

Weaviate Cheatsheet

Connection

```
import weaviate; client = weaviate.Client('http://localhost:8080')
```

Connect to Weaviate instance using Python client

```
client = weaviate.Client('http://localhost:8080',  
auth_client_secret=weaviate.AuthApiKey(api_key='your-api-key'))
```

Connect with API key

```
client.is_ready()
```

Check if Weaviate is ready

Schema Management

```
client.schema.create_class({'class': 'Article',  
'properties': [{'name': 'title', 'dataType': ['text']} ]})
```

Create a schema class

```
client.schema.get()
```

Get schema

```
client.schema.delete_class('Article')
```

Delete a class

Data Operations

```
client.data_object.create({'title': 'Example Article' },  
                          'Article')
```

Add data object

```
client.data_object.get_by_id('uuid', 'Article')
```

Get data object by id

```
client.data_object.update({'title': 'Updated Article' },  
                          'Article', 'uuid')
```

Update data object

```
client.data_object.delete('uuid', 'Article')
```

Delete data object

Vector Search

```
client.query.get('Article', ['title']).with_near_text({  
'concepts': ['search term'] }).do()
```

Perform vector search

```
client.query.get('Article', ['title']).with_near_text({  
'concepts': ['search term'] }).with_limit(10).do()
```

Search with limit

```
client.query.get('Article', ['title']).with_near_vector({  
'vector': [0.1, 0.2, ...]  
}).with_additional('distance').do()
```

Search with distance

Batch Operations

```
client.batch.configure(batch_size=100)
```

Create batch

```
client.batch.add_data_object({'title': 'Batch Article'  
, 'Article')
```

Add to batch

```
client.batch.add_data_object({'title': 'Vector Article'  
, 'Article', vector=[0.1, 0.2, ...])
```

Create batch with vectors

```
client.batch.flush()
```

Flush batch

GraphQL Queries

```
client.query.raw('{Get{Article{title}}}')  
Raw GraphQL query
```

```
client.query.get('Article', ['title']).with_hybrid({  
'query': 'search term', 'alpha': 0.5 }).do()
```

Hybrid search