











First example code-

# -\*- coding: utf-8 -\*-

#

# Licensed to the Apache Software Foundation (ASF) under one

# or more contributor license agreements.  See the NOTICE file

# distributed with this work for additional information

# regarding copyright ownership.  The ASF licenses this file

# to you under the Apache License, Version 2.0 (the

# "License"); you may not use this file except in compliance

# with the License.  You may obtain a copy of the License at

#

#   http://www.apache.org/licenses/LICENSE-2.0

#

# Unless required by applicable law or agreed to in writing,

# software distributed under the License is distributed on an

# "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY

# KIND, either express or implied.  See the License for the

# specific language governing permissions and limitations

# under the License.

"""

### Tutorial Documentation

Documentation that goes along with the Airflow tutorial located

[here](https://airflow.incubator.apache.org/tutorial.html)

"""

from datetime import timedelta

import airflow

from airflow import DAG

from airflow.operators.bash\_operator import BashOperator

# These args will get passed on to each operator

# You can override them on a per-task basis during operator initialization

default\_args = {

    'owner': 'airflow',

    'depends\_on\_past': False,

    'start\_date': airflow.utils.dates.days\_ago(2),

    'email': ['airflow@example.com'],

    'email\_on\_failure': False,

    'email\_on\_retry': False,

    'retries': 1,

    'retry\_delay': timedelta(minutes=5),

    # 'queue': 'bash\_queue',

    # 'pool': 'backfill',

    # 'priority\_weight': 10,

    # 'end\_date': datetime(2016, 1, 1),

    # 'wait\_for\_downstream': False,

    # 'dag': dag,

    # 'adhoc':False,

    # 'sla': timedelta(hours=2),

    # 'execution\_timeout': timedelta(seconds=300),

    # 'on\_failure\_callback': some\_function,

    # 'on\_success\_callback': some\_other\_function,

    # 'on\_retry\_callback': another\_function,

    # 'trigger\_rule': u'all\_success'

}

dag = DAG(

    'tutorial',

    default\_args=default\_args,

    description='A simple tutorial DAG',

    schedule\_interval=timedelta(days=1),

)

# t1, t2 and t3 are examples of tasks created by instantiating operators

t1 = BashOperator(

    task\_id='print\_date',

    bash\_command='date',

    dag=dag,

)

t1.doc\_md = """\

#### Task Documentation

You can document your task using the attributes `doc\_md` (markdown),

`doc` (plain text), `doc\_rst`, `doc\_json`, `doc\_yaml` which gets

rendered in the UI's Task Instance Details page.

![img](http://montcs.bloomu.edu/~bobmon/Semesters/2012-01/491/import%20soul.png)

"""

dag.doc\_md = \_\_doc\_\_

t2 = BashOperator(

    task\_id='sleep',

    depends\_on\_past=False,

    bash\_command='sleep 5',

    dag=dag,

)

templated\_command = """

{% for i in range(5) %}

    echo "{{ ds }}"

    echo "{{ macros.ds\_add(ds, 7)}}"

    echo "{{ params.my\_param }}"

{% endfor %}

"""

t3 = BashOperator(

    task\_id='templated',

    depends\_on\_past=False,

    bash\_command=templated\_command,

    params={'my\_param': 'Parameter I passed in'},

    dag=dag,

)

t1 >> [t2, t3]