When we explored all of the data sets, we noticed that some of them, like class\_size and hs\_directory, don't have a DBN column. hs\_directory does have a dbn column, though, so we can just rename it.

However, class\_size doesn't appear to have the column at all. Here are the first few rows of the data set:



CSD BOROUGH SCHOOL CODE                SCHOOL NAME GRADE  PROGRAM TYPE  \

0    1       M        M015  P.S. 015 Roberto Clemente     0K       GEN ED

1    1       M        M015  P.S. 015 Roberto Clemente     0K          CTT

2    1       M        M015  P.S. 015 Roberto Clemente     01       GEN ED

3    1       M        M015  P.S. 015 Roberto Clemente     01          CTT

4    1       M        M015  P.S. 015 Roberto Clemente     02       GEN ED

Here are the first few rows of the sat\_results data, which does have a DBN column:



DBN                                    SCHOOL NAME  \

0  01M292  HENRY STREET SCHOOL FOR INTERNATIONAL STUDIES

1  01M448            UNIVERSITY NEIGHBORHOOD HIGH SCHOOL

2  01M450                     EAST SIDE COMMUNITY SCHOOL

3  01M458                      FORSYTH SATELLITE ACADEMY

4  01M509                        MARTA VALLE HIGH SCHOOL

From looking at these rows, we can tell that the DBN in the sat\_results data is just a combination of the CSD and SCHOOL CODE columns in the class\_size data. The main difference is that the DBN is padded, so that the CSD portion of it always consists of two digits. That means we'll need to add a leading 0 to the CSD if the CSD is less than two digits long. Here's a diagram illustrating what we need to do:

CSDPaddedCSD10119192029999

As you can see, whenever the CSD is less than two digits long, we need to add a leading 0. We can accomplish this using the [pandas.Series.apply()](https://pandas.pydata.org/pandas-docs/stable/generated/pandas.Series.apply.html" \t "_blank)method, along with a custom function that:

* Takes in a number.
* Converts the number to a string using the [str()](https://docs.python.org/3/library/stdtypes.html" \l "str" \t "_blank) function.
* Check the length of the string using the [len()](https://docs.python.org/3/library/functions.html" \l "len" \t "_blank) function.
  + If the string is two digits long, returns the string.
  + If the string is one digit long, adds a 0 to the front of the string, then returns it.
    - You can use the string method [zfill()](https://docs.python.org/3/library/stdtypes.html" \l "str.zfill" \t "_blank) to do this.

Once we've padded the CSD, we can use the addition operator (+) to combine the values in the CSD and SCHOOL CODE columns. Here's an example of how we would do this:



dataframe["new\_column"] = dataframe["column\_one"] + dataframe["column\_two"]

And here's a diagram illustrating the basic concept:

