



To avoid circular dependencies adhere to the following principles:

- 1.) A script should not import anything above itself (safest).
- 2.) Move shared logic into a third module.
- 3.) Use “late imports” only when necessary.

```
def process():
    from . import b
    return b.do_something()
```

- 4.) Avoid running real logic at import
- 5.) Use interfaces, not concrete implementations.
- 6.) Avoid circular imports in `__init__.py`
- 7.) Use type-checking imports correctly.
- 8.) Use type-checking imports correctly.
- 9.) If modules truly depend on one another merge them.
- 10.) Use dependency direction as a design rule.

*NOTE: SackosArt creates a subclass of the Python dictionary to accomplish inter-script communication*