future-import-boilerplate

Most Python files should include the following boilerplate at the top of the file, right after the comment header:

```
from __future__ import (absolute_import, division, print_function)
```

This uses Python 3 semantics for absolute vs relative imports, division, and print. By doing this, we can write code which is portable between Python 2 and Python 3 by following the Python 3 semantics.

absolute_import

When Python 2 encounters an import of a name in a file like `import copy` it attempts to load `copy.py` from the same directory as the file is in. This can cause problems if there is a python file of that name in the directory and also a python module in `sys.path` with that same name. In that case, Python 2 would load the one in the same directory and there would be no way to load the one on `sys.path`. Python 3 fixes this by making imports absolute by default. `import copy` will find `copy.py` from `sys.path`. If you want to import `copy.py` from the same directory, the code needs to be changed to perform a relative import: `from . import copy`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\testing\sanity\(ansible-devel)(docs)(docsite)(rst)(dev_guide)(testing)(sanity)future-import-boilerplate.rst, line 26)

Unknown directive type "seealso".

```
.. seealso::

    * `Absolute and relative imports <https://www.python.org/dev/peps/pep-0328>`_
```

division

In Python 2, the division operator (`/`) returns integer values when used with integers. If there was a remainder, this part would be left off (aka, *floor division*). In Python 3, the division operator (`/`) always returns a floating point number. Code that needs to calculate the integer portion of the quotient needs to switch to using the floor division operator (`//`) instead.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\testing\sanity\(ansible-devel)(docs)(docsite)(rst)(dev_guide)(testing)(sanity)future-import-boilerplate.rst, line 38)

Unknown directive type "seealso".

```
.. seealso::

    * `Changing the division operator <https://www.python.org/dev/peps/pep-0238>`_
```

print_function

In Python 2, :func:`python:print` is a keyword. In Python 3, :func:`python3:print` is a function with different parameters. Using this `__future__` allows using the Python 3 print semantics everywhere.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\testing\sanity\(ansible-devel)(docs)(docsite)(rst)(dev_guide)(testing)(sanity)future-import-boilerplate.rst, line 45); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\testing\sanity\(ansible-devel)(docs)(docsite)(rst)(dev_guide)(testing)(sanity)future-import-boilerplate.rst, line 45); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\testing\sanity\(ansible-devel)(docs)(docsite)(rst)(dev_guide)(testing)(sanity)future-import-boilerplate.rst, line 48)

Unknown directive type "seealso".

```
.. seealso::

    * `Make print a function <https://www.python.org/dev/peps/pep-3105>`_
```