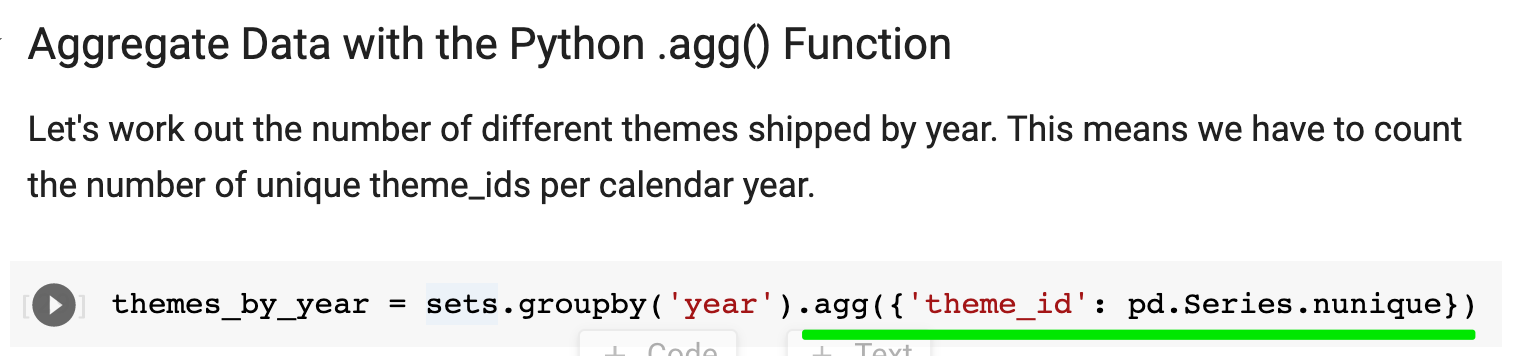
**How to use the Pandas .agg() function**

Often you find yourself needing to summarise data. This is where the .groupby() function comes in really handy. However, sometimes you want to run even more operations based on a particular DataFrame column. This is where the [.agg()](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.agg.html) method comes in.

In our case, we want to calculate the number of different themes by calendar year. This means we have to group the data by year and then count the number of unique theme\_ids for that year.

**Number of Themes per Calendar Year**

We can accomplish this by chaining the .groupby() and the .agg() functions together:



Note, the .agg() method takes a dictionary as an argument. In this dictionary, we specify which operation we'd like to apply to each column. In our case, we just want to calculate the number of unique entries in the theme\_id column by using our old friend, the .nunique() method.

Let's give our column in themes\_by\_year a more appropriate name and let's take a look at what we've got:



Here we can see that LEGO only had 2 themes during the first few years, but just like the number of sets the number of themes expanded manifold over the years. Let's plot this on a chart again.

Challenge

Create a line plot of the number of themes released year-on-year. Only include the full calendar years in the dataset (1949 to 2019).

.

.

..

...

..

.

.

**Here's the solution:**

1. plt.plot(themes\_by\_year.index[:-2], themes\_by\_year.nr\_themes[:-2])

Again, we're using the same slicing technique as before. In the chart, we can see that LEGO has pretty consistently added more and more themes until the mid-1990s. From then the number of themes has stagnated for around 10 years or so until the early 2010s.

