

[Python scripting](#)

- 0

[Using Python](#)

[Importing OpenAlea modules](#)

[Importing visualea nodes and dataflows](#)

[References](#)

Python scripting

OpenAlea packages are available under the python scripting language. This allows to use the components of packages (as in visualea), but also to directly import python modules and get access to all functionalities documented in package's API.

Using Python

Python is a scripting language widely spread in the scientific community. It has a lot of advantages :

It is easy to learn, even for non programmers.
It is an high level language, based on the object paradigm
It is extensible with external libraries
Multi-platform (Linux, Windows, Mac)

You can learn the basics with the [official tutorial](#) or the English/French translated online book [Dive into python](#)

Python scripts could be launched by invoking python from a command line, e.g. :

```
python myScript.py
```

You can also launch a Python interpreter and run or test your script step by step.

The default interpreter could be launch by typing *python* in a terminal (Linux, Mac) or in a DOS windows. Under Windows, you may also use one of the following method:

Use Start Menu ? openalea ? python shell

You can use the default python interpreter GUI : IDLE (start menu ? Python 2.4 ? IDLE)

You can also use more ergonomic user interface, like [ipython](#), that provides usefull functionalities (completion, coloration, execution of block of lines,?)

Importing OpenAlea modules

The magic line which will make available all openalea modules is :

```
import openalea
```

All OpenAlea modules are available in the `openalea` namespace. Refer to the modules documentation to learn how to use them.

You can also write simple python scripts in order to execute the same code several times.

Importing visualea nodes and dataflows

All the openalea nodes available from the package manager can be run from any Python script. To do so, you will have to import the package manager, retrieve the node factory of the component you want to use, and transform it into a regular python function. The `openalea.core` package has functionalities that facilitate this task. For instance, to run the node '+' in the package 'openalea.math', you can type:

```
from openalea.core.alea import load_package_manager, function
```

```
pm = load_package_manager()  
node_factory = pm['openalea.math']['+']  
sum = function(node_factory)
```

```
c = sum(1,2)
```

```
d= sum(a=1,b=2)
```

Note that for dataflows, arguments to the function correspond to ports of the IN node, and returned values to the ports of the OUT node.

References

<http://www.python.org/>