# Garden Club: Featured Boxes on Specific Post Page

**The Issue:** In the design compositions for the Garden Club Front-End, or customer-facing, pages, there are a set of featured boxes which may contain either promotions or image links to other Garden Club content that create flow for the content of the post around themselves. The only way to achieve this effect in HTML and CSS is to use the CSS float property on a box-level element. This means that these items must be integrated into the body HTML of the post in order to exhibit that behavior.

We have identified 4 possible solutions to this issue:

1. **Editor Positioned**This would require the Development team to create a GUI for the creation of such boxes and a content producer to manually place these boxes inside the body of the post.  
   **Pros:** This allows the content producer to have absolute control over where these elements get placed. This alleviates the Development team from needing to create a logic-based solution to positioning.  
   **Cons:** This will be a complicated GUI element to create and will require at least intermediate document-editing skills. Also, the content producer will now, and even be required to have a direct effect on the layout of any and all post pages.
2. **Programmatic Positioned**This is where a content producer simply identifies which featured boxes need to be included, but then some aspect of the Garden Club software identifies where and how to position these elements.
   1. **php Positioned**As both parts get called from the database, the Development team utilizes a custom algorithm to position the element, and incorporate the boxes’ HTML directly into that of the body.  
      **Pros:** This alleviates the content Producer from having to place the element. No complicated GUI element needs to be created for the CMS to allow content producers to manually add them.  
      **Cons:** The algorithm will be complicated to develop, as it will have to make a judgment based mostly on character length (the number of characters present in the body) as to where to place the boxes and may lead to unexpected layouts.
   2. **Javascript Positioned**The element gets loaded towards the end of the page, and then Javascript positions it   
      **Pros:** The Development team will still need to create an algorithm for creating it, but will be able to take advantage of the DOM in order to position the element. Further, since Javascript can access information about window and element heights, it will likely be better at placing these elements. No complicated GUI element needs to be created for the CMS to allow content producers to manually add them, only a simple input.  
      **Cons:** Javascript can be either turned off or be unexpectedly interpreted from a browser so that the elements don’t get placed at all or misplaced. At the very least, it is likely that the customer may experience the element repositioning itself as a brief flash or reload.
3. **Both**This will allow the content producer to manually place the elements as in option 1, but also give them access to a programmatic solution. In the CMS, they may either place the element in the body of the post, or fill in an input field as described in option 2, and let the software determine the boxes’ positions.  
   **Pros:** All of the above pros.  
   **Cons:** All of the above cons.