Workflow:

|  |  |  |
| --- | --- | --- |
| layout() | One-time Init | Updates the initial user interface for the cipher handler. This is a one-time operation. If the editEntry parameter is passed on the URL, then that entry is loaded and the cipher is initialized with it as if it were loaded from the menu. Any additional URL parameters are parsed and passed in as the initial state values. |
| buildCustomUI() | One-time Init | Initializes the layout of the handler. This initializes the contents of the precmds,postcmds,cmdbuttons,pre undocmds divs. These fields are filled in one time and the divs are replaced by unnamed divs so that the customUI is not initialized a second time. |
| attachHandlers() | After Any DOM change | Called whenever any elements are added to the DOM to connect the UI element to the corresponding code. |
| genPreCommands() | One-time Init | Generates the UI elements associated with the precmds div. |
| genPostCommands() | One-time Init | Generates the UI elements associated with the postcmds div. |
| genCmdButtons() | One-time Init | Generates the UI elements associated with the cmdbuttons div. |
| genUndoRedoButtons() | One-time Init | Generates the UI elements associated with the undocmds div. |
| restore() | Anytime | Restores a previous saved state and repopulates the UI (used for Load, URL Initialization, Undo/Redo) |
| save() | Anytime | Create a copy of the current state of the solver/generator |
| cmdButtons[] |  | Buttons to populate on the page (used by genCmdButtons()) |
| init() | One-time Init | Called when the class is initially instantiated |
| createFreqEditTable() | On new Cipher load | Called to create the HTML table to display the frequency of characters. It is used to populate the .freq div. |
| load() | Click the load button | Loads cipherstring data into a solver, preserving all solving matches made |
| reset() | Click the reset button | Loads cipher string data into a solver, resetting all solving matches made. |
| setUIDefaults() | Anytime values are changed | Cleans up any settings, range checking and normalizing any values. This doesn't actually update the UI directly but ensures that all the values are legitimate for the cipher handler Generally you will call updateOutput() after calling setUIDefaults() |
| updateOutput() | Anytime | Update the output based on current state settings. This propagates all values to the UI |
| build() | When cipher or options are changed | Generates the UI for the given cipher |
| updateSel() | User event | Handle a dropdown event when they are changing the mapping for a character . Note that it may have to process swapping any other characters which it is replacing |
| setChar() | User event | Process setting a new mapping for a character. |
| displayFreq() | On new Cipher load | Update all the frequency values in the table |