JavaScript: In Magento, we’ll be having two flavors of JavaScript:

* Vanilla JS
* Knockout JS

Vanilla JS - Vanilla JS/JQuery scripts are managed by a JavaScript Package Management called RequireJS. For Vanilla JS, we follow some rules (e.g. class.slider should have a sass file called \_slider.scss and for a js file we can create a file called dyson-slider.js, here we created a prefix called Dyson to avoid any naming clashes). We can go to the slider class to find the corresponding JavaScript file based on block and element. If JavaScript doesn’t related to specific things, it means if they don’t follow BEM, we’ll try to keep it cleaner like any code module/object. Magento can be configured with a mixture of to deploy with JavaScript bundled and JavaScript unbundled

Knockout JS –It’s a framework that Magento ships and basically gives a Single Page Functionality (SPA) for the checkout. Knockout JS. It’s mostly used in the checkout. For checkout’s they don’t use Magento checkout longer, Dyson will use a combination of the Amasty One Step Checkout commercial extension and Single Page Checkout that enhances the Amasty one to create an OXP like experience for customers. In the Dyson homepage, it’ll shows an alternative basket icon in the header whenever there’re items in the basket. The Ajax call to a controller can be done in JS file and if the call is successful, and if they’re items in the cart, then JQuery select the basket elements or item and adds it to basket item, this class will give the background image of that element

Dyson Theme leap - Dyson Theme Leap is built using naming conventions from BEM and was originally a child theme of Snowdog/blank theme. Below are some of the site header types:

* Site Header: we use the block for header and keep it sticky and make the theme is navigable, the code for site header page can be found in the below css file path "dyson/web/css/layouts/\_header.scss"
* Site Header in XML: We use the default.xml for all the layouts, pages and sites. In Dyson site header xml file we will be having Dyson header top menu, search and mini cart items of the homepage.
* Site Header Menu: In the site header menu, there’re few classes having child items, it means they have some categories when viewed and rest of classes doesn’t had child items. We can find the code for the Site header menu in the path of "vendor/dyson/dyson-theme-leap/Magento\_theme/templates/html/topmenu.phtml"
* Site Header Search: When we enters data onto the search the html element attaches the ‘data-toggle-content’ and register it as trigger and will grab value and finds the element with the ID that matches of attached attribute and registers as an target item, then the target item is presented and done in CSS in reaction to the searched and opened class for the target.
* Site Header Mini Cart: The code of mini cart can be found in path of following path "vendor/dyson/dyson-theme-leap/Magento\_Checkout/templates/cart/Minicart.phtml”
* Basket: This is probably an area of release and Dyson basket is part of it, it still represents Magento theme. The main layout file is an override of checkout\_cart\_index.xml. We have soft product layer for adding sub items inside, we have available delivery method table which is brought in by sorted pro module and this tables gets rendered at the bottom of the basket if sorted pro is enabled and available. If sorted pro is disabled, we can configure the store configurations called basket assurances.
* Single page checkout: Dyson’s checkout requirements are extended from base module of Amasty single page checkout and the configurations are found in admin at stores configurations Amasty extension Single page checkout. The functionality look and feel of the checkout is modified by G&V module , this module includes plugin’s like JavaScript mimins , layout.xml and php plugin’s

Dyson CMS Components: Dyson CMS has many row types and only some of them were active/inactive and can be reactivated. We have one row type and pattern/concept for the remaining other row is same. Each and Every row type has its own individual or collection of phtml’s in frontend and each row has some admin fields to fill depending on what you have chosen. RCC and Trade up cards were the two of the most used Dyson CMS components. Dyson CMS row type phtml’s are found in "vendor/dyson/module-cms/view/frontend/templates/row/"

We have following Dyson CMS components.

* Dyson CMS RCC: It’ll power all the large banners and other heavy promotional contents like images, videos in the Dyson website. Dyson CMS RCC is a concept in OXP and we added RCC to Dyson CMS for an attempt to standardize the content population and asset creation between Magento and OXP. If you choose row type as RCC in Dyson admin section in the homepage then everything falls dynamic under row type RCC
* Dyson CMS RCC in admin : Basically it is regarding content and attributes in admin page

1. Content 1 – RCC can be full width and will be just using Content 1, e.g.: the figure in Dyson CMS RCC
2. Content 2 – RCC can be partial which will be having two sets of content i) RCC Content Anatomy ii)Positioning and Theme

* Dyson CMS RCC Templates: RCC is the biggest and most complex row type. Most of the Dyson CMS row types have a single phtml file. RCC had its own folder and where it’s having multiple phtml’s making it easily manageable. It can be found in the templates/row/rcc folder

Dyson CMS Trade up Cards Template: The Trade up Cards were included in the Dyson CMS. We’ll be having Trade up Cards aligned horizontally with a slider adjusted to view the cards as well as the grid of a cards that will wrap onto the new rows. The functionality of the Trade up Cards is to show the collection of the products in a card format. The content of cards is managed in products not by the trade up cards as it’s just a template. Dyson CMS Trade up Cards existed before the Dyson CMS and could be shown on the product or category page in the fixed position