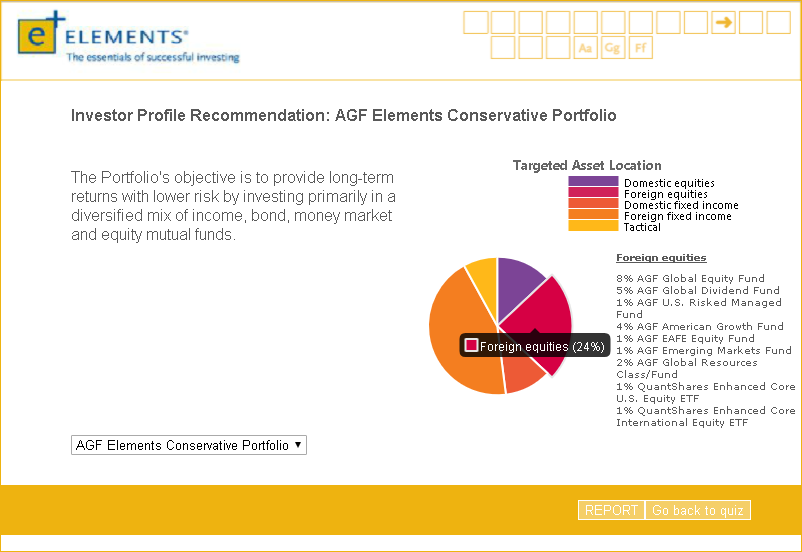
About the Pie Chart

Along with the questionnaire results that are displayed on the last slide, a pie chart is generated to deliver visual information about the targeted asset locations for each type of Elements portfolio.

Pie Chart features:

* Dynamic (number of slices, legend items, chart title, colors)
* On-hover animation (colour of slice darkens)
* Tooltip displays name of legend item + its % on hover
* Slices explode/implode upon being clicked
* Slices display associated funds list upon being clicked

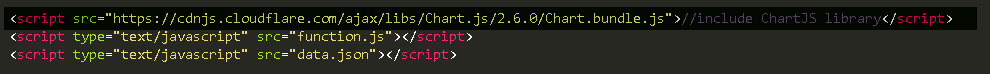


Getting into the Code

The pie chart is created using Chart.js, a JavaScript framework for developers to create different types of graphs. [Official Chart.js Documentation](http://www.chartjs.org/docs/latest/)

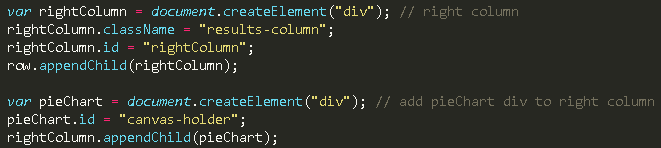
## HTML components

**1. To use Chart.js, the library must be referenced in the HTML as such:**



This line indicates to the browser running the script that it will be using features included in the chart.js library. The version used in this code is v2.6.0.

**2. Create the HTML canvas container (div) *within the JavaScript*:**



Due to the way the quiz layout is set up (a large banner where a section slides into view), the canvas container cannot be declared directly in the HTML. Using the lines of code above, a div with id “canvas-holder”. Since it will be in the right column of the results page, the div is appended to parent id “rightColumn”.

**3. Previously created canvas/iframe must be removed because another chart is loaded in its place:**



This code must go before the pie chart creation function is called so that any existing canvas and iframe is removed. If this is not done, any pie chart that is created over an existing one will glitch out and display data from previous charts.

**4. “Create Pie Chart” function:**



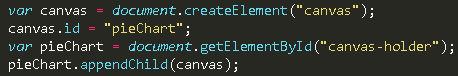
There are many instances in which a pie chart needs to be generated throughout the code; creating a function, “createPieChart”, allows it to be easily called whenever necessary. The rest of the code regarding the pie chart goes within the two curly brackets to indicate that they make up the function. In order to perform its function, 3 parameters need to be passed in: “slide”, “i”, “portfolio”.

**“slide”** – the blank slide that the pie chart is generated on; portion that slides in/out of view

**“i”** – the numerical index of the slide on which the pie chart is generated

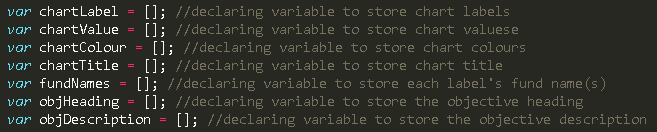
**“portfolio”** – the investment portfolio (e.g. yield, conservative, balanced, growth, global) that the investor has been matched with

**5. Create the canvas HTML node *within the JavaScript*:**

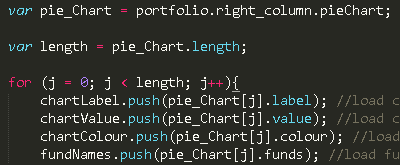


The code above creates a <canvas> node with id “pieChart” within the HTML, which is where the pie chart is rendered. It is also appended to the “canvas-holder” div so that it can be styled on the page.

**6. Declare variables that will be used to store the dynamic data from the JSON files:**



**7. Store the data from the JSON files in the corresponding variables:**



The variable, “pie\_Chart”, stores the array of objects assigned to the object key “pieChart” from the JSON file. The variable, “length”, stores the *number of indexes* this array contains. In context, this number refers to the number of slices the pie chart will have.

The “.push” function loads the preceding variable with data from within the brackets that follow. Arrays can also be loaded into a variable using “.push” when the index is incrementally looped (instead of replacing existing variable data, it is added onto the end and stored as the value of the next index).

8.