Python Luigi quick reference

basics

Import the libraries:

```
import luigi
```

Single task program

Define a task:

```
class HelloWorldTask(luigi.Task):
    task_namespace = 'examples'

    def run(self):
        taskName = self.__class__.__name__
        print("%s␣says:␣Hello!" % taskName)
```

Create the main:

```
if __name__ == '__main__':
    luigi.run(['examples.HelloWorldTask',
    '--workers', '1', '--local-scheduler'])
```

Run the program:

```
$ luigi --module examples.hello_world
> examples.HelloWorldTask --local-scheduler
```

Output of a task

Adding an output method to a task:

```
def output(self):
    return luigi.LocalTarget("data/o1.txt")
```

Specifying dependencies

Adding a dependency to a task:

```
def requires(self):
    return WritePipelineTask()
```

Passing parameters to tasks

Specifying an accepted parameter of a task:

```
class MyTask(luigi.Task):
    flow_id = luigi.Parameter()
    ...
```

Invoking a task with a parameter:

```
MyTask(flow_id="something")
```

Wrappers

Create a dummy Task to trigger a multitude of pipelines:

```
class AllTasks(luigi.WrapperTask):

    def requires(self):
        yield SomeReport()
        yield SomeOtherReport()
        ...
```

Resources

Documentation