

tingle tangle

tingle is a library to *tangle* modern markup and data languages into python programs. *tangle* is a concept from literate programming that describes converting document into compiled program languages.

tingle allows folks to think different about their literature that can serve alternative roles as programs and data.

tingle uses **docutils** and **myst_parser** to parse RST and Markdown to intermediate representations as **docutils** documents. from the **docutils** document heuristics are applied to transform the document language to python or yaml files.

tingle is a **pytest** extension that can test literate programs written in markdown or rst formats.

importing literate programs and data

```
import tingle
with tingle.loaders.Markdown():
    ...
```

developing

the tasks below require **doit**, #pip install doit.

```
def task_book():
```

Build the docs using jupyter book.

```
    return dict(actions=["jb build ."])

def task_pdf():
    return dict(actions=["jb build . --builder pdfhtml"], file_dep=['_toc.yml'], targets=
['_build/pdf/book.pdf'])

def task_test():
```

Test the **tingle** package using **pytest**.

```
    return dict(actions=["pytest -pno:pytest-pidgy"])

def task_sphinx_config():
    """translate jupyter book configuration to an extended sphinx configuration.

    extensions += 'autoapi.extension'.split()
    autoapi_type = 'python'
    autoapi_dirs = ['tingle']
    jupyter_execute_notebooks = "off"
    """
    def append():
        with open("conf.py", "a") as f:
            list(map(f.write, task_sphinx_config.__doc__.splitlines(True)[1:]))

    return dict(actions=[
        "jb config sphinx . > conf.py",
        append,
        "pandoc -f markdown -t rst readme.md > readme.rst"], targets=["conf.py", "readme.rst"],
    file_dep=["readme.md"])

def task_sphinx():
    return dict(actions=["sphinx-build . docs/_build"], file_dep=["conf.py"], targets=
["docs/_build"])
```