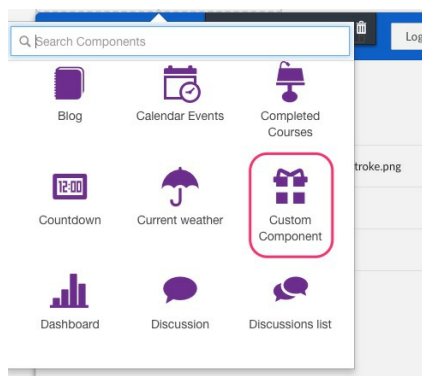


## Creating Custom Components for Pages

### What is a custom component?

You can extend what's available in the list of components within pages by creating a custom component.



### How to create custom component ?

1. Navigate to **Admin > Pages > Component** click on **Add new Component**
2. Fill out the form

**Component Name:** the name of the component that will be displayed

**Description:** short description what the component does, description will also be searchable in the component dialog box.

**Width Constraints:** add how many columns this component can be expanded 1 (minimum) 4 (maximum)

**Height Constraints:** specify if there are height constraint of the component

**HTML :** place html code here

It is recommended not to use id attributes because we don't know how many times this component might be added to a page. Using `element.find('.<class-name>')` from JavaScript to target elements works as a way around this. At Claromentis we add a js- prefix to this class name which lets our designers know that these are not CSS classes.

**CSS:** place css code here

The CSS class mentioned while editing a custom component will be automatically added to a parent element that contains this component. This allows you to control styles for the entire component (header bar and any decoration) rather than just the element that is created in the HTML editor. So far we have only used this to turn the background transparent if a component is made of multiple visible parts (news list, fast access buttons, etc)

Bootstrap 3 is used throughout Claromentis and will already be loaded on any Pages page.

**JS:** place javascript code here

Code is run inside a closure once Pages has added the component to the page and the DOM is ready. Two variables are exposed -

- **element** A jQuery object for the element containing your HTML.
- **options** An object containing your option fields and values.

jQuery is available as jQuery or \$. Requirejs is also available for loading external libraries if needed.

**Custom Component**  
Develop your own pages component by filling out the form below.

**Component Name**  tile-custom-component  
\* Symbolic name for css references.

**Description**

**Width Constraints**

**Height Constraints**

**Glyphicon**

**Options**   
[Add Option](#)

**Add your code below to create your component**  
Your major css class is **tile-custom-component**

**HTML**

```
1 Hello World
```

**CSS**

```
1 .tile-custom-component {
2
3 }
```

**JS**

```
1
```

## What is Options?

Options allowing value to be parsed within the settings for example "Title" of the component

Here is example implementation

**Option**

**Label**

**Internal Name**

**Type**

**Input Control**

**Options (As JSON)**

```
{
  "first": "First",
  "second": "Second",
  "third": "Third",
  "fourth": "Fourth",
  "last": "Last"
}
```

**Placeholder**

**Default Value**

[Cancel](#) [Save](#)

### HTML:

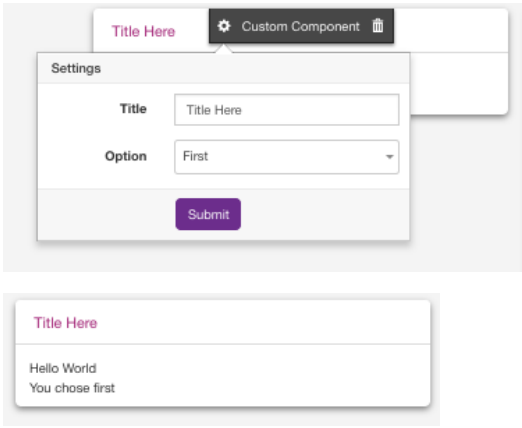
```
<div class="panel-body">
Hello World<br>
You chose <span class="js-choice"></span>
</div>
```

### JS:

```
element.find('.js-choice').text(options.option_1);
```

## Finished Component

Here is example of finished custom component with Title and Option.



You can download example component here

tile-custom-component.zip

Last modified on 1 December 2023 by [Hannah Door](#)

Created on 6 September 2018 by [Michael Christian](#)  
Tags: [CMS](#), [component](#), [module](#), [pages](#), [tile](#), [dashboard](#)