Macros Well Kept Secrets

LibreOfficiant

Macros Scripting Framework

- **Python**

  in 2008

- **Basic**

  *Office legacy

- **API**

BeanShell

JavaScript

C++

Java

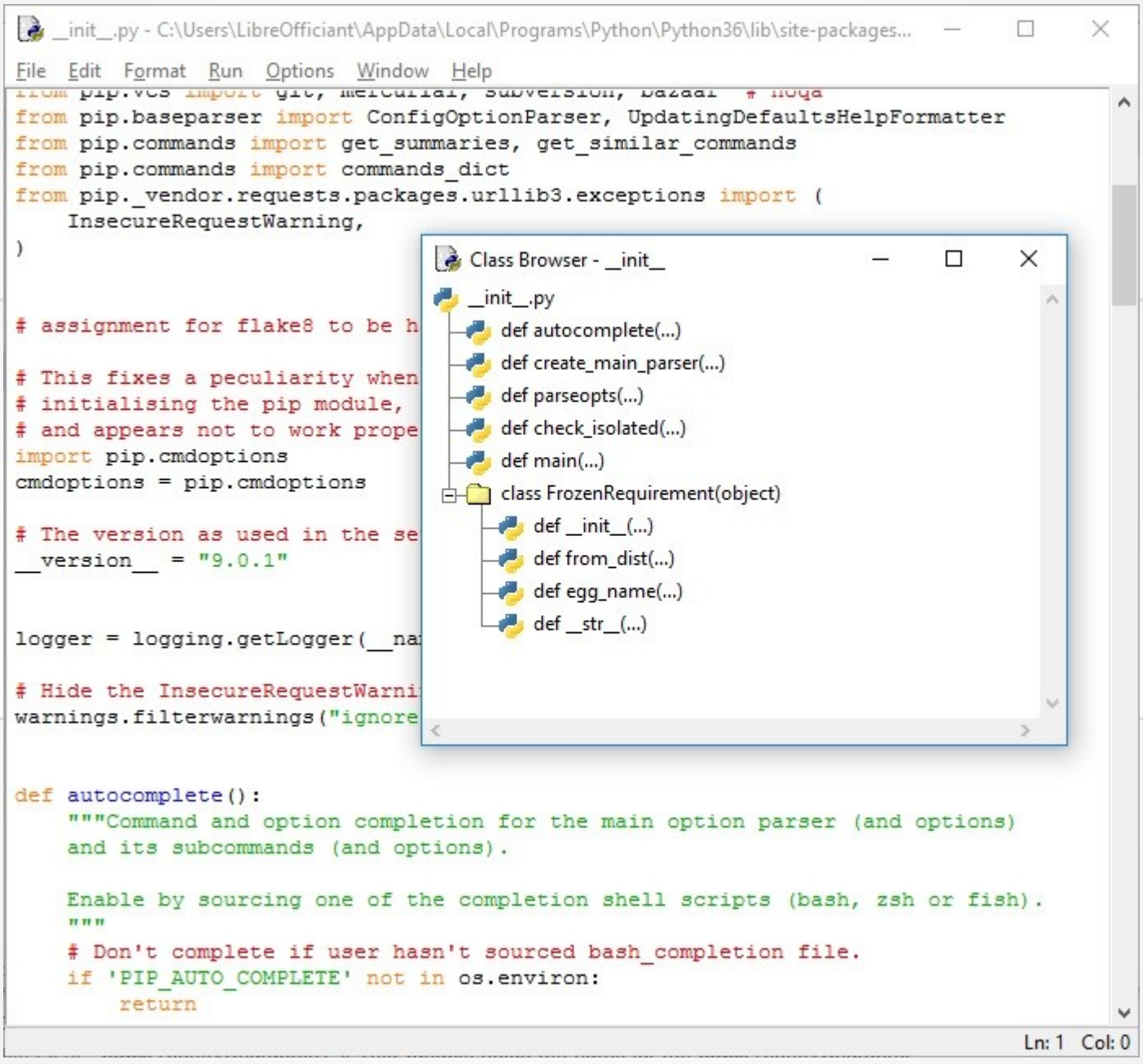
Python

Python

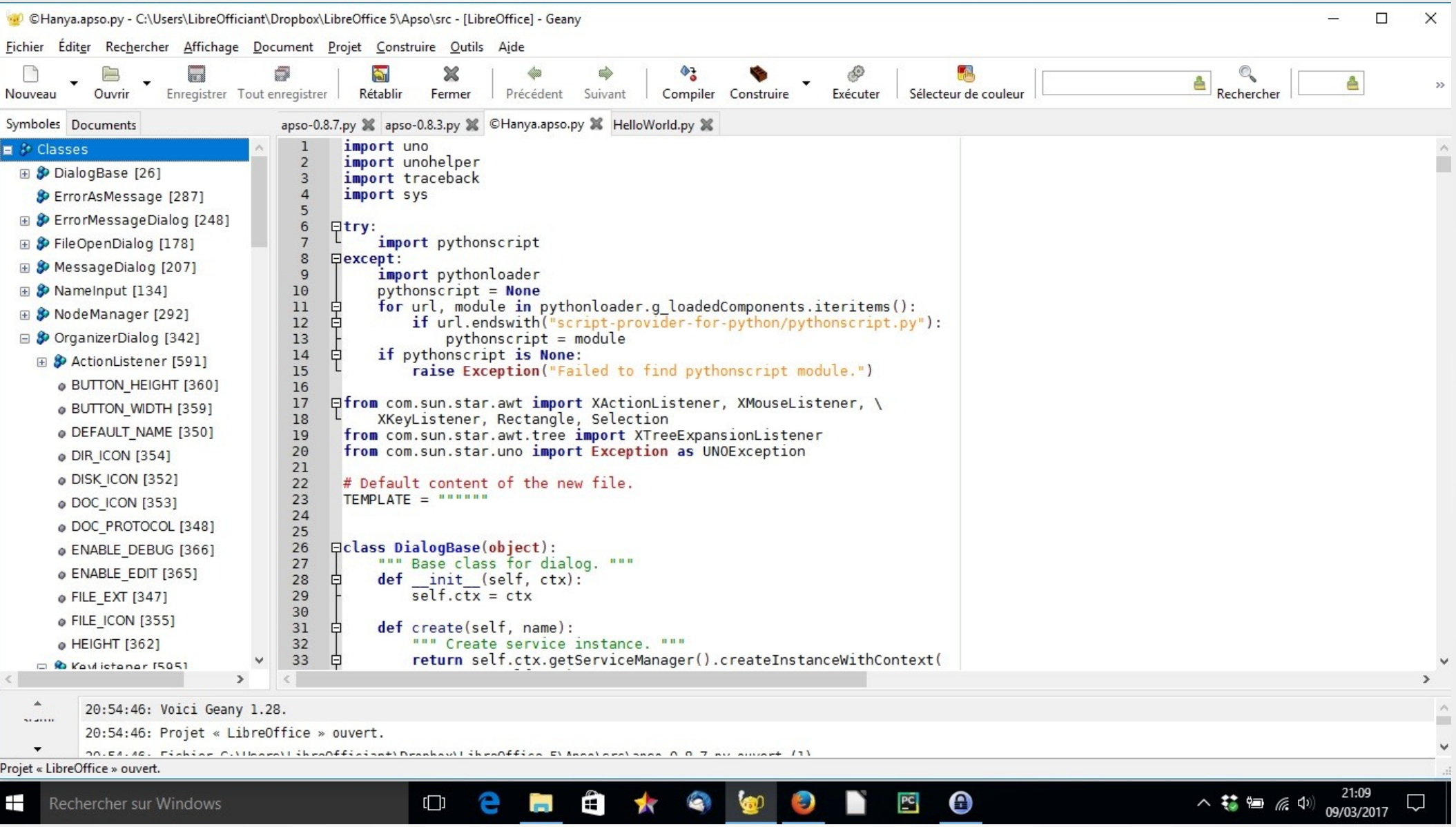
- Source explorer
- Syntax Hiliting
- Code completion
- Debugging / REPL
- Coding guidelines PEP
- Test-driven development
- Version control

- Extensions
  - Mri, Xray
  - APSO
- IDE's
  - Python shell REPL
  - IDE_utils
  - Geany, PyZo, PyCharm

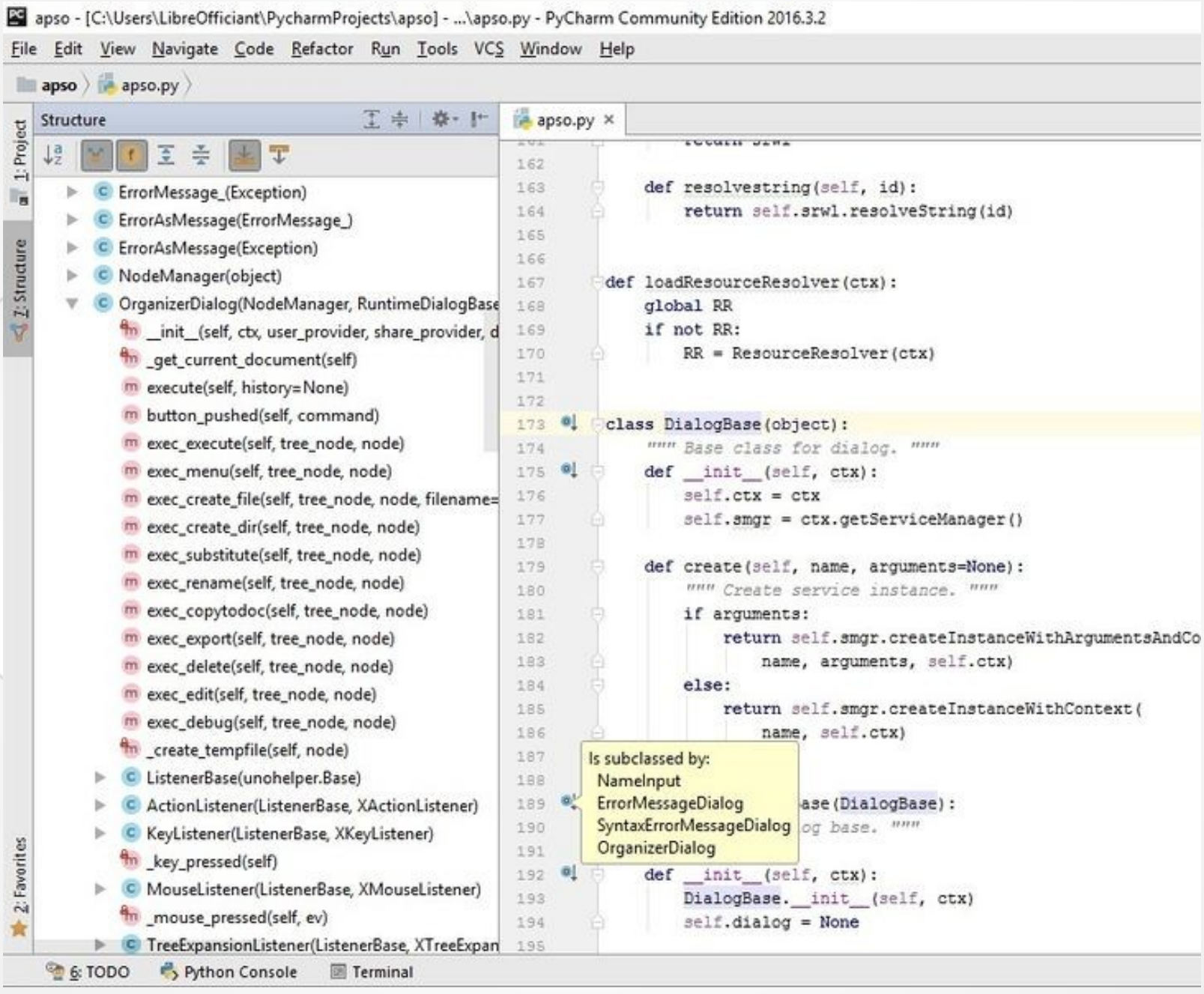
IDLE: class browser syntax hiliting



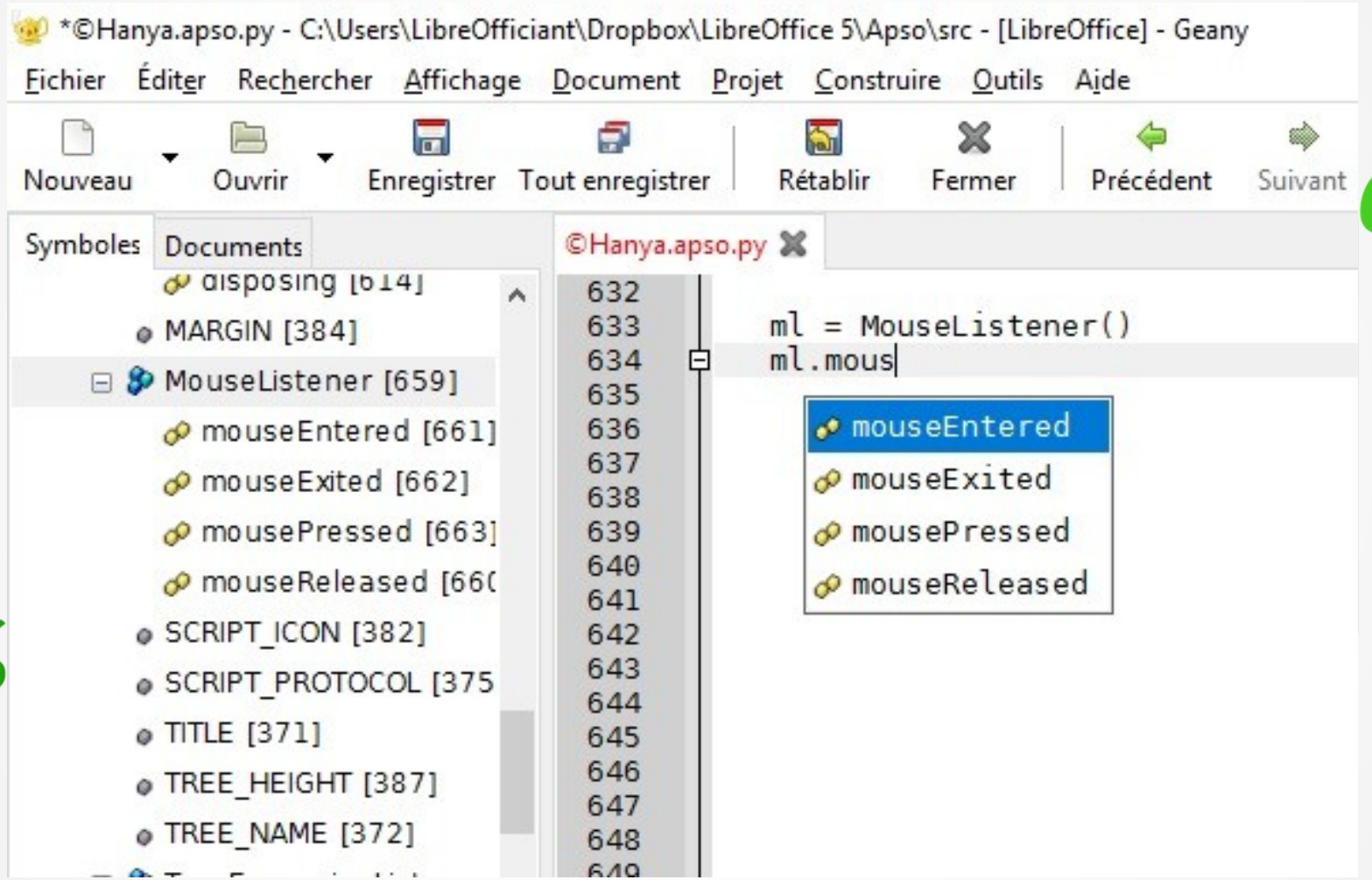
Geany: srce explorer, syntax hiliting, code folding



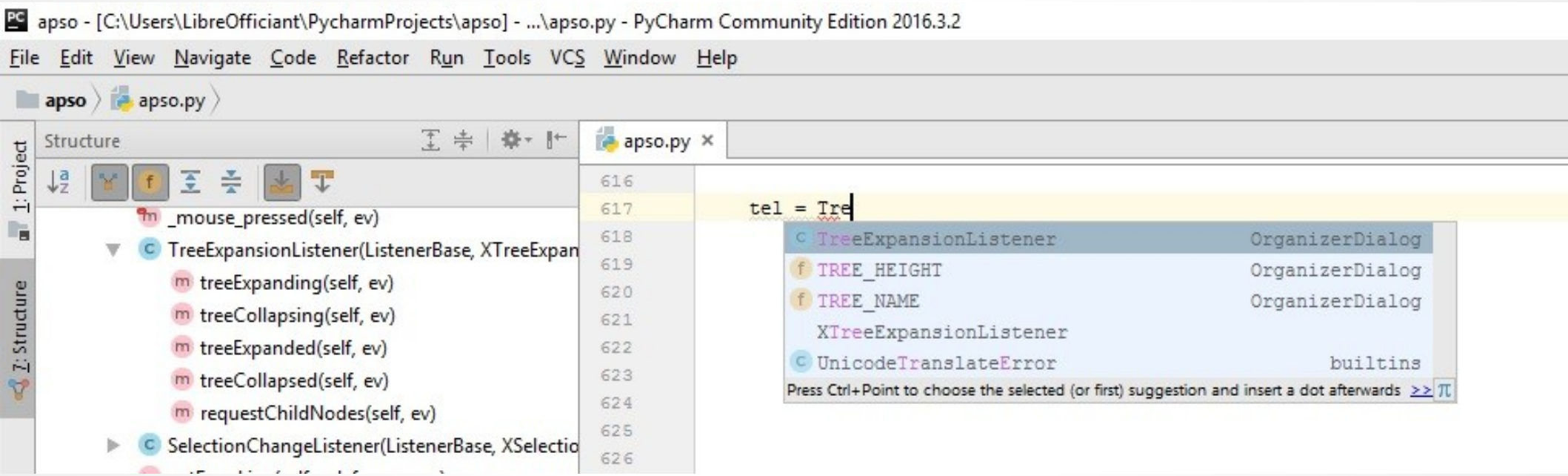
PyCharm: various explorers, syntax hiliting, code folding



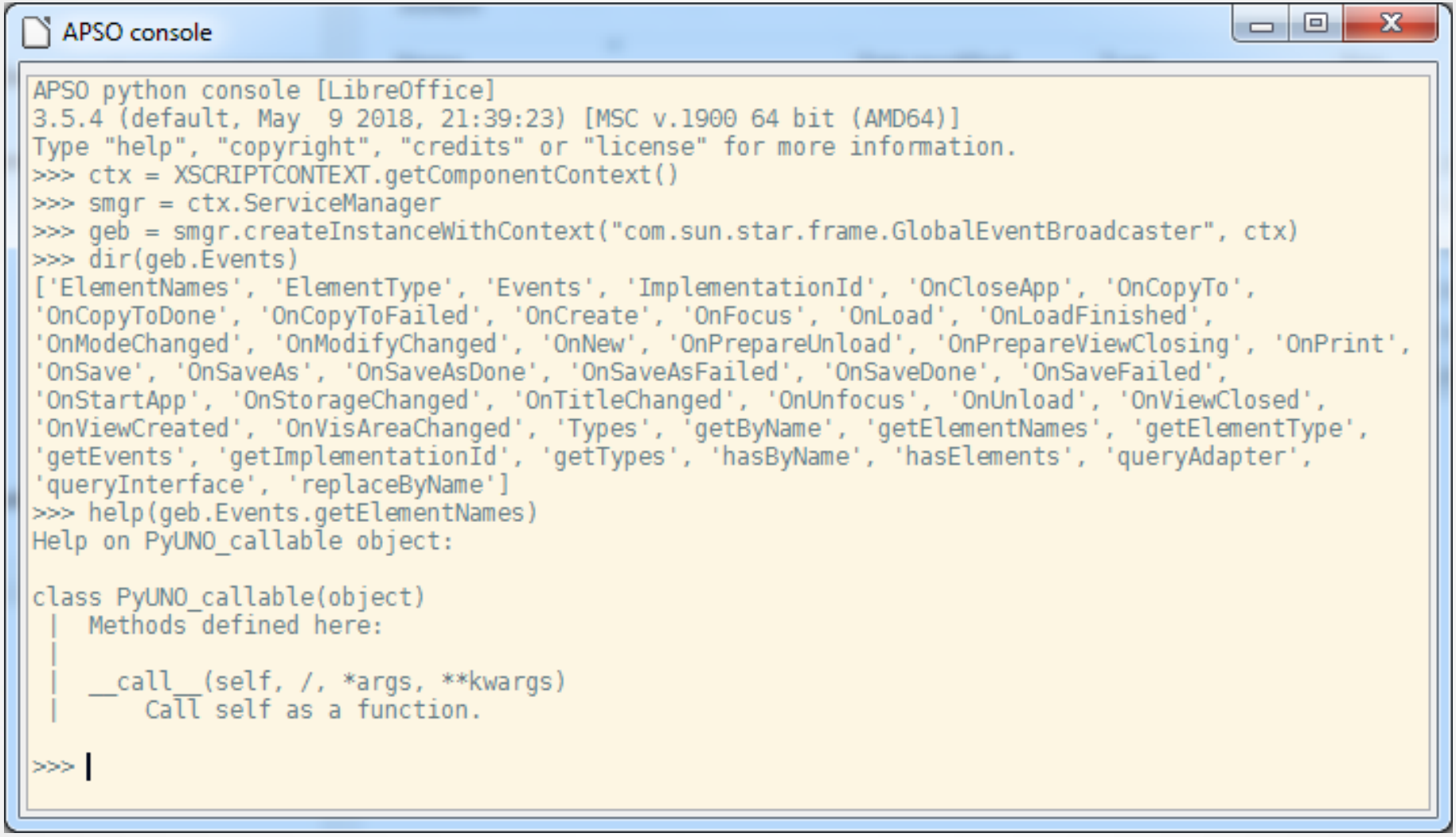
Geany: code completion



PyCharm: code completion



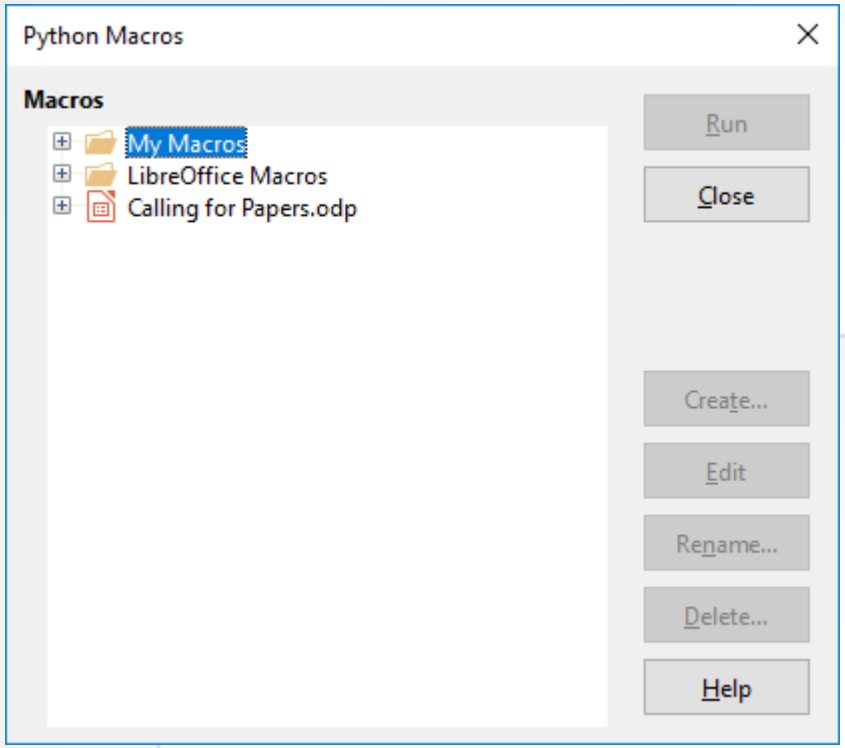
APSO: interactive console



Read Evaluate Print Loop

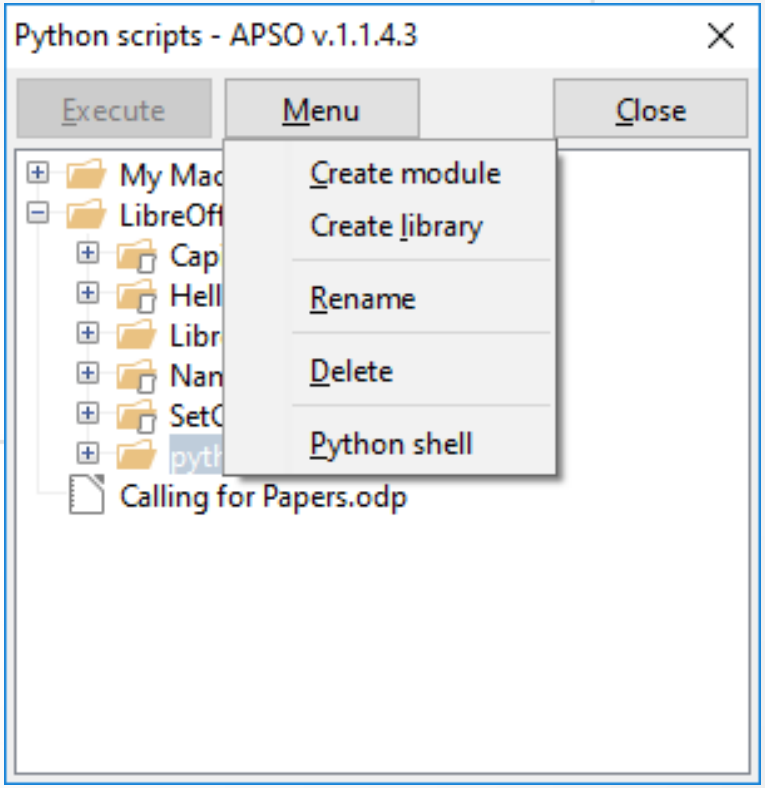
see also Mri, Xray

Altern. Python Script Orgzer



..Less

More although..



LibreOffice Basic

- Enumerations
- Named Arguments
- Variable args list

- Object-oriented Programming
- Custom Error codes

- Opt. Compatible
- Enum
- ParamArray
- Optl arg=default

- Opt. ClassModule
- Err object

Compiler VBA options

Option Compatible

ParamArray

arg = defaultValue

Option ClassModule

Initialize

Terminate

Property Get/Set

VBA Enum.erations

- Long datatypes
  - Enum statement
- User defined typed
  - Type .. End Type statement
- Collection class

Note: Fun with MsgBox & Print

MsgBox buttons= prompt=, title=

Print default=, prompt=, title=

Tip: Spreadsheet Functions

Don’t reinvent the wheel

reuse Calc functions instead !

com.sun.star.sheet.FunctionAccess

Basic: Unit Testing

> Basic Test-Driven Development using Python unittest standard module in conjunction with API Scripting Framework

Rules of thumb

1. Stick to LibreOffice Basic + API
2. Extend with VBA ‘Option Compatible’
3. OOP with ‘Option ClassModule’
4. ‘Option VBASupport 1’ with caution
   - Warning: Affects Basic behaviour
   - Limit it to doc conversion scenarii
5. avoid CompatibilityMode(bool)
   - Bad coding practice

e-References

local/online **Help**

Basic Macros and Scripting

Wiki

Designing & Developing Python Apps

Q & A

Thanks !