed by using either scripting or another CSS module.

The most obvious shortcoming of the CSS Multi-column layout is that columns are all of the same dimensions and placed next to each other. In more complex layouts, content can flow from one area of the page to next one without limitation of the area sizes and positions. For complex layouts, these areas need to be explicitly defined; in this specification they are called regions. Regions are based on the rectangular geometry of the CSS box model. In addition to regions, exclusion areas can be defined and they can act as arbitrary-shaped float elements.

Two other features that are commonly found in complex layouts are region-based styling and text fitting. Region-based styling allows styling to be assigned based on the content placement (e.g. in which region it falls), rather than structure (e.g. selectors). This can be seen as an extension of CSS 2.1 `:first-line` pseudo-element. Text fitting is a feature that forces the layout engine to fill a given set of regions with the given content completely, so that there is no gap left at the end.

Defining the flow and the regions

In most cases, the content presented using complex layouts can be thought of consisting of multiple *flowing threads*. Typically there is one main flowing thread in the document (the "body" thread) and multiple auxiliary ones (sideboxes, pull quotes, etc.). Each flowing thread is usually given a name.

Defining the threads: 'flow-thread' property

<i>Name:</i>	<i>flow-thread</i>
<i>Value:</i>	<name> none
<i>Initial:</i>	none
<i>Applies to:</i>	block-level elements, flows
<i>Inherited:</i>	yes
<i>Percentages:</i>	N/A

<i>Media:</i>	visual
<i>Computed value:</i>	as specified

The ‘flow-thread’ property is used to assign a HTML element to the flowing thread with the specified name. HTML elements that belong to a flowing thread are always displayed in the region marked for the same flowing thread(with region-thread property). If no region with the specified flowing thread is available, elements in the thread are not displayed at all.

Elements with specified flow-thread will be taken out of the normal layout flow and rendered inside the regions.

The ‘flow-thread’ property does not affect CSS cascade and inheritance.

The following “display” properties are compatible with the flow-thread concept:
block | inline-block.

If the ‘display’ property is set to ‘none’ then the ‘flow-thread’ property will be ignored.

In case of other display modes, the flow-thread property will have priority and will make other display modes render as blocks.

Example:

```
<div style="flow-thread:content">
  Lorem ipsum dolor sit amet ...
</div>
```

```
<div style="flow-thread:quotations">
  Nihil sine Deo ...
</div>
```

Defining the threads: ‘region-thread’ property

<i>Name:</i>	<i>region-thread</i>
<i>Value:</i>	<name> none
<i>Initial:</i>	none
<i>Applies to:</i>	regions
<i>Inherited:</i>	no

<i>Percentages:</i>	N/A
<i>Media:</i>	visual
<i>Computed value:</i>	as specified

Assign the region to display the contents of a specified flowing thread.

The content that participates in the specified flowing thread will be laid out across all the regions in the order it appears in DOM.

The content of region elements will be suppressed. The element will act as a block or inline block, depending on display property. Other modes will be default to block.

Example:

```
<div style="flow-thread:content ">  
  Lorem ipsum dolor sit amet ...  
</div>
```

```
<div style="flow-thread:quotations ">  
  Nihil sine Deo ...  
</div>
```

```
<div style="region-thread:content"></div>  
<div style="region-thread:quotations"></div>  
<div style="region-thread:content"></div>
```

Sample rendering

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum non mi libero. Aliquam ut massa eu nibh laoreet convallis. Sed commodo convallis nisl quis sodales. Phasellus ligula nibh, accumsan eu interdum ac, iaculis eu augue. Aliquam suscipit nisl

```
<div style="
  region-thread: content">
```

Nihil sine Deo

Donec posuere, diam nec luctus luctus, lectus leo lobortis tellus, accumsan adipiscing turpis risus eu ipsum. Aenean tempus convallis tincidunt.

```
<div style="
  region-thread: quotations">
```

eget nisi gravida eleifend. Vestibulum dui orci, ultricies at porta id, consectetur sit amet nisi. In volutpat, ante in eleifend venenatis, velit dolor eleifend enim.

```
<div style="
  region-thread: content">
```

Text from the “**content**” flow-thread is rendered through two different elements. In between them, the text from the “**quotations**” flow-thread is displayed in a separate element.

Defining the threads: ‘region-index’ property

<i>Name:</i>	<i>region-index</i>
<i>Value:</i>	<integer>
<i>Initial:</i>	0
<i>Applies to:</i>	regions
<i>Inherited:</i>	no
<i>Percentages:</i>	N/A
<i>Media:</i>	visual
<i>Computed value:</i>	as specified

Defines the order in which content is flowed inside the regions. If region A has its region-index

lower than region B region-index, the content will flow in region A before region B. Negative values are allowed for this property. Regions with equal values will be sorted using the order in DOM. By default all the regions will be sorted using DOM order.

Example:

```
<div style="display:region; region-index:1"></div>
<div style="display:region; region-index:2"></div>
<div style="display:region; region-index:3"></div>
<div style="display:region; region-index:4"></div>
```

Sample rendering

"region-index" directional flow

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum non mi libero. Aliquam ut massa eu nibh laoreet convallis. Sed commodo convallis nisl quis sodales. Phasellus ligula nibh, accumsan eu interdum ac, iaculis eu augue. Aliquam suscipit nisl. Aenean dictum volutpat tincidunt. Phasellus vulputate auctor nulla, id tempus erat faucibus id. Curabitur porttitor risus in libero suscipit euismod. Suspendisse faucibus, nisi vitae tempor dictum, lorem ante convallis libero, in venenatis ipsum vulputate a vulputate justo. Donec sit amet tellus quis felis lobortis dapibus. Phasellus eu augue lacus. Donec venenatis sem eu quam imperdiet quis cursus tortor vulputate. Duis ac diam libero, vitae dapibus mi. Donec id porttitor massa. Sed egestas facilisis semper. Maecenas sit amet arcu sit adipiscing id a libero. Vestibulum sit amet quam quam, in pulvinar lorem. In suscipit, dui eu tincidunt placerat, mauris justo .

Text flows from one region to another based on the priority set by the "region-index" property.

Consuming content from the flow: 'content-consumption' property

Name:	content-consumption
Value:	consume retain
Initial:	consume

<i>Applies to:</i>	regions
<i>Inherited:</i>	no
<i>Percentages:</i>	N/A
<i>Media:</i>	visual
<i>Computed value:</i>	as specified

In most cases, each piece of content in the flow is displayed only once. In some cases, however, content needs to be displayed multiple times (e.g. chapter content at the page header). Setting the ‘content-consumption’ property to *retain* on a region displays the available content from the flow in the region in the usual manner. However, the same content displayed in the region above also gets displayed in the subsequent region(s) as if that region did not exist to consume the content.

Example:

```
<div style="flow-thread:chapter ">
  Chapter I: "The beginning"
</div>
<div style="flow-thread:content ">
  Lorem ipsum dolor sit amet ...
</div>

<div style="region-thread:chapter;
  content-consumption:retain;"></div>

<div style="region-thread:content;"></div>

<div style="region-thread:chapter;
  content-consumption:retain;"></div>

<div style="region-thread:content;"></div>
```

Sample rendering

```
<div style="
  region-thread: chapter;
  content-consumption: retain">
```

Chapter I: "The beginning"

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum non mi libero. Aliquam ut massa eu nibh laoreet convallis. Sed commodo convallis nisi quis sodales. Phasellus ligula nibh, accumsan eu interdum ac, iaculis eu augue. Sed vestibulum ipsum non mauris semper volutpat. Nam quis lorem lorem. Integer id arcu nunc. Aliquam vitae lectus massa, sit amet lobortis tortor. Fusce suscipit aliquet fermentum. Quisque eu metus odio, et porttitor ligula. Fusce eget sapien eget mi tempus gravida. Nunc dictum laoreet mauris, eu aliquet felis commodo lobortis. In hac habitasse platea aliquet eget fermentum dictumst

Chapter I: "The beginning"

donec quis sagittis lorem. Proin dapibus, metus vel auctor hendrerit, leo mi auctor nibh, ut sagittis mi leo et dolor. Sed vel nibh at urna rhoncus lobortis. Proin commodo ornare felis eget fringilla. Integer tristique rhoncus luctus. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Quisque ac ipsum lacus. Duis et diam massa, non aliquet nunc. Praesent lectus mauris, volutpat ut lobortis vel, fringilla id lectus. Vestibulum vitae arcu erat. Sed dapibus risus sed sapien varius quis elementum nisi mattis.

Text from the "chapter" flow-thread is repeated on each region which has the "content-consumption" property set to "retain".

Text from the "content" flow-thread is distributed across the regions. This is equivalent to setting the "content-consumption" property to "consume".

Region shape: 'region-shape' property

<i>Name:</i>	<i>region-shape</i>
<i>Value:</i>	[[<length> <length>]]* [[<length>, <length>]]*
<i>Initial:</i>	EMPTY
<i>Applies to:</i>	regions

<i>Inherited:</i>	no
<i>Percentages:</i>	relative to the region width (odd items, counting from 1) or height (even items)
<i>Media:</i>	visual
<i>Computed value:</i>	sequence of lengths (in pixels)

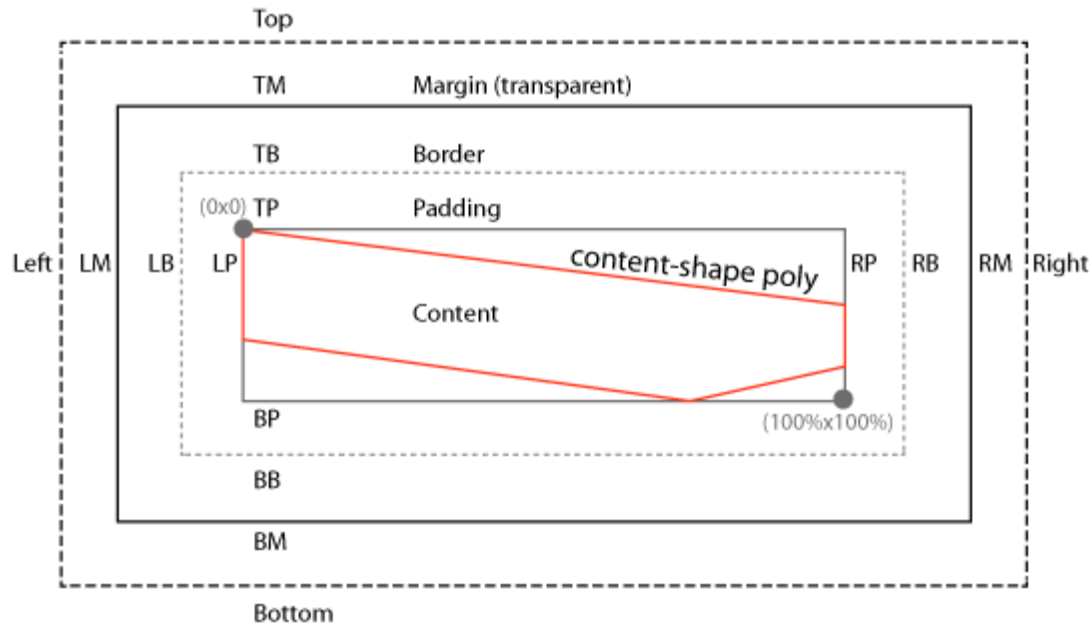
This property defines a boundary where the flow content (text, images, other blocks) should be laid out. By default, if 'region-shape' is not specified, the regular box model is used. If it is specified, it must be a series of points that define a polygon, in the form x1, y1, x2, y2, ..., xN, yN.

The first x and y coordinate pair and the last should be the same to close the polygon. When these coordinate values are not the same, user agents should infer an additional coordinate pair to close the polygon. The (x,y) coordinates are relative to the block's content box.

When defining a polygon area using percentages, the content box of the region must always be fully known in advance (including the height).

One can specify more than one polygon to compose the region-shape. The polygons, separated by ",", will be intersected to compute the final shape that will be taken into account. If the intersection of the polygons is empty, the content will be displayed in the regular box as if 'region-shape' property would have not been defined.

The polygon defined (or computed when more than one) in the *region-shape* property must be fully contained within the content box. The user agent will clip the polygon with the content box.



Region-based styling: using 'region' pseudoselector-like syntax

The styling of the elements in the flow is determined, in general, by the same rules as for all other content. When the content is placed in the region, an additional higher-specificity styling can be applied only to the content. When content is flowed from one region to another, only the fragment of the content that is rendered inside the new region will be restyled using the new computed style.

Using a new pseudoselector-like 'region' keyword, you can define additional styling rules that needs to be applied to the content falling into region.

```
::region <selector> {
  ... CSS styling rules ...
}
```

The styling rules are only applied to the elements that belong to the region's thread and they are only active for the content that falls in the particular region to which they apply. The specificity of region styling rules is considered to be greater than all of the regular rules and smaller than inline rules (which come from style attribute).

Only some of the CSS properties are available when styling the region:

- font properties
- color properties
- background properties
- word-spacing
- letter-spacing

- text-decoration
- vertical-align
- text-transform
- line-height
- clear

The above properties are similar to those available to first-line pseudoselector.

Example:

```
<style type="text/css">
  div
  {
    font-size: 1em;
    color: black;
  }

  ::region div.emphasize
  {
    font-size: 1.3em;
    color: red;
  }
</style>
```

```
<div style="region-thread:content;"></div>
```

```
<div class="emphasize" style="region-thread:content;"></div>
```

```
<div style="region-thread:content;"></div>
```

Sample rendering

Custom region style with @region-styling

Lorem ipsum dolor ist amet non mauris semper volutpat. Nam quis lorem lorem. Integer id arcu nunc. Aliquam vitae lectus massa, sit amet lobortis tortor. Fusce suscipit aliquet fermentum. Quisque eu metus odio, et porttitor ligula. Fusce eget sapien eget mi tempus gravida. Nunc dictum laoreet mauris, eu aliquet felis commodo lobortis. In hac habitasse platea aliquet eget fermentum dictumst donec quis sagittis lorem. Proin dapibus, metus vel auctor hendrerit, leo mi auctor nibh, ut sagittis mi leo et dolor. Sed vel nibh at urna rhoncus lobortis. Proin commodo ornare felis eget fringilla. Integer tristique rhoncus luctus. Pellentesque habitant morbi tristique senectus et netus et malesuada fames.

```
<div style="
  region-thread: content;>
```

```
<div class="emphasize"
  style="region-thread: content;">
```

```
<div style="
  region-thread: content;>
```

Text in the second region has its styles overwritten by higher specificity rules under “region-styling: emphasize”

Fitting content in regions: ‘content-fitting’ property

<i>Name:</i>	<i>content-fitting</i>
<i>Value:</i>	none auto
<i>Initial:</i>	none
<i>Applies to:</i>	region and block elements
<i>Inherited:</i>	no
<i>Percentages:</i>	N/A

<i>Media:</i>	visual
<i>Computed value:</i>	as specified

Tries to fit the element in the allocated space. Setting the ‘content-fitting’ property is ignored when the content width and height are not specified in advance.

When set on a region, the whole content participating in the region-thread will fit inside all the regions.

none

The content is laid out using normal CSS box layout rules.

auto

Content is distributed (stretched or shrunk) across the whole content box or all regions in the thread; this is done by adjusting effective values of the properties that affect text width (word spacing, letter spacing, horizontal scaling).

Defining the exclusions

Exclusions are arbitrary shapes that will be avoided by the user agent when laying out content. Exclusion areas are calculated based on the margin box of other elements that are specified as exclusions elements. Content elements can define a list of exclusions using a CSS selector.

The following restrictions apply when defining exclusions:

- A node should not reference (as exclusions) other nodes that are related to it - i.e. child nodes or ancestor nodes.
- A content (flow) node and its region(s) should not attempt to mutually exclude each other.
- There should be no circular references created by the exclusion areas. When the user agent detects circular references, it may choose to ignore any of them so that the circular references are eliminated.

Exclusions list: ‘exclusions’ property

<i>Name:</i>	<i>exclusions</i>
<i>Value:</i>	CSS selector
<i>Initial:</i>	EMPTY

<i>Applies to:</i>	any element
<i>Inherited:</i>	yes
<i>Media:</i>	visual

Define a list of exclusion elements identified by the specified CSS selector.

Exclusion shape: ‘exclusion-shape’ property

<i>Name:</i>	<i>exclusion-shape</i>
<i>Value:</i>	[[[<length> <length>]]* [[<length>, <length>]]*]
<i>Initial:</i>	EMPTY
<i>Applies to:</i>	all block elements
<i>Inherited:</i>	no
<i>Percentages:</i>	relative to the region margin width (odd items, counting from 1) or margin height (even items)
<i>Media:</i>	visual
<i>Computed value:</i>	sequence of lengths (in pixels)

The ‘exclusion-shape’ is intended to be applied on intruding elements, ex. floating images. The regions are not affected by the ‘exclusion-shape’, only the content that flows into the regions is affected. One particular example when both ‘exclusion-shape’ and ‘region-shape’ can be set on the same element, is for a sidebar floating text-box region.

The ‘exclusion-shape’ property can be used to define a tighter wrap-around area for a given region. It is similar to the ‘region-shape’ and is specified also as a list of points.

The ‘*exclusion-shape*’ for a block selected with the "exclusions" is by default the rectangle that includes the margins.

One can define more than one polygon as an exclusion shape. When more than one polygon is defined, the union of polygons will be considered as the final shape that will be excluded by the layout engine when flowing the content. This is different than in the case of ‘region-shape’ property, where the intersection of polygons is taken into account.

Example:

```
<div id="circle" style="exclusion-shape: poly(x1, y1, .., xN, yN)"></div>
```

```
<div style="region-thread:content; exclusions: #circle;"></div>
```

Sample rendering

Circular exclusion-shape within text.

```
<div id="circle" style="exclusion-shape: poly(x1, y1, ... xN, yN) ;>
```

>Lorem ipsum dolor sit amet, consectetur adipi-
scingelit. Cras diam nisi mollis vel eleifend mollis,
consectetur sit amet odio. Nam mi lacus,
tincidunt id or- nare vitae, frin-
gilla at augue. Ut sed urna
quis nulla pul- vinar portti-
tor sed in elit. Nunc cursus
gravida turpis, ut commodo
risus dignis- sim quis. Pel-
lentesque eu est at purus sem-
per auctor eget ut est. Etiam et nunc a
diam ultricies auctor. Vestibulum tristique volut-
pat ligula et posuere. Cum sociis natoque pena-
tibus et magnis dis parturient montes, nascetur

```
<div style=" region-thread:content; exclusions: #circle;>
```

The #circle DIV's shape is used as an exclusion for the region with content.

References

[CSS3 pseudo-elements](#)

Changelog

Version	Author	Date(mm/dd/yyyy)	Change
---------	--------	------------------	--------

0.6	Mihnea Ovidenie	03/07/2011	<ul style="list-style-type: none">* Remove display flow-content and region values* Region shape can contain more than one polygon whose intersection is used to determine the final shape* Region styling changed from 'region-styling' to a pseudoselector-like definition
0.6	Alexandru Chiculita	03/07/2011	<ul style="list-style-type: none">* Change text fitting property from 'region-fitting' into 'content-fitting' as it can be used for regions and other content too.
0.6	Mihnea Ovidenie	03/07/2011	<ul style="list-style-type: none">* Restrict the properties available to region styling.