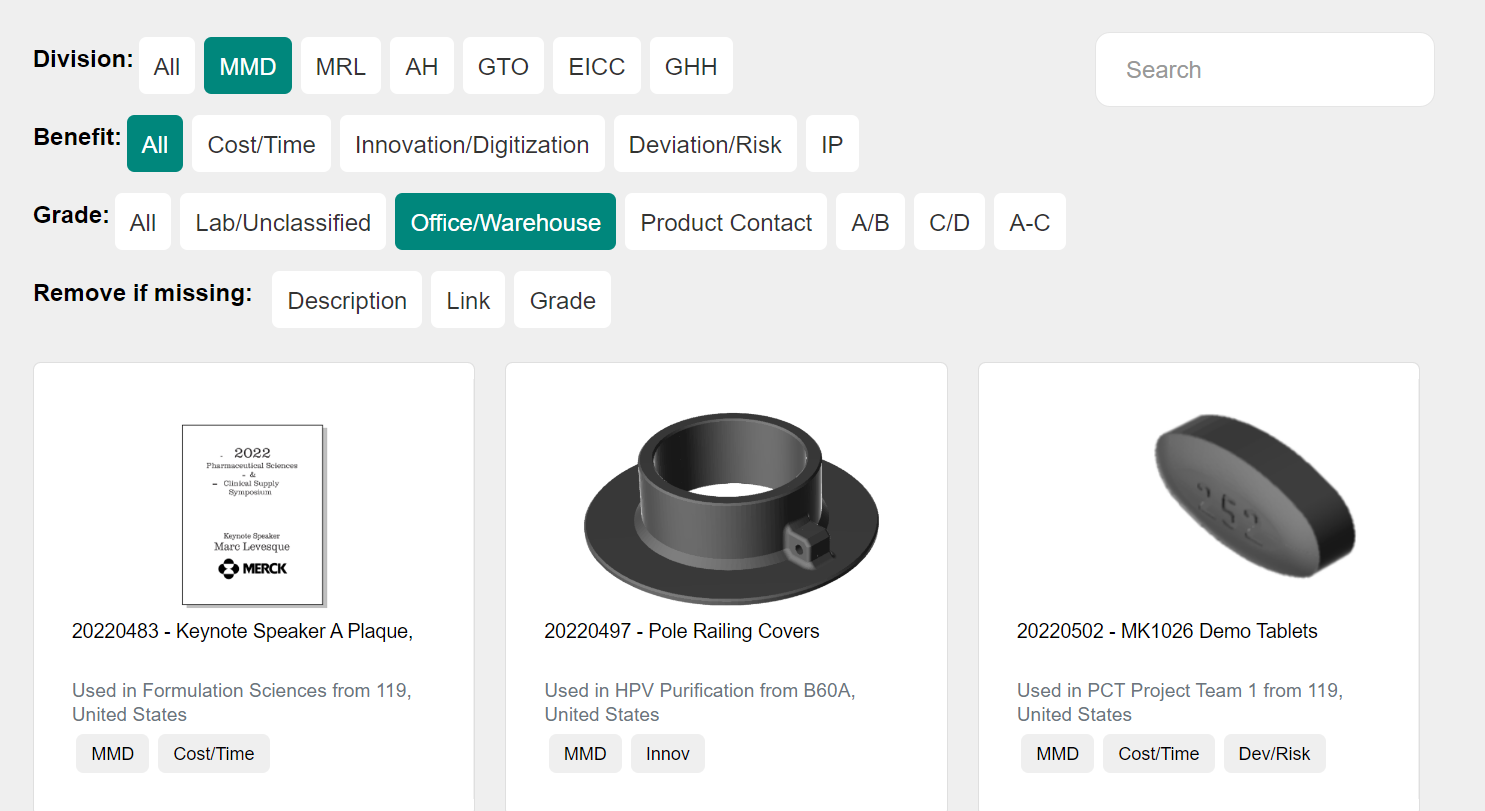
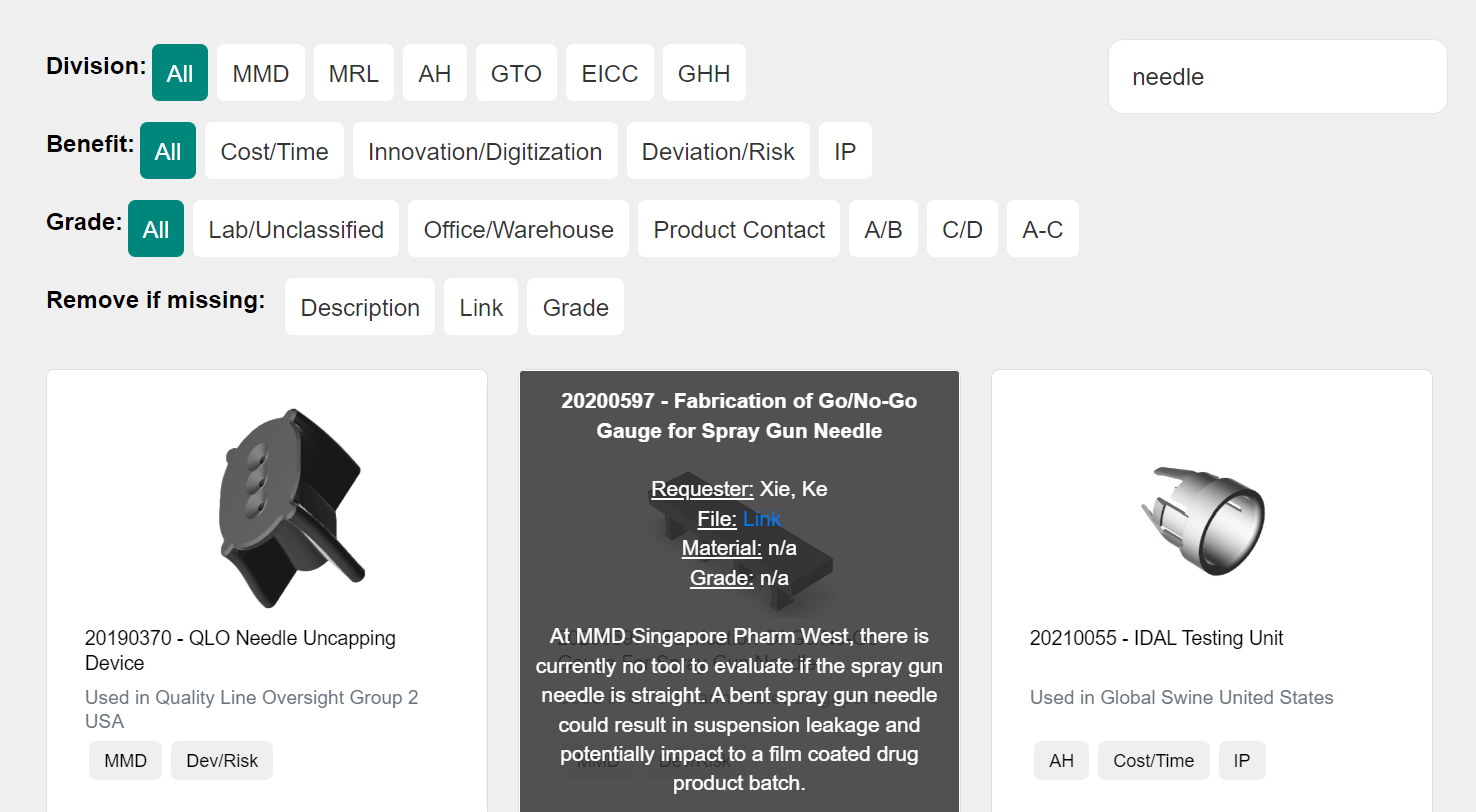
# About

The 3DT catalog was designed and programmed by Zuting Chen, an IT Intern for Merck in the summer of 2022. This document was made to explain some of the programming and how to update certain things.

**Combination Isotope Filtering**:



**Searching & Information** (hover or tap on card):

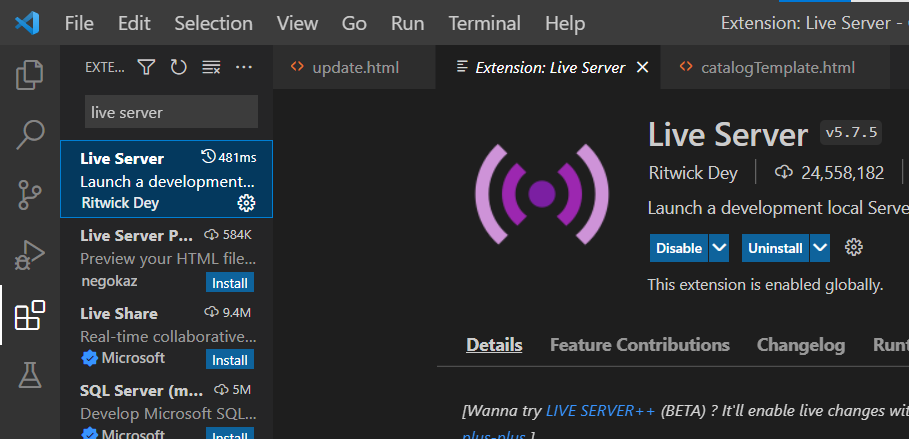


# Live Server Setup

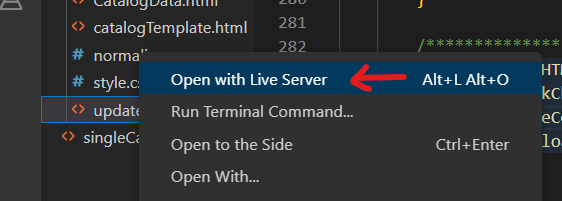
**! Important:**This update.html needs to be opened as a "Live Server" in Visual Studio Code.

*Download Live Server from VS Code Extensions library. Right click on the .html file and "Open with Live Server"  
To stop a live server (Ctrl + Shift + P -> Live Server: Stop Live Server)*

Installing extension:

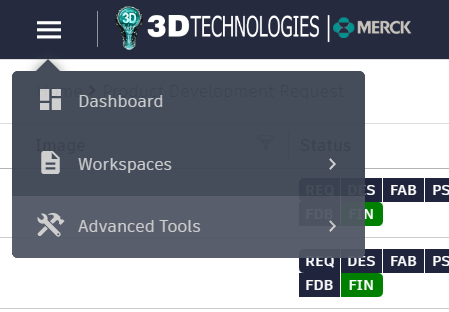
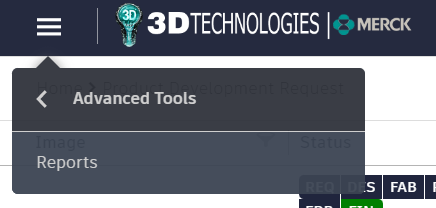


Opening Live Server:

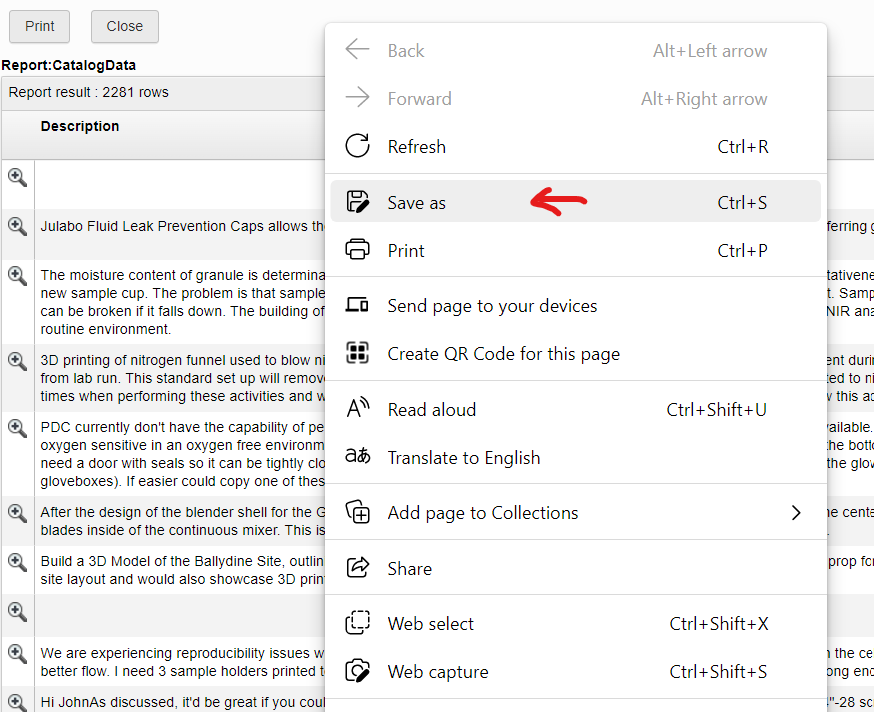
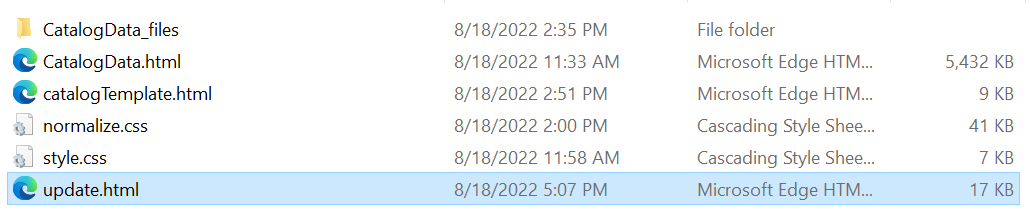


# How to update catalog data

1. Go to [PLM Reports](https://merck.autodeskplm360.net/reportListAction.do) in Advanced Tools on top left of PLM.

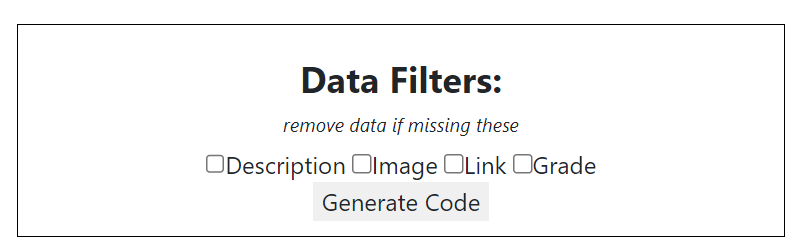


1. After the page loads, save the page HTML in the folder with this file using "Save As" (right click -> Save As, OR Ctrl + S).

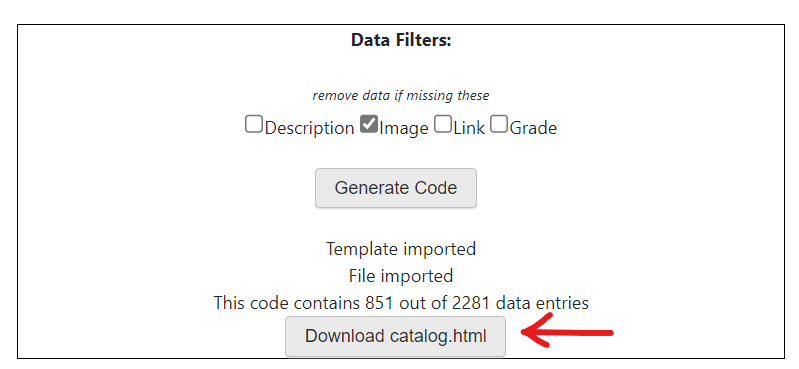
 

^ Save in same folder as update.html

1. The saved file should be named "CatalogData.html" and folder "CatalogData\_files".
2. Once set, select your data filters and hit "Generate Code" (box on this page).



1. After download button shows up, download the file "catalog.html”.



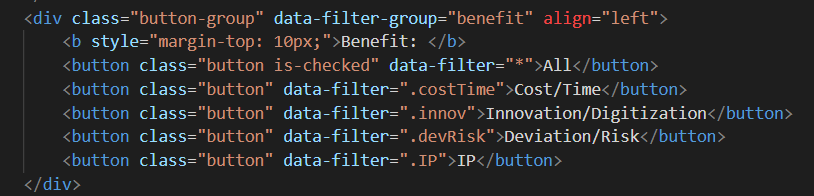
^ Code with 1000+ data entries are already high load. Best to keep under 1500

1. All done! Make sure you have catalog.html, normalize.css, style.css, CatalogData.html, and the CatalogData\_files folder in the same folder if moved.

# How to update filtering buttons

For if you would like to add additional filtering options for isotope filtering at the top. The filtering JS used in “catalog.html” is based on [this isotope filtering example](https://codepen.io/desandro/pen/WyOZjx).

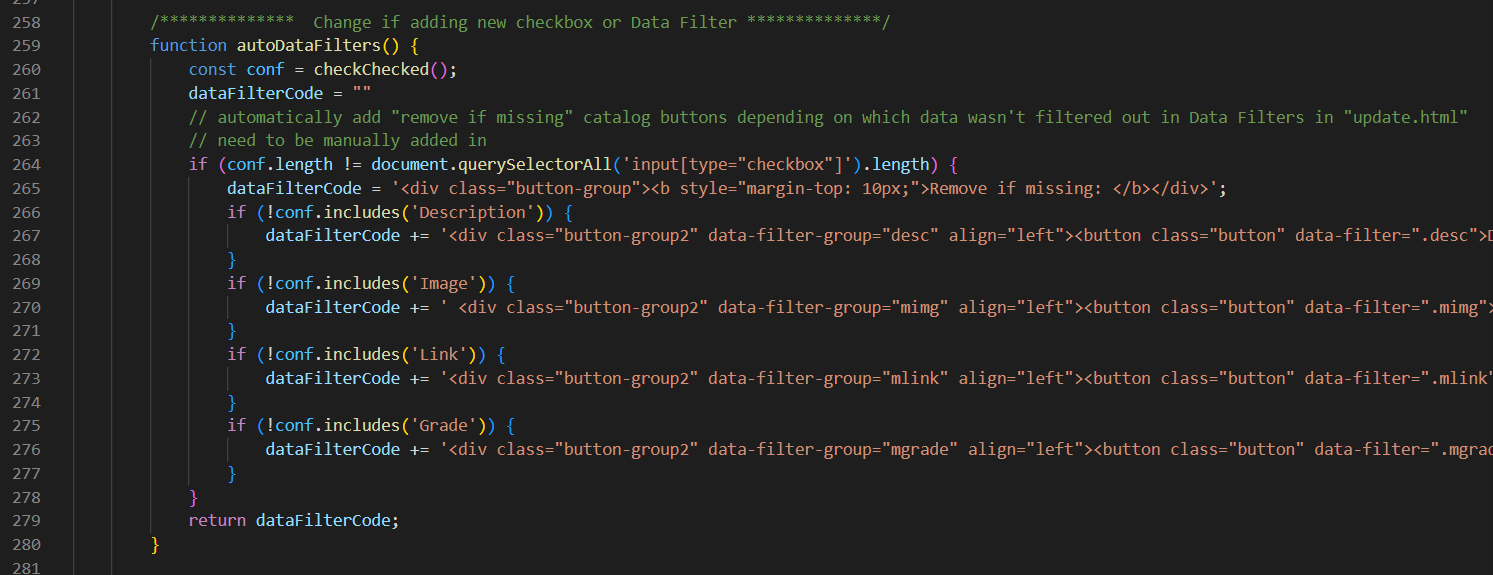
## Buttons for filtering by sections



* 1. Go to “catalogTemplate.html” in VS Code.
  2. Search for the section with <button>s, as shown above.
  3. Copy from the <div> with class=“button-group” to the end of div </div>
  4. Change data-filter-group to be a unique value (if making a new row of buttons)
  5. Change data-filter for each button to a corresponding class name with . in front of it
     1. This class name then needs to be changed in codePiece() in “update.html”. Make a condition for if the data should be selected. If that condition is true, then add to missingClasses the aforementioned class with a space in front of the class name

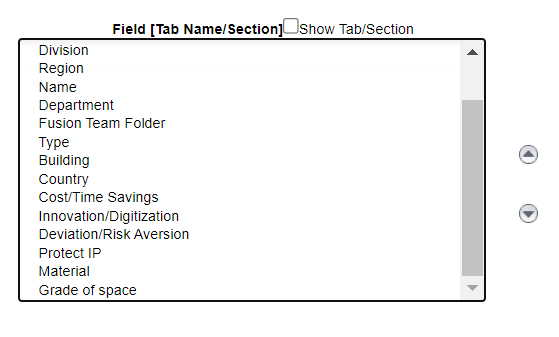


## Buttons for filtering by “Remove if missing”

* The “Remove if missing” buttons are automatically generated for catalog.html based on the checkboxes in update.html
  + But you will need to go into autoDataFilters() in “update.html” to add the button details

# Updating PLM Report Fields

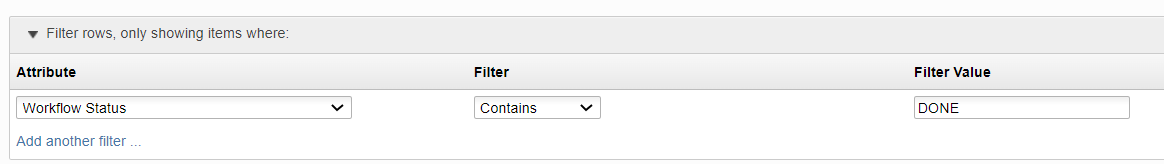
This change can require the most changes in the code. In order to prevent extra work, please add additional fields below the ones prepared in the CatalogData report



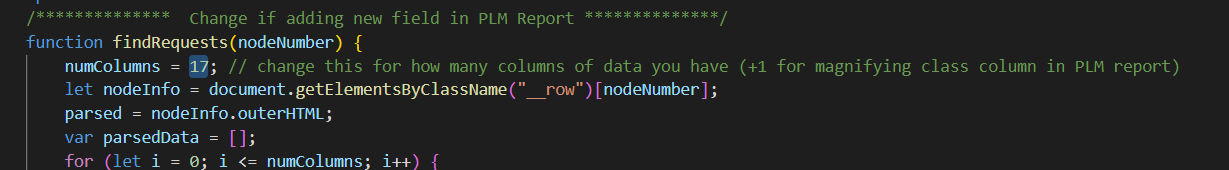
* Each of these fields correspond to an array value as outlined in [parsedData](#parsedData). The code is hard coded to correspond to those specific fields.

### Adding a Field to the PLM

1. Go to PLM Reports in Advanced Tools on top left of PLM.
2. Find a report named CatalogData or make one as such with fields in the order as specified in [parsedData](#parsedData)
   1. If making a report, also filter by “Workflow Status” attribute to remove unfinished requests



1. Generating catalog.html and updating the button filters is the same process as above.
2. Change numColumns in findRequests() in “update.html” to match the number of columns in PLM report (or number of fields + 1)



1. Update codePiece() in “update.html” according to how you want the extra information to be displayed in each card. More about the card code in [cardCode](#cardCode).

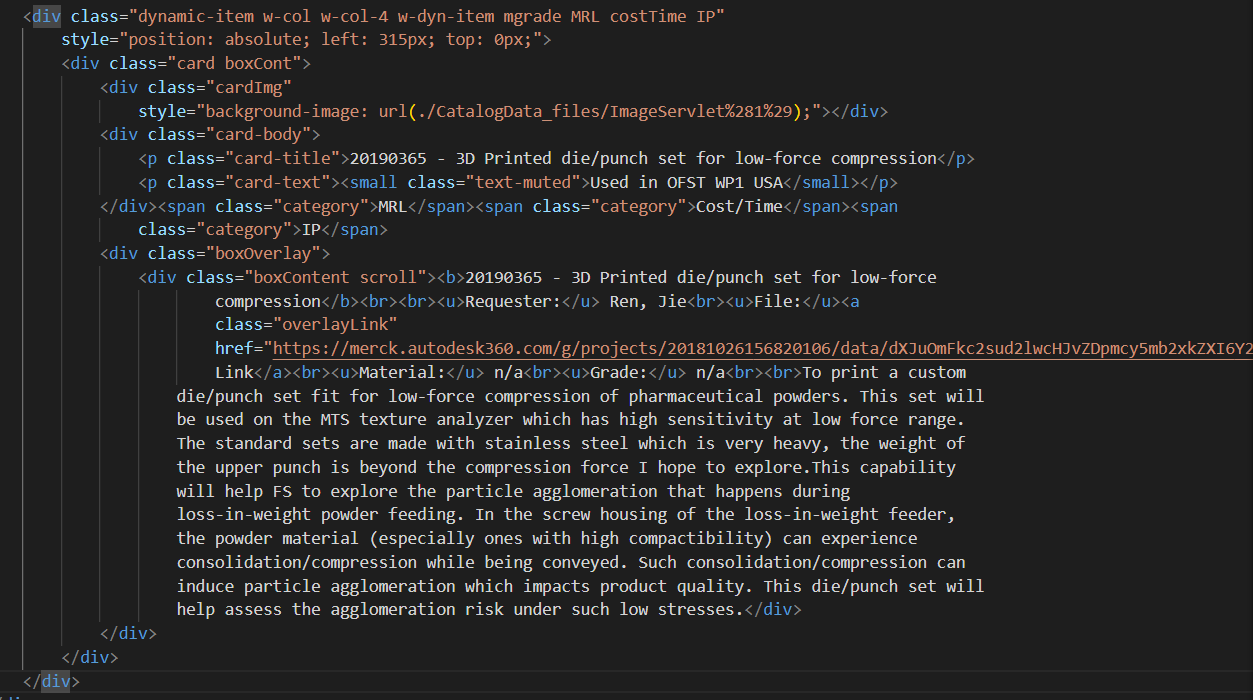
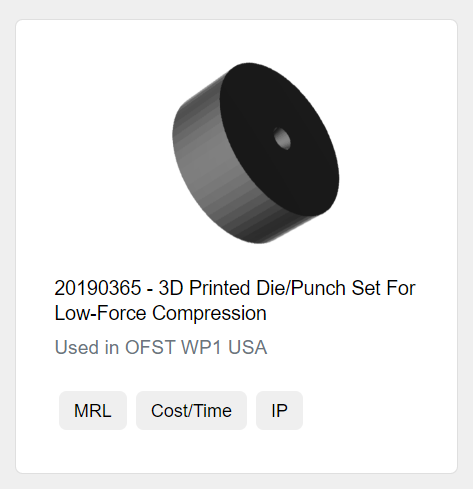
# Understanding the Code

## Files

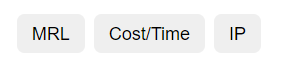
* CatalogData\_files folder (imported)
  + Images from PLM report
* CatalogData.html (imported)
* catalog.html
  + MAIN webpage file
  + Generated in [How to update catalog data](#_How_to_update_1)
* catalogTemplate.html
  + Catalog format without the cards
  + Update this if you need to [update filtering buttons](#_How_to_update)
* normalize.css
* style.css
* update.html
  + Run this file using a [Live Server](#_Live_Server_Setup) to get “catalog.html”

## Common Terms, Variables, and Functions

* **catalogTemplate.html**
  + data-filter-group: differentiates one group of isotope filtering selections from another. Name of this doesn’t matter as long as they are distinct
  + data-filter: what the isotope filtering looks for in the card classes. If selected, then only cards classed with those data-filter values will be shown
* **update.html**
  + cardCode: gathers up the HTML code (shown below) for a single card in codePiece()



* + cardTags: gathers up the HTML code for the tags for a single card in codePiece()



* + dataFilterCode: used in autoDataFilters to store generated HTML code for “Remove if missing” buttons
  + missingClasses: gathers up the classes for isotope filtering for a single card in codePiece()
  + numColumns: in findRequests(). The highest value of parsedData (below) or the amount of columns in the PLM report (including magnifier)
  + parsedData: an array used to store a single row of data in the PLM report at any given moment
    - parsedData[0] - Magnifier, links to request on PLM
    - parsedData[1] - Description
    - parsedData[2] - Image
    - parsedData[3] - Request title
    - parsedData[4] - Division
    - parsedData[5] - Region
    - parsedData[6] - Name
    - parsedData[7] - Department
    - parsedData[8] - Fusion team folder
    - parsedData[9] - Type
    - parsedData[10] - Building
    - parsedData[11] - Country
    - parsedData[12] - Cost/Time Savings
    - parsedData[13] - Innovation/Digitization
    - parsedData[14] - Deviation/Risk Aversion
    - parsedData[15] - Protect IP
    - parsedData[16] - Material
    - parsedData[17] - Grade of space
* **Other Functions**
  + Broadly speaking, these are used to generate code in “update.html”. The contents of the generated code aren’t affected. These are functions that put together operations.
  + loadHTML()
    - Initiate data update process when user clicks “Generate Code”
  + checkChecked()
    - Retrieves checked checkboxes
  + writeCode()
    - Compiles code pieces (cardCode) into code for the pge
  + downloadFile()
    - Download function for downloading “catalog.html”

# Future Work

* Add 3DT navbar and footer for integration into 3DT website
* More flexible filter and filter buttons, possibly. More flexibility in code generation to not require manual input of data configurations
  + Currently, you may need to update multiple sections of code to change things like:
    - PLM Report contents
    - Filtering options/buttons
    - Tags on cards
* Reduce load on website through caching and etc.
  + Current website has notable load times for over 1000+ requests
  + Maybe a different type of search or filtering
* Easier data retrieval and automated update process
  + Tinh Nguyen-Demary suggested that the Forge or Fusion 360 API connects the PLM data to other interfaces
* PLM Request details
  + Start adding material info and grade of space
  + Keep objects in images centered
  + Utilize images and descriptions