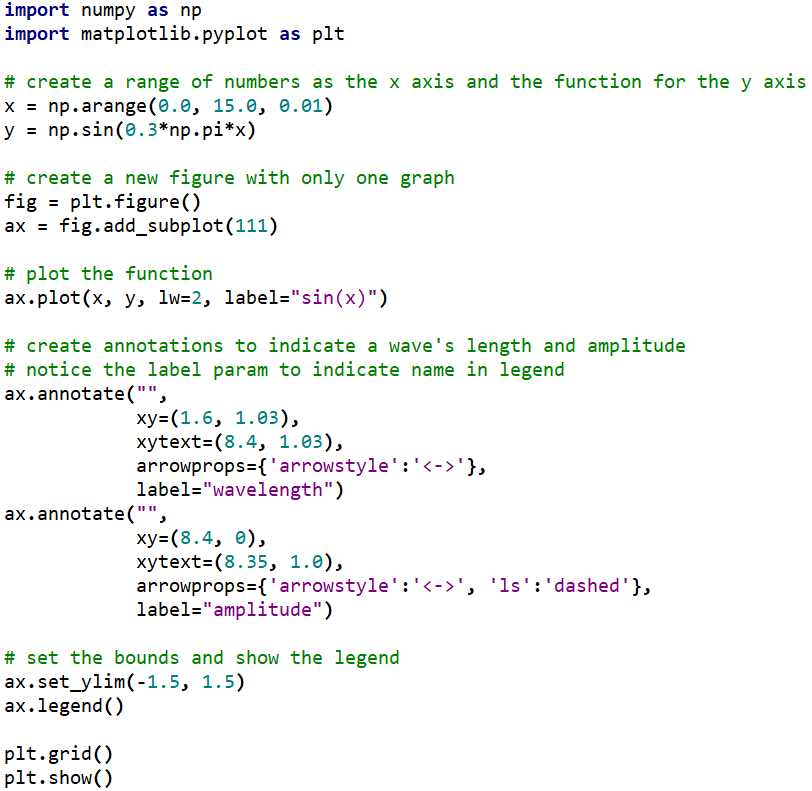
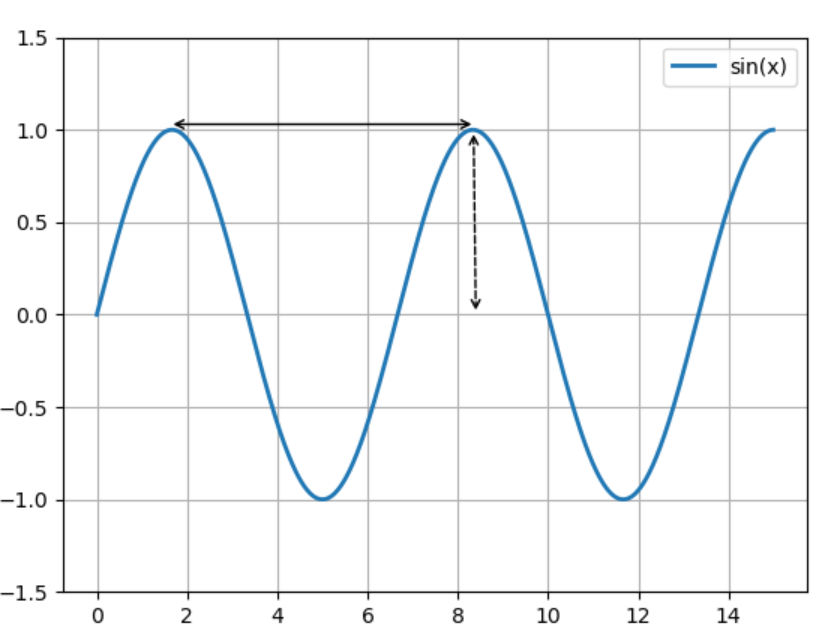
**Bug/Issue:** [Legend does not show ‘annotate’ #8236](https://github.com/matplotlib/matplotlib/issues/8236)

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





**Description:**

Legend items for annotations are currently not operational despite inputs currently being legal for them. Boxing the bi-directional arrows above into a legend to denote amplitude and wavelength is an example of a desirable use case.

When the legend() method on an Axes object gets called, the program eventually adds items to a list of handles to be inserted into the Legend for that Axes (see legend.py:1308, 1313). Annotations, stored in the Axes field texts (as Annotations are a sub-class of the Text class), are not currently added to this list. Additionally, the handler to construct the legend items for Annotations and Texts do not currently exist in the file legend\_handler.py (and are subsequently not mapped in legend.py:805).

**Approach to Solution:**

While we know the exact missing components, fixing this issue will require a sizable amount of effort. The bulk of the job will be in coding the handler to create an adequate construction of legend items for Annotations. There is very little documentation concerning legend items, so we will be required to reverse-engineer the handlers for the other objects like lines. Additionally, this is an issue that will require some decision-making as to how the design of the final legend item should look like.

The exact work necessary is indicated in the UML’s. Yellow portions will require modifications to accommodate the addition of an extra type of legend item. Red portions will require creation from scratch. Note that there is no difference to how the program currently operates when it comes to using other items in legends (the handlers in red are already implemented in those cases), so they were cut out for brevity and to focus attention on what needs to be implemented to get Annotations functional with legends. The sequence diagram consists of two separate operations: commanding the Axes to generate a legend and later actually rendering the result via draw(). Both are called from higher-level operations or via direct advanced users. The legend() command in the given scenario assumes no arguments (in which case it legends every available item to the Axes of interest).

The good news is that this is a mostly self-contained issue. Any further bugs or errors because of attempting to work on this issue is likely to be local to the chain of operations regarding the legend.

The steps needed to fix the issue are as follows:

1. Study the way legend items are constructed in the other handlers and learn the techniques required to position and create the necessary components.
2. Code the handler, add it to the default handler map in legend.py (line 805), and allow Annotations to be returned by the generator Legend.\_get\_legend\_handles().
3. In addition to standard suggested testing, test that legend items for other types of components still work and display properly even when mixed in with annotations.
4. Re-evaluate the design decisions made for the structure of the new legend items.