Service Catalog Helper Functions

Function: forceField()	forceField is a helper function that allows you to perform repeatable steps in the ServiceNow Service Catalog. The function utilizes a 'For Loop' which pre-allocates memory in order to increase efficiency…speed up your page load. This becomes increasingly necessary when you have 100+ variables in your catalog.	

Example:	var **mojo** = new forceField(); //This is an assignment of a variable to the function. Necessary to initiate the function. Do this once at the top of your code.
	
	var **fields** = new Array('name','address','job code');
	
	mojo.show(**fields**);	

Method	Format	Details	Views
full()	assignment.full(Array);	Hides all fields that do not have a value. To exclude a field, simply leave it out of the array. This is useful for fulfillment in order to present necessary data.	**CMS,RITM,Task**
spec()	assignment.spec(Array);	g_form.setDisplay(field, false) and g_form.setMandatory(field,false). If fields contain ' ', 'Please Select', 'Application ID'. **This is a special circumstance method for the System Account catalog item.**	**SP**
hide()	assignment.hide(Array);	g_form.setDisplay = false and g_form.setMandatory = false and g_form.clearValue	**SP,CMS,RITM,Task**
show()	assignment.show(Array);	g_form.setDisplay = true	**SP,CMS,RITM,Task**
showMan()	assignment.showMan(Array);	g_form.setDisplay = true and g_form.setMandatory = true	**SP,CMS,RITM,Task**
clear()	assignment.clear(Array);	g_form.clearValue	**SP,CMS,RITM,Task**
nul()	assignment.nul(Array);	g_form.setValue = Null or Empty	**SP,CMS,RITM,Task**
noMan()	assignment.noMan(Array);	g_form.setMandatory = false	**SP,CMS,RITM,Task**
read()	assignment.read(Array);	g_form.setReadOnly = true	**SP,CMS,RITM,Task**
noRead()	assignment.noRead(Array);	g_form.setReadOnly = false	**SP,CMS,RITM,Task**

Function: guardian()	Guardian is HTML element manipulater. Because of the way SP works, we have lost the the ability to hide and show variable sets. Guardian give this capability back to you. Additionally, there are additional functions to manipulate ANY element on your form (ie. buttons, labels, input elements).	

Example:	var **marvel** = new guardian(); //This is an assignment of a variable to the function. Necessary to initiate the function. Do this once at the top of your code.
	marvel.showVsetAll(4); //This will show all variable sets starting with the 4th one on your list.
	marvel.showVset(1); //This will show the 1st variable set on your list	

Method	Format	Details	Views
clas()	assignment.clas(element,id,style,value);	ex: marvel.clas('div.b-t.wrapper-md.ng-scope',0,'display','none'); . This example will hide the button group at the bottom of the form. The first parameter is the HTML element type and the Class, each part separated by a '.' You can use the 'inspect' tool in your browser to find this information. The second parameter, id, is the instance of that class. If it's the first instance, then it's 0, the second would be 1, and so on. With the button group, there's only one on the page, so I could easily assume the id = 0. 'Display' is the CSS directive, and the 'none' is the value. You can find these by going to W3C Schools and searching for CSS. A class is one of many, an ID is a single identifer for only one HTML element.	**SP**
ele()	assignment.ele(element,style,value);	ex: marvel.ele('sp_formfield_IO:0812b699db94c7003217f3861d961974','display','none'); . This example will hide the text box input on a form. The first parameter is the HTML element ID. You can use the 'inspect' tool in your browser to find this information. When you look at the element, it will have an attribute labeled...'ID'. 'Display' is the CSS directive, and the 'none' is the value. You can find these by going to W3C Schools and searching for CSS.A class is one of many, an ID is a single identifer for only one HTML element.	**SP**
hideVset()	assignment.hideVset(#);	ex: marvel.hideVset(2);. This will hide the 2nd variable set in your list of variable sets	**SP**
showVset()	assignment.showVset(#);	ex: marvel.showVset(2);. This will hide the 2nd variable set in your list of variable sets	**SP**
hideVsetAll()	assignment.hideVsetAll(#);	ex: marvel.hideVsetAll(2);. This will hide all variable sets starting withyour 2nd variable set.	**SP**
showVsetAll()	assignment.showVsetAll(#);	ex: marvel.showVsetAll(2);. This will show all variable sets starting withyour 2nd variable set.	**SP**