**PyOrator program guidance**

**Python**

PyOrator is being developed using Python version 3.8 from the Python Software Foundation ([PSF](https://www.python.org/)) with the following additional modules install using [pip](https://pypi.org/project/pip/) is the Python package installer:

* openpyxl
* pandas
* pyqt5

PyOrator is written using the PEP 8 [Style Guide for Python](https://www.python.org/dev/peps/pep-0008/)

However, some code stanzas may not yet conform to PEP 8

**Variable naming convention used in PyOrator**

Important suffixes and prefixes:

|  |  |
| --- | --- |
| **prefix** | **meaning** |
| \_adjust \_adj\_ | adjusted |
| appld | applied |
| atmos\_ | atmospheric |
| c\_ | carbon |
| \_coef | coefficient |
| cow\_ | input of carbon from organic waste |
| cumul | cumulative |
| \_dem | demand |
| \_denit | denitrify |
| dpth\_ | depth |
| ex\_ | extra as in extra organic waste |
| factr | factor |
| \_fc | field capacity |
| fert | fertiliser |
| fwd | forward |
| grow\_dds gdds | growing degree days |
| harv\_ | harvest |
| indx | index |
| inorg | inorganic |
| inrt | inert |
| mgmt | management |
| n2 | nitrogen |
| n2o | nitrous oxide |
|  |  |
| \_nitrif | nitrification |
| \_no | nitric oxide |
| no3 | nitrogen trioxide or nitrate radical |
| nut\_ | nutriment |
| opt | optimum |
| ow\_ | organic waste |
| org\_ inorg\_ | organic inorganic |
| pcnt\_ | percentage |
| pet | potential evapotranspiration |
| \_rcoef\_ | response coefficient |
|  |  |
| pi\_ | plant input |
| precip | precipitation |
| prodn | production |
| prop\_ | proportion |
| rat\_ | ratio |
| \_relse | release |
| scle\_factr | Scaling factor |
| \_sply | supply |
| strss | stress |
| \_ss | steady state |
| t\_ | top soil |
| tair | temperature |
| \_typ, \_atyp | typical, atypical |
| uptk | uptake |
| volat | volatilisation |
| wc\_ | water content |
| iws\_ | water stress index |
| \_yld\_ | yield |
|  |  |

Additional variable naming conventions

|  |  |
| --- | --- |
| hydrol\_eff | Hydrologically effective |
| \_fwd | Forward run |
| \_ss | Steady state |
| pettmp | precipitation, PET, temperature |

# test only

# ========

from numpy import arange

for yld in arange(0.3, 0.9, 0.15):

prop\_n\_opt2 = prop\_n\_optimal\_from\_yield(yld, crop\_vars[crop\_curr])

**Soil nitrogen**

Inputs of nitrate

* Atmospheric deposition
* Fertiliser inputs
* Nitrification

Losses of nitrate

* Immobilisation
* Leaching
* Denitrification
* Crop uptake

Inputs of ammonium

* Atmospheric deposition
* Fertiliser inputs
* Mineralisation

Losses of ammonium

* Immobilisation
* Nitrification
* Volatilisation

**Notes on nomenclature**

Plant production: harvestable yield

Harvest index: is one of the parameters in the model and is C in harvested product over the total C in plant

**Notes on carbon pools**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| DPM | pool\_c\_dpm | pi\_to\_dpm | cow\_to\_dpm | c\_loss\_dpm |
| RPM | pool\_c\_rpm | pi\_to\_rpm |  | c\_loss\_rpm |
| HUM | pool\_c\_hum | c\_input\_hum | cow\_to\_hum | c\_loss\_hum |
| BIO | pool\_c\_bio | c\_input\_bio |  | c\_loss\_bio |
| IOM | pool\_c\_iom |  | cow\_to\_iom |  |

**Notes on outputted Excel workbooks**

|  |  |
| --- | --- |
| ora\_classes\_excel\_write.py | Classes defining each workbook |
| ora\_excel\_write.py | Creates sheets using classes in ora\_classes\_excel\_write.py |
| ora\_excel\_write\_cn\_water.py | Writes one workbook for each biophysical group, no class dependencies |