

Div container

Simple macro to place your contents in an HTML div container

You can set it's ID and specify your CSS styles.

Macro renders an HTML div around the content you put in it's body with CSS styles you set.



Due to the bug in how Confluence renders user macros the contents of this div macro might be rendered incorrectly (outside the div...). Also user macro rendering struggles to render properly the *textarea* objects

```
## @param Id:title=DIV ID|type=string|required=true|desc=DIV element ID
## @param Css:title=CSS Style|type=string|required=false|desc=CSS styles for this DIV container

<div id="$paramId" #if({$paramCss})style="${$paramCss}"#end>
$body
</div>
```

Macro Browser Information

Macro Name *

Macro names are converted to lower case

Visibility

- ☒ Visible to all users in the Macro Browser
☐ Visible only to system administrators in the Macro Browser

Macro Title *

Description

Categories

Administration
Communication
Confluence content
Development
External content

Icon URL

Absolute URL (e.g. <http://mysite.com/myimage.png>), or relative to the Confluence base URL (e.g. /images/logo/confluence_48_white.png)

Documentation URL

Definition of User Macro

Macro Body ☐ No macro body

Processing ☐ Escaped

☐ Unrendered

You should use this option for bodies that are processed within the template before being output. Ensure that HTML is ultimately output by the template.

☒ Rendered

The body will be rendered so most HTML entered will be passed to the template unmodified but Confluence specific mark up such as macro definitions will be rendered.

Template *

```
## @param Id:title=DIV ID|type=string|required=true|desc=DIV element ID
## @param Css:title=CSS Style|type=string|required=false|desc=CSS styles for this DIV container

<div id="$paramId" #if({$paramCss})style="${$paramCss}"#end>
$body
</div>
```

In action below

Renders "Hello" in red colour with dashed border

Hello

You can easily make this container scrollable by setting the CSS styles parameter as follows

```
overflow: scroll; overflow-y: auto; overflow-x: visible; height: 300px;
```

Important bit is "height:300px", it sets the heights of the container. In this example the container heights will be 300 pixels