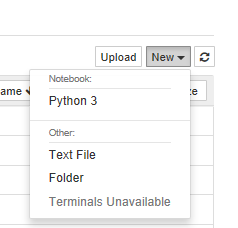
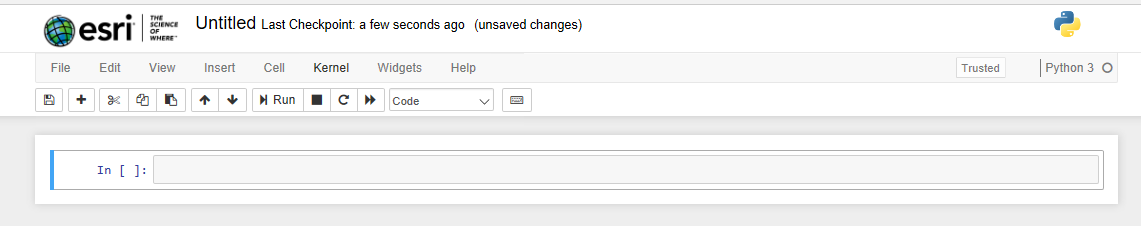
Getting started with the API

1. Launch <http://notebook.esri.com>
2. Click New -> Python 3





1. The first step is to import the ArcGIS API for Python in your notebook

In [1]: **from** **arcgis.gis** **import** GIS

You can 'run' or 'execute' a cell by clicking on the 'run cell' button from the toolbar. Alternately, you can hit Shift + Enter in your keyboard. When a cell is executing the cell number turns to an asterisk (\*) and the circle next to the kernel name turns solid.

1. To get started, create an anonymous connection to ArcGIS Online as shown below:

In [2]:gis = GIS()

1. You can search for content as shown below. Since you are connected to ArcGIS Online and as an anonymous user, the query shown below will return public content that is shared on ArcGIS Online

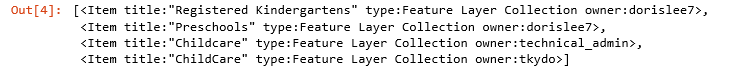
In [3]:public\_content = gis.content.search("ChildCare Singapore", item\_type="Feature Layer", max\_items=5)

1. To see the contents of the variable public\_content you created above, simply type it in a new cell and run it (by hitting Shift + Enter)

\*\*Note\*\*: Expect to see different results when you run this each time on your computer. The code above performs a new search each time you execute it and the search results may vary based on which layers are available publicly.

In [4]:public\_content

Out[4]:



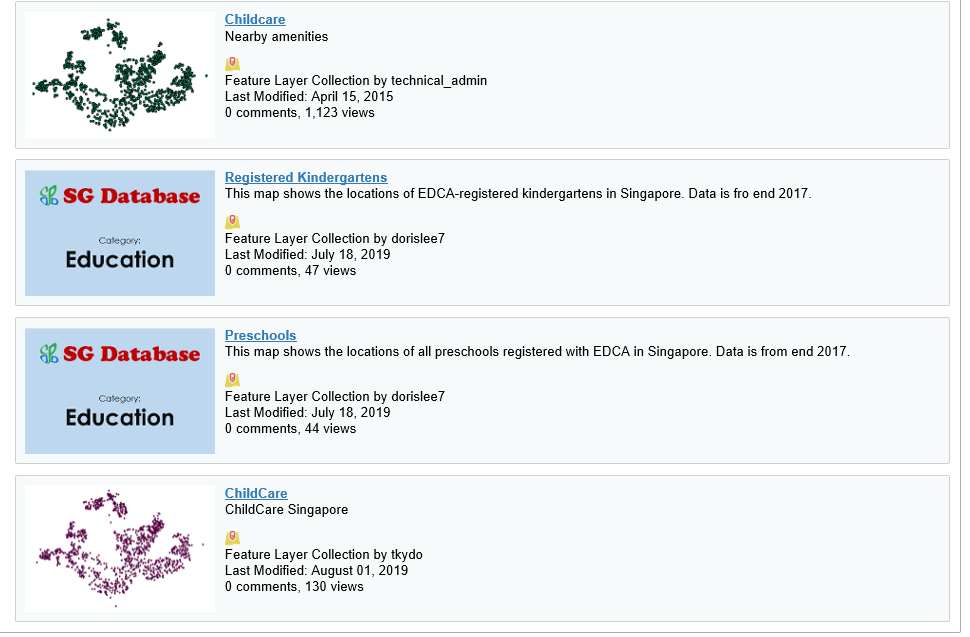
1. The ArcGIS API for Python extends the Jupyter Notebook IDE to display ArcGIS Items in rich HTML notation. Thus, you can loop through each of the items in the search result and display it with thumbnails and metadata as shown below:

In [5]:

**from** **IPython.display** **import** display

**for** item **in** public\_content:

display(item)



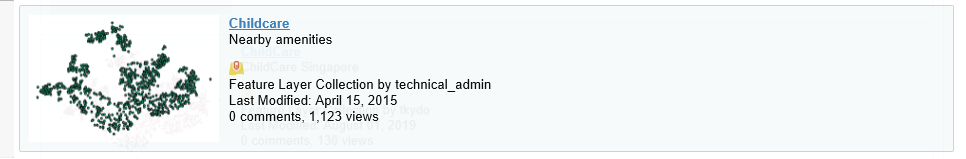
1. Display layers on a map

The ArcGIS API for Python adds a map widget to the Jupyter Notebook. This map widget can be used to display layers from any of the above public content items. Pick an example item:

In [6]:

example\_item = public\_content[0]

display(example\_item)



1. You can then easily pull up a map, as shown below. It may take a few seconds for the map to display the first time.

In [7]:

*#Create a new map object*

map1 = gis.map()

*#Focus the map to the part of the world containing the example item*

map1.extent = example\_item.extent

*#Display the map*

map1

1. You can now add your example item to the map you just created.

In [8]:

map1.add\_layer(example\_item)