OPEN SOURCE ENGINEERING

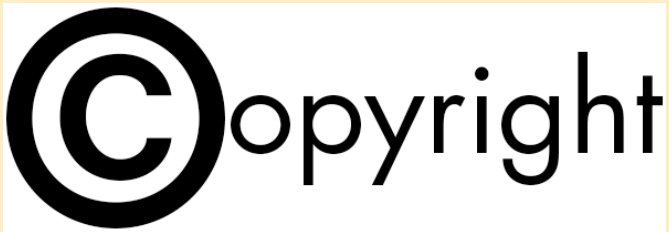
Asst.Prof. Cüneyt Ertal

November 15, 2016

Table of contents

1. Introduction
   - Copyright v.s. Copyleft
   - Free and open-source software (FOSS)
   - Open-Source License
2. Free And Open Source Programming : PYTHON
   - Python Modules : NumPy
   - Python Modules : SciPy
3. Free And Open Source Computer Aided Drawing (CAD) : FreeCAD
4. Free And Open Source Finite Element Analysis (FEM) : Code-Aster

Copyright



Copyright

Copyright is a legal right created by the law of a country that grants the creator of an original work exclusive rights for its use and distribution.

Copyright

Copyright is a legal right created by the law of a country that grants the creator of an original work exclusive rights for its use and distribution.

This is usually only for a limited time. The exclusive rights are not absolute but limited by limitations and exceptions to copyright law, including fair use. A major limitation on copyright is that copyright protects only the original expression of ideas, and not the underlying ideas themselves.

Copyleft

Copyleft

Copyleft is the practice of offering people the right to freely distribute copies and modified versions of a work with the condition that the same rights be preserved in derivative works.

Copyleft

Copyleft is the practice of offering people the right to freely distribute copies and modified versions of a work with the condition that the same rights be preserved in derivative works.

Copyleft is a form of licensing, and can be used to maintain copyright conditions for works ranging from computer software, to documents, to art.

Copyleft

In general, copyright law is used by an author to prohibit recipients from reproducing, adapting, or distributing copies of their work.

Copyleft

In general, copyright law is used by an author to prohibit recipients from reproducing, adapting, or distributing copies of their work.

In contrast, under copyleft, an author may give every person who receives a copy of the work permission to reproduce, adapt, or distribute it, with the accompanying requirement that any resulting copies or adaptations are also bound by the same licensing agreement.

Free and open-source software (FOSS)

Free and open-source software (FOSS) is computer software that can be classified as both free software and open-source software.

Free and open-source software (FOSS)

Free and open-source software (FOSS) is computer software that can be classified as both free software and open-source software. That is, anyone is freely licensed to use, copy, study, and change the software in any way, and the source code is openly shared so that people are encouraged to voluntarily improve the design of the software.

Free and open-source software (FOSS)

Free and open-source software (FOSS) is computer software that can be classified as both free software and open-source software. That is, anyone is freely licensed to use, copy, study, and change the software in any way, and the source code is openly shared so that people are encouraged to voluntarily improve the design of the software.

This is in contrast to proprietary software, where the software is under restrictive copyright and the source code is usually hidden from the users.

Free and open-source software (FOSS)

The benefits of using FOSS can include decreasing software costs, increasing security and stability , protecting privacy, and giving users more control over their own hardware.

Free and open-source software (FOSS)

The benefits of using FOSS can include decreasing software costs, increasing security and stability , protecting privacy, and giving users more control over their own hardware.

Free, open-source operating systems such as Linux are widely utilized today, powering millions of servers, desktops, smartphones (e.g. Android), and other devices.

Free and open-source software (FOSS)

The benefits of using FOSS can include decreasing software costs, increasing security and stability , protecting privacy, and giving users more control over their own hardware.

Free, open-source operating systems such as Linux are widely utilized today, powering millions of servers, desktops, smartphones (e.g. Android), and other devices.

Free software licenses and open-source licenses are used by many software packages.

A program is free software if the program's users have the four essential freedoms:

- **Freedom 0** The freedom to run the program as you wish, for any purpose.
- **Freedom 1** The freedom to study how the program works, and change it so it does your computing as you wish. Access to the source code is a precondition for this.
- **Freedom 2** The freedom to redistribute copies so you can help your neighbor.
- **Freedom 3** The freedom to distribute copies of your modified versions to others.

Open-Source License

An open-source license is a type of license for computer software and other products that allows the source code, blueprint or design to be used, modified and/or shared under defined terms and conditions.

- GNU General Public License (GNU GPL or GPL)

The GNU General Public License (GNU GPL or GPL) is a widely used free software license, which guarantees end users the freedom to run, study, share and modify the software.

Open-Source Alternatives Of Proprietary Softwares

PROPRIETARY SOFTWARE	OPEN-SOURCE ALTERNATIVE(S)
Matlab	Octave, Scilab
Mathematica	Maxima
Solidworks	Freecad
Ansys	Code-Aster
Fluent	OpenFoam

Python Programming Language

Python's name is derived from the television series Monty Python's Flying Circus.

Features

- **Simple :** Python is a simple and minimalistic language. Reading a good Python program feels almost like reading English, although very strict English! This pseudo-code nature of Python is one of its greatest strengths. It allows you to concentrate on the solution to the problem rather than the language itself.

Python Programming Language

- **Easy to Learn :** As you will see, Python is extremely easy to get started with. Python has an extraordinarily simple syntax, as already mentioned.

Python Programming Language

- **Free and Open Source :** Python is an example of a FLOSS (Free/Libre and Open Source Software). In simple terms, you can freely distribute copies of this software, read it's source code, make changes to it, use pieces of it in new free programs, and that you know you can do these things. FLOSS is based on the concept of a community which shares knowledge. This is one of the reasons why Python is so good - it has been created and is constantly improved by a community who just want to see a better Python.

Python Programming Language

- **High-level Language :** When you write programs in Python, you never need to bother about the low-level details such as managing the memory used by your program.

Python Programming Language

- **Portable:** Due to its open-source nature, Python has been ported (i.e. changed to make it work on) to many platforms. All your Python programs can work on any of these platforms without requiring any changes at all if you are careful enough to avoid any system-dependent features.

  You can use Python on Linux, Windows, FreeBSD, Macintosh, Solaris, OS/2, Amiga, Palm OS, VMS, VxWorks, PlayStation, Sharp Zaurus, Windows CE and even PocketPC.

Python Programming Language

- **Interpreted:** A program written in a compiled language like C or C++ is converted from the source language i.e. C or C++ into a language that is spoken by your computer (binary code i.e. 0s and 1s) using a compiler with various flags and options. When you run the program, the linker/loader software copies the program from hard disk to memory and starts running it.

Python Programming Language

Python, on the other hand, does not need compilation to binary. You just run the program directly from the source code.

Internally, Python converts the source code into an intermediate form called bytecodes and then translates this into the native language of your computer and then runs it.

All this, actually, makes using Python much easier since you don't have to worry about compiling the program, making sure that the proper libraries are linked and loaded, etc, etc.

Python Programming Language

- **Object Oriented:** Python supports procedure-oriented programming as well as object-oriented programming.

  In procedure-oriented languages, the program is built around procedures or functions which are nothing but reusable pieces of programs.

  In object-oriented languages, the program is built around objects which combine data and functionality.

Python Programming Language

Python has a very powerful but simplistic way of doing OOP, especially when compared to big languages like C++ or Java.

Python Programming Language

- **Extensible:** If you need a critical piece of code to run very fast or want to have some piece of algorithm not to be open, you can code that part of your program in C or C++ and then use them from your Python program.

Python Programming Language

- **Embeddable:** You can embed Python within your C/C++ programs to give 'scripting' capabilities for your program's users.

Python Programming Language

- **Extensive Libraries:** The Python Standard Library is huge indeed. It can help you do various things involving regular expressions, documentation generation, unit testing, threading, databases, web browsers, CGI, ftp, email, XML, XML-RPC, HTML, WAV files, cryptography, GUI (graphical user interfaces), Tk, and other system-dependent stuff. Remember, all this is always available wherever Python is installed. This is called the **'Batteries Included'** philosophy of Python.

NumPy

NumPy is an extension to the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large library of high-level mathematical functions to operate on these arrays.

Using NumPy in Python gives functionality comparable to MATLAB since they are both interpreted, and they both allow the user to write fast programs as long as most operations work on arrays or matrices instead of scalars.

SciPy

SciPy is a collection of mathematical algorithms and convenience functions built on the Numeric extension for Python. It adds significant power to the interactive Python session by exposing the user to high-level commands and classes for the manipulation and visualization of data. With SciPy, an interactive Python session becomes a data-processing and system-prototyping environment rivaling sytems such as Matlab, IDL, Octave, R-Lab, and SciLab.

Matplotlib

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like wxPython, Qt, or GTK+. There is also a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB. SciPy makes use of matplotlib.

Introduction

FreeCAD is a parametric 3D modeler made primarily to design real-life objects of any size.

The FreeCAD project was started by Jürgen Riegel who is working as computer engineer in DaimlerChrysler in January 2001.

Introduction

Yorik van Havre joined the project in 2008 and started work on the Draft Module. Before that point, there was no way to create 2D geometry through the GUI.

This module was programmed entirely in Python rather than in C++, the core programming language used in FreeCAD. This proved that Python integration was a success and could be used to extend or customize FreeCAD's capabilities.

Features of FreeCAD

- FreeCAD is multi-platform. It runs and behaves exactly the same way on Windows Linux and macOS platforms.
- FreeCAD is a full GUI application.
- FreeCAD also runs as a command line application, with low memory footprint. In command line mode, FreeCAD runs without its interface, but with all its geometry tools. It can be, for example, used as server to produce content for other applications.

Features of FreeCAD

- FreeCAD can be imported as a Python module, inside other applications that can run Python scripts, or in a Python console. Like in console mode, the interface part of FreeCAD is unavailable, but all geometry tools are accessible.
- Workbench concept: In the FreeCAD interface, tools are grouped by workbenches. This allows to display only the tools used to accomplish a certain task, keeping the workspace uncluttered and responsive, and the application fast to load.

Features of FreeCAD

- Parametric associative document objects: All objects in a FreeCAD document can be defined by parameters. Those parameters can be modified on the fly, and recomputed anytime. The relationship between objects is also stored, so modifying one object also modifies its dependent objects.
- Parametric primitive creation (box, sphere, cylinder, etc)
- Graphical modification operations like translation, rotation, scaling, mirroring, offset or shape conversion, in any plane of the 3D space.

Features of FreeCAD

- Graphical creation of simple planar geometry like lines, wires, rectangles, arcs or circles in any plane of the 3D space.
- Modeling with straight or revolution extrusions, sections and fillets.
- Topological components like vertices, edges, wires and planes (via Python scripting).
- Testing and repairing tools for meshes: solid test, non-two-manifolds test, self-intersection test, hole filling and uniform orientation.

Features of FreeCAD

- Annotations like texts or dimensions.
- Undo/Redo framework: Everything is undo/redoable, with access to the undo stack, so multiple steps can be undone at a time.
- Built-in scripting framework: FreeCAD features a built-in Python interpreter, and an API that covers almost any part of the application, the interface, the geometry and the representation of this geometry in the 3D viewer. The interpreter can run single commands up to complex scripts, in fact entire modules can even be programmed completely in Python.

Features of FreeCAD

- Built-in Python console with syntax highlighting, autocomplete and class browser: Python commands can be issued directly in FreeCAD and immediately return results, permitting scriptwriters to test functionality on the fly, explore the contents of the modules and easily learn about FreeCAD internals.
- User interaction mirroring on the console: Everything the user does in the FreeCAD interface executes Python code, which can be printed on the console and recorded in macros.

Features of FreeCAD

- Full macro recording & editing: The Python commands issued when the user manipulates the interface can then be recorded, edited if needed, and saved to be reproduced later.
- Compound (ZIP based) document save format: FreeCAD documents saved with .fcstd extension can contain many different types of information, such as geometry, scripts or thumbnail icons. The .fcstd file is itself a zip container, so a saved FreeCAD file has already been compressed.

Introduction

Code-Aster is a multiphysics finite element code for structural mechanics:

- mechanical phenomena
- thermal phenomena
- acoustics, metallurgy and hydration/drying

Introduction

- It is developed as an in-house application by EDF (Electricity of France) is a French electric utility company.
- Code_Aster contains 1,500,000 lines of source code, most of it in Fortran and Python.

  The documentation of Code_Aster includes more than 14,000 pages and encompasses user's manuals, theory manuals compiling EDF's know-how in mechanics, example problems, verification manuals.

Introduction

- The vast majority of the documentation is in French with computer translations in English also available.
- Code_Aster is used actively by EDF engineers for the design of nuclear plants and for the expertise and the maintenance of power plants and electrical networks.

Salome-Meca

Code_Aster is plugged in a userfriendly pre/postprocessing environment: Salome-Meca.

Salome is an open-source software that provides a generic platform for Pre and Post-Processing for numerical simulation.

$$\boxed{\text{Salome} + \text{Code-Aster} = \text{Salome-Meca}}$$