

move-to macro

{move-to} Macro

Description

The move-to macro is used to move a chunk of content from one place on the screen to another. Why would you want to do this? Well, it allows you to move content stored on a page in to a Builder panel, that's why! 😊 Now you can add page specific content to sidebars such as a list of "see also" links, etc.

You can also use this macro to place less important content at the bottom of a page and then move it in to the desired position once the page loads - this allows you to have the most important content output near the top of the page resulting in better search engine rankings.

⚠ The move-to macro is incompatible with javascript driven macros

Usage

```
{move-to:targetId|hidesource=false|showtarget=false}
stuff to move
{move-to}

{move-to:target=targetId|hidesource=false|showtarget=false}
stuff to move
{move-to}
```

Parameters

Property	Required	Default	Notes
default	✓		The HTML Id of the target
target	✓		Alternative way of specifying the target
hidesource	✗	true	Whether the source should be hidden while the page is loading
showtarget	✗	true	Whether the target should be revealed once the content has been moved over

Note:

The **default** and **target** parameters are essentially the same thing so you only need to specify one of them.

Examples

Defining the target

Before you can move content in to a target, you first need to define the target! The target can be any HTML element with an "id" attribute, but we recommend using a `` or `<div>` element as shown below:

```
{div:id=target1}You can even have default content in the target - stuff to show should nothing
get moved to the target :){div}
```

Which results in:

You can even have default content in the target - stuff to show should nothing get moved to the target 😊

Note: The "id" needs to be unique on the page - never use the same id more than once per page! We recommend using descriptive names for id's such as "target-see-also-links", but for this tutorial we're going to be really lazy and just call them target1, target2, etc.

As you can see, it looks just like normal content, but that "id" means that you can move stuff in to it, which we'll discuss more further down this page. But before that, let's look at a few other things you can do with targets...

You can customise your target, for example:

```
{div:id=target1/style=background-color:yellow;}
whoo! yellow background
{div}
```

Which gives:

whoo! yellow background

You get the general idea. 😊 In many cases, you will also want to hide the target until it has something moved in to it:

```
{div:id=target1/style=background-color:yellow;display:none;}
this will only appear when something is moved here
{div}
```

The "display:none" bit hides the target.

Moving content to a target

Now we've got our target defined, we need to move some content in to it as follows:

```
{div:id=target2/style=background-color:yellow;}
this is my target
{div}

{move-to:target2}this is my stuff to move{move-to}
```

Which results in:

Unknown macro: {div}

~~this is my target~~

Unknown macro: {move-to}

this is my stuff to move

Moving to a hidden target

As we mentioned earlier, you can define hidden targets that only appear when something is moved in to them:

```
{div:id=target3/style=background-color:yellow;display:none;}
this will only appear when something is moved here
{div}
{div:id=target4/style=background-color:blue;display:none;}
this will only appear when something is moved here
{div}

{move-to:target=target3}make target 3 appear!{move-to}
```

As you can see, our "target4" (with the blue background) doesn't appear because nothing has been moved to it.

CSS Customisation

Not applicable for this macro.

Hints and Tips

None at present.

Frequently Asked Questions

None at present.

See Also

- [div macro](#)
- [span macro](#)