# Python Package Creation

To publish your Python package, you typically follow these steps:

1. **Prepare Your Package:** Ensure your package is structured correctly with a setup.py file and follows Python packaging standards. Your package should include the necessary files and folders such as \_\_init\_\_.py, README.md, and any other required files.

## Package Structure: -

CAS\_RN\_Validator/

|- README.md

|- setup.py

|-src

|- cas\_rn\_validator/

|- \_\_init\_\_.py

|- validator.py ->

|-validateCAS\_RN()

|-validateCASwithMessage()

|- search.py

|- getChemicalFormulaWithCAS()

|- getSynonymsWithCAS()

|- getCASNumberWithSynonyms()

|- getChemicalFormulaWithSynonyms()

|- getCASNumberWithChemicalFormula()

|- getSynonymsWithChemicalFormula()

|- search\_by\_cas\_number(input\_str)

|- search\_by\_synonyms(input\_str)

|- search\_by\_chemicalformula(input\_str)

|- searchChemicalFormulaWithCAS(input\_str)

|- searchSynonymsWithCAS(input\_str)

|- searchCASNumberWithSynonyms(input\_str)

|- searchChemicalFormulaWithSynonyms(input\_str)

|- searchCASNumberWithChemicalFormula(input\_str)

|- searchSynonymsWithChemicalFormula(input\_str)

|- other\_module.py

1. **Set up setup.py:** Your setup.py file should contain metadata about your package such as its name, version, description, dependencies, and other relevant information.
2. **Build Your Package:** Use setuptools to create a distribution package. You can do this by running python setup.py sdist which will create a source distribution package.

* pip install setuptools
* pip install wheel
* python setup.py sdist bdist\_wheel
* pip install .\casrnvalidator-0.1.0-py3-none-any.whl

# Publish your Python package on PyPI

1. **Create Accounts:** You might need accounts on package distribution platforms like PyPI (Python Package Index) or TestPyPI if you want to test your package first.
2. **Upload to PyPI:** Use twine to upload your package to PyPI. First, install twine if you haven't already (pip install twine). Then, run twine upload dist/\* to upload your package.
3. **Test Your Package:** After uploading your package, you might want to install it in a clean environment to ensure everything works as expected.
4. **Documentation and Support:** Provide proper documentation for your package, including installation instructions, usage examples, and possibly a README file.
5. **Update and Maintain:** Regularly update your package with bug fixes and new features as necessary.