Cheatography

GraphSpace Python Client Cheat Sheet

by annaritz via cheatography.com/67574/cs/16994/

Import Statements

```
from graphspace.client import GraphSpace
from graphspace.classes.sgraph import GSGraph
from graphspace.classes.sgroup import GSGroup
from graphspace.classes.slayout import GSLayout
```

Create a Graph

`G = GSGraph()`	Create empty graph
`G.add_node('a')`	Add node to graph
`..., label='A', ...)`	Label to display
`..., popup='html text', ...)`	Popup text
`G.add_edge('a', 'b')`	Add undirected edge to graph
`..., directed=True, ...)`	Directed edge
`..., popup='html text', ...)`	Popup text
`G.set_name('My Sample Graph')`	Add name/title to graph
`G.set_tags(['sample'])`	Add tags to graph

Add Node Style

`G.add_node_style('a', shape='ellipse')`	Node shape
`..., color='blue', ...)`	Node color
`..., color='#A152B8', ...)`	HTML color
`..., bubble='blue', ...)`	Bubble style
`..., height=40, width=20, ...)`	Custom size
`..., style='dashed', ...)`	Border style
`..., border_color='red', ...)`	Border color

Add Node Style (cont)

`..., border_width=2, ...)`	Border width

Arguments should be combined into one `add_node_style` command.

Add Edge Style

`G.add_edge_style('a', 'b', edge_style='dotted')`	Edge style
`..., directed=True, ...)`	Directed edge
`..., color='#F5733', ...)`	HTML color
`..., arrow_shape='tee', ...)`	Arrow shape
`..., width=4, ...)`	Edge width
`..., color='magenta', ...)`	Edge color
`..., arrow_fill='hollow', ...)`	Arrow fill
`..., width=4, ...)`	Edge width

Arguments should be combined into one `add_edge_style` command.

Post Graph to GraphSpace

`gs = GraphSpace('uname', 'pssword')`	Connect to GraphSpace
`graph = gs.post_graph(G)`	Post (new) graph to GraphSpace
`graph = gs.update_graph(G)`	Update (existing) graph to GraphSpace
`gs.delete_graph(graph=graph)`	Delete graph from GraphSpace

Sponsored by **ApolloPad.com**
Everyone has a novel in them. Finish Yours!
https://apollopad.com

GraphSpace Python Client Cheat Sheet

by annaritz via cheatography.com/67574/cs/16994/

Update Graph if it Exists, Otherwise Post it

```
def post(G, gs):
   try:
        graph = gs.upd ate _gr aph(G)
    except:
        graph = gs.pos t_g raph(G)
    return graph
```

Fetch Posted Graph

```
graph = gs.get  _g r a ph  (gr  ap h _ na  me='My Sample Graph')
```

Fetch by graph name

```
graph = gs.get  _g r a ph  (gr  ap h _ id= 12345)
```

Fetch by graph id

```
graph.name
```

Name of graph

```
graph.id
```

ID of graph

Make Graph Public

```
gs.pub lis h_g rap h(g rap h=g ra ph)
```

Make graph public

```
gs.unp ubl ish _gr aph (gr aph =g raph)
```

Make graph private

Groups

```
mygroup = GSGrou p(n ame='My group', descri pti on= 'sample gr oup')
```

Create group

```
group = gs.pos t_g rou p(m ygroup)
```

Add group

```
group = gs.get _gr oup (gr oup _na me='My group')
```

Fetch group

```
gs.sha re_ gra ph( gra ph= graph, group_ nam e='My group')
```

Share graph with group

Groups (cont)

```
gs.uns har e_g rap h(g rap h=g raph, group_ nam e='My group')
```

Layouts

```
L = GSLayout()
```

```
L.set_ nod e_p osi tio n('a', y=38.5, x=67.3)
```

```
L.add_ nod e_s tyle(),L.add_ edg e_s tyle()
```

```
L.set_ nam e('My Sample Layout')
```

```
L.set_ is_ sha red(1)
```

```
layout = graphs pac e.p ost _gr aph _la yout(L, graph
```

```
layout = graphs pac e.g et_ gra ph_ lay out (la you t aph= graph)
```

```
layout1 = graphs pac e.u pda te_ gra ph_ lay out (la
```

Graphs usually have manually-determined layouts from GraphSpace, which are saved through the user interface.

Sponsored by ApolloPad.com
Everyone has a novel in them. Finish Yours!
https://apollopad.com