boto Cheat Sheet

S3

```
Basic Modules to import
    from boto.s3.connection import S3Connection
    from boto.s3.key import Key
    import boto
Basic Operations
    Connecting:
        c = S3Connection('<AWS_KEY_ID>', '<AWS_SECRET_KEY>' [,region,...])
        c = boto.connect_s3()
    Creating a bucket:
        c.create_bucket('<bucket-name>')
    Getting a bucket:
        b = c.get_bucket('<bucket-name>')
    Deleting a bucket:
        c.delete_bucket(b)
    Getting a bucket object:
        k = Key(b)
        k.key = 'object-name'
    Downloading said object to file:
        k.get_contents_to_filename('<filename>')
    Downloading string data:
        k.get_contents_as_string()
    Creating a new object key:
        k = b.new_key('<key-name>')
    Uploading file to bucket:
        k.key = 'object-name'
        k.set_contents_from_filename(<path_to_file>)
    Uploading from string:
        k.set_contents_from_string('<string>')
Other Operations:
    Setting Access Controls (bucket-wide):
        b.set_acl('public-read')
        # or any of 'private','public-read-write','authenticated-read'
    Object-specfic Access Control:
        b.set_acl('private', 'confidential.txt')
    Setting Object Metadata:
        k.set_metadata('meta1', 'This is the first metadata value')
        k.set_metadata('meta2', 'This is the second metadata value')
    Getting Object Metadata:
        k.get_metadata('meta1')
        'This is the first metadata value'
```

SQS

```
Basic Imports:
    from boto.sqs.connection    import SQSConnection
    from boto.sqs.message       import Message
    import boto
Basic Operations:
    Connecting:
        c = SQSConnection(AWS_KEY_ID, AWS_SECRET_KEY[, region])
        c = boto.connect_sqs()
    Creating a queue:
        q = c.create_queue('<que_name>'[,visibility_timeout])
    Listing all queues in region:
        qs = c.get_all_queues([prefix='<prefix>'])
    Getting a specific queue:
        q = c.get_queue('<queue_name>')
    Writing messages:
       m = Message()
       m.set_body('<body_text>')
       res = q.write(m)
    Reading Messages:
       rs = q.get_messages([num_messages,...])
       mbody = rs[0].get_body()
    Deleting Messages:
        q.delete_message(m)
    Emptying a queue:
        q.clear() #use carefully
    Deleting (Empty) queues:
        c.delete_queue(q)
```