

Custom tags and math operators

This document will describe how to create and use custom tags and math operators.

Creating a tag

You will need to create a `cowx` file. Make sure to link this file in all documents that need it. To do that, add this in the document's head:

```
<head>
  // other things...

  <cowx>path/to/your/file.cowx</cowx>
</head>
```

The `cowx` file can contain any number of tag definitions. Here is the syntax for a custom tag definition:

```
<!tag_name>
  // Tag contents here
</tag_name>
```

where `tag_name` is the name of your tag. Make sure not to use already defined names such as `div` or `span`. And here for a custom math operator:

```
<?operator_name>
  // Operator contents here
</operator_name>
```

Using the tag

In your document, simply include:

```
<!tag_name/>
```

and this will be replaced by the contents specified in the `cowx` file. To use a math operator, use it like the default ones (in math environment, of course):

```
?operator_name
```

Adding parameters

The tags can use arguments. In the declaration, add this:

```
<!tag_name :argument_name>
  // Other things...
  <:argument_name/>
  // Other things...
</tag_name>
```

And use it like that in your document:

```
<!tag_name :argument_name="Some value..."/>
```

When the tag will be replaced, `<:argument_name/>` will be replaced with `Some value...`.

You can declare parameters for math operators exactly in the same way (see math documentation to see how to use them in the document).

You can add as many argument as you want.

:inner parameter

If you declare a parameter with the name `:inner`:

```
<!tag_name :inner>
  // Other things...
  <:inner/>
  // Other things...
</tag_name>
```

it will have a special behavior. The value of the parameter will be the tag inner content in the document. Here:

```
<!tag_name :inner>
  Text inside!
</tag_name>
```

`<:inner/>` will be replaced by `Text inside!`