

[User Guide](#)

[Installation](#)

[Using OpenAlea](#)

[Visual programming](#)

[Python scripting](#)

[How to save and run your application ?](#)

## User Guide

edit	1-27	1266392940	documentation:user_guide	Edit
------	------	------------	--------------------------	------

## Installation

For general informations about installation, please read the [Installation guide](#).

If you are looking for a specific package installation, refer to the package documentation.

edit	28-270	1266392940	documentation:user_guide	Edit
------	--------	------------	--------------------------	------

## Using OpenAlea

Models under OpenAlea are in the form of components that could be used as such, or combined together to build user-customised applications. OpenAlea provides two different ways to interact with components:

by visual programming, using visualea

by writing scripts, using a standard python development environment

Both methods allows you to save your application and run it routinely in batch mode. Visual programming is easier to start with, and it allows you to rapidly discover components of a package. Python scripting allows for programming more complex tasks, and provides an access to additional functionalities of models, by importing python modules that compose them.

**Visual programming** OpenAlea provides an high level visual programming interface Visualea

[Beginner's tutorial](#)

edit	1-169	1266392940	documentation:user_guide	Edit
------	-------	------------	--------------------------	------

## How to use the application

[Visualea manual](#)

[Visualea package](#)

## How to create models as dataflow

[Simple example](#)

[Plant modeling demonstration](#)

**Python scripting** OpenAlea modules are available under the python scripting language. This allows to directly use the modules from a Python interpreter.

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](#)

.

## Under Linux

You can use the default interpreter in a terminal : `python`

[ipython](#) provides also some usefull functionality (completion, coloration?).

## Under windows

*Use Start Menu ? openalea ? python shell*

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

The magic line which will make available the 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.

## Howto use dataflow nodes from Python

All the openalea nodes available from the package manager can be run from any Python script. For instance, to run the node '+' in the package 'openalea.math':

```
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)

assert c == d
```

## References

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

## How to save and run your application ?

Under visualea, you application can be saved as a dataflow, and run interactively from visualea

You can also use the 'Export to Application' item in the Workspace menu, and run the exported dataflow apart from visualea

To run your model in batch mode, you can either write python scripts (see above) or use the *alea-script* command in a shell