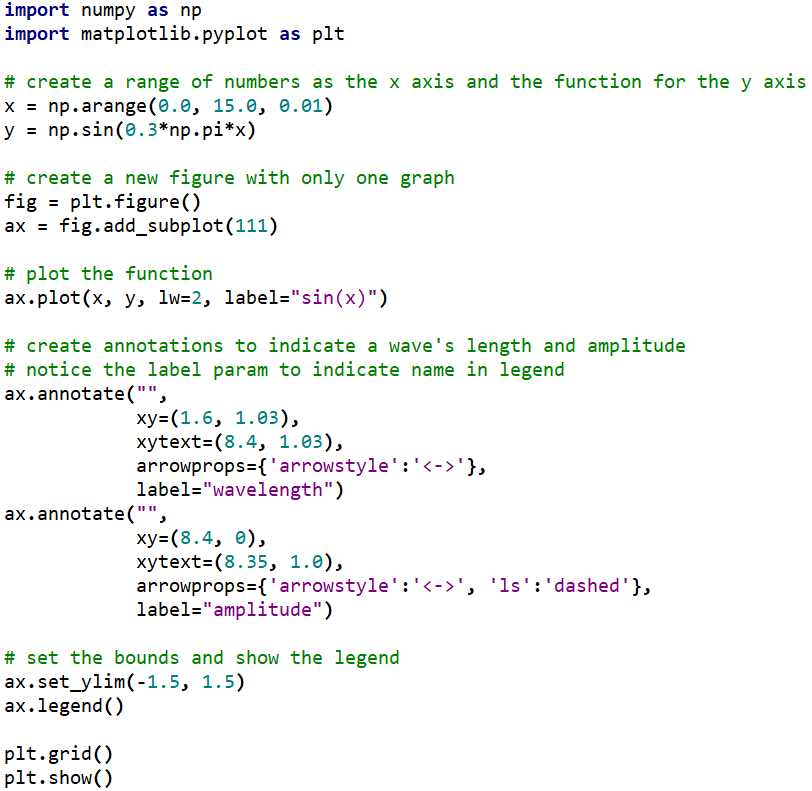
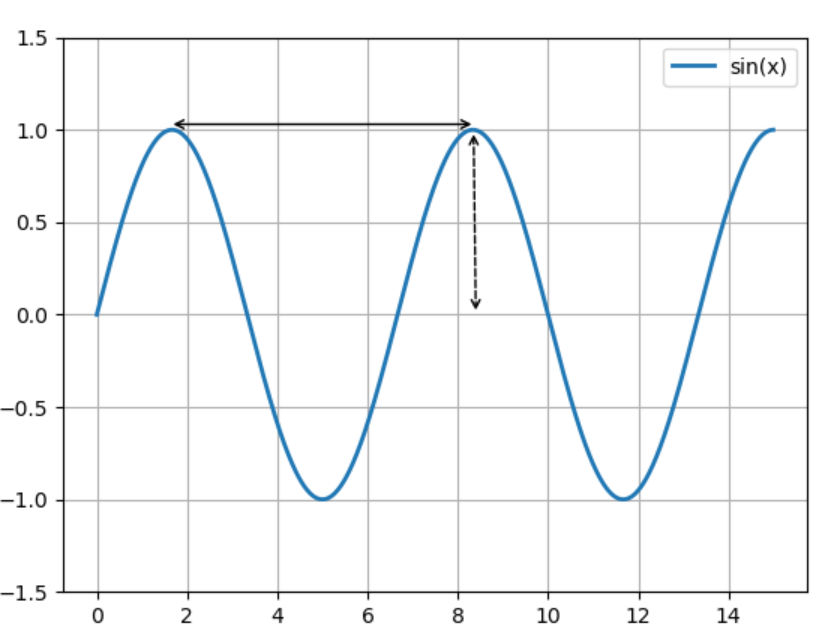
**Bug/Issue:** [Legend does not show ‘annotate’ #8236](https://github.com/matplotlib/matplotlib/issues/8236)

**Code Reproduction for Bug:** /bug\_snippets/legend\_annotate.py





**Description:**

The issue here is that annotations, such as the arrows indicating the wavelength and amplitude, do not appear on the legend. When the legend method gets called, it looks for all the labels that need to be handled and added to the legend, except, annotations do not appear to be included. In addition, the annotate method takes in the keyword argument ‘label’, and thus it should be expected to be included in the legend when it is shown. This issue was probably due to an oversight of the developers since an annotation already provided a text and didn’t need to be included as part of the legend. However, the graph shown above (i.e., when annotation text isn’t used) is an example where having annotations as part of the legend is useful.

**Approach to Solution:**

Fixing this issue will require hours of effort since it requires learning about how the ‘legend’ and ‘legend\_handler’ module works. Moreover, a new legend handler needs to be created so that it will be displayed in the legend properly and in a nice way. The estimated hours needed to fully fix the bug is around 10-15 hours, but this does not include creating tests or validating that the code conforms to the standards set up by the matplotlib team.

The steps needed to fix the issue are as follows:

1. Understand how to properly implement a legend handler.
2. Add ‘annotations’ as part of the list that the legend handles.
3. Make sure that ‘annotations’ are implemented by the legend handler properly.
4. Explore how to display the legend in a nice way and implement it.

Relevant files are:

*legend.py* – add ‘annotations’ as part of the list that a legend handles (line 1300, 1307-1312). Also, must make sure that it passes the ‘has\_handler’ check (line 1326).

*legend\_handler.py –* create a class to handle ‘annotations’ and display it in the legend properly.

*text.py –* Base class for ‘annotations’ and thus should be explored how to add it as part of the list to be handled and displayed to the legend.

The files that will be affected are those described above. No other areas of code should be affected since the legend is independent from the other important code, except *\_axes.py*. Although, ‘texts’ is just being added to that file for the legend to handle, that is it. Regardless, the only code affected will be those that handle adding and displaying to the legend.