



LibreOffice
The Document Foundation



Scripting LibO Python Macros

Macros Well Kept Secrets

Alain Romedenne
LibreOfficiant @sfr.fr

Almería | 11-12 Sept. 2019



Agenda

- Scripting Framework
- Why Python ?
 - Srce Explorer, Syntax Hiliting, Code Completion, Debugging,
 - Coding Guidelines, Test-Driven Dev, Version Control
- Walkthrough Geany, PyZo, PyCharm
- Extension: APSO - JMZ
 - Xray by BM, Mri by Hanya
- IDEs
 - Projects Setup
- Scripting Python Macros
 - Help, Libraries & Modules
 - Programming Basics
 - Dialogs, Listeners, Monitors
 - X-Scripting
- Q & A





Macros Scripting Framework

→ **Python**

– In 2008

BeanShell
JavaScript

→ **Basic / VBA**

– *Office legacy

– In 2004

→ **API**

C++
Java
Python



Agenda

- Scripting Framework
- Why Python ?
 - Srce Explorer, Syntax Hiliting, Code Completion, Debugging,
 - Coding Guidelines, Test-Driven Dev, Version Control
- Walkthrough Geany, PyZo, PyCharm
- Extension: APSO - JMZ
 - Xray by BM, Mri by Hanya
- IDEs
 - Projects Setup
- Scripting Python Macros
 - Help, Libraries & Modules
 - Programming Basics
 - Dialogs, Listeners, Monitors
 - X-Scripting
- Q & A





why Python?

- ✓ Source explorer
 - ✓ Syntax Highlighting
 - ✓ Code completion
 - ✓ Debugging / REPL
 - ✓ Coding guidelines
PEP
 - ✓ Test-driven
development
 - ✓ Version control
- Extensions
 - ◆ Mri, Xray
 - ◆ APSO
 - IDEs
 - Python shell
REPL
 - IDE_utils
 - Geany, PyZo,
PyCharm



Python console

```
C:\Program Files\LibreOffice\program\python.exe
>>> import uno
>>> ctx = uno.getComponentContext()
>>> smgr = ctx.getServiceManager()
>>> dir(smgr)
['AvailableServiceNames', 'DefaultContext', 'ElementType', 'ImplementationId', 'ImplementationName', 'Properties', 'PropertySetInfo', 'SupportedServiceNames', 'Types', 'addEventListener', 'addPropertyChangeListener', 'addVetoableChangeListener', 'createContentEnumeration', 'createEnumeration', 'createInstance', 'createInstanceWithArguments', 'createInstanceWithArgumentsAndContext', 'createInstanceWithContext', 'dispose', 'disposing', 'getAvailableServiceNames', 'getAvailableServiceNames', 'getAvailableServiceNames', 'getElementType', 'getImplementationId', 'getImplementationName', 'getProperties', 'getPropertyByName', 'getPropertySetInfo', 'getPropertyValue', 'getSupportedServiceNames', 'getTypes', 'has', 'hasElements', 'hasPropertyByName', 'initialize', 'insert', 'queryAdapter', 'queryInterface', 'remove', 'removeEventListener', 'removePropertyChangeListener', 'removeVetoableChangeListener', 'setPropertyValue', 'supportsService']
>>>
```

IDEs

	OSs	Languages	FOSS
Geany	Gnu/Linux *BSD MacOS AIX Solaris Windows	43	yes
PyCharm	Gnu/Linux MacOS Windows	english	no
PyZo	Linux OS X Windows	10	yes

Aptana

Liclipse

PyDev for Eclipse



IDLE: class browser, syntax highlighting

```
File Edit Format Run Options Window Help
from pip.vcs import git, mercurial, subversion, bazaar
from pip.baseparser import ConfigOptionParser, UpdatingDefaultsHelpFormatter
from pip.commands import get_summaries, get_similar_commands
from pip.commands import commands_dict
from pip._vendor.requests.packages.urllib3.exceptions import (
    InsecureRequestWarning,
)

# assignment for flake8 to be happy
# This fixes a peculiarity when
# initialising the pip module,
# and appears not to work properly
import pip.cmdoptions
cmdoptions = pip.cmdoptions

# The version as used in the setup.py
__version__ = "9.0.1"

logger = logging.getLogger(__name__)

# Hide the InsecureRequestWarning
warnings.filterwarnings("ignore")

def autocomplete():
    """Command and option completion for the main option parser (and options)
    and its subcommands (and options).

    Enable by sourcing one of the completion shell scripts (bash, zsh or fish).
    """
    # Don't complete if user hasn't sourced bash_completion file.
    if 'PIP_AUTO_COMPLETE' not in os.environ:
        return
```

Class Browser - __init__

- __init__.py
 - def autocomplete(...)
 - def create_main_parser(...)
 - def parseopts(...)
 - def check_isolated(...)
 - def main(...)
 - class FrozenRequirement(object)
 - def __init__(...)
 - def from_dist(...)
 - def egg_name(...)
 - def __str__(...)



Agenda

- Scripting Framework
- Why Python ?
 - Srce Explorer, Syntax Hiliting, Code Completion, Debugging,
 - Coding Guidelines, Test-Driven Dev, Version Control
- Walkthrough Geany, PyZo, PyCharm
- Extension: APSO - JMZ
 - Xray by BM, Mri by Hanya
- IDEs
 - Projects Setup
- Scripting Python Macros
 - Help, Libraries & Modules
 - Programming Basics
 - Dialogs, Listeners, Monitors
 - X-Scripting
- Q & A





Geany: srce explorer, syntax highlighting, code folding

The screenshot displays the Geany IDE interface. The title bar shows the file path: `C:\Users\LibreOfficiant\Dropbox\LibreOffice 5\Apso\src - [LibreOffice] - Geany`. The menu bar includes `Fichier`, `Éditer`, `Rechercher`, `Affichage`, `Document`, `Projet`, `Construire`, `Outils`, and `Aide`. The toolbar contains icons for `Nouveau`, `Ouvrir`, `Enregistrer`, `Tout enregistrer`, `Rétablir`, `Fermer`, `Précédent`, `Suivant`, `Compiler`, `Construire`, `Exécuter`, and `Sélecteur de couleur`. The left sidebar shows a `Classes` tree with various classes like `DialogBase`, `ErrorAsMessage`, etc. The main editor window shows a Python file with the following code:

```
1 import uno
2 import unohelper
3 import traceback
4 import sys
5
6 try:
7     import pythonscript
8 except:
9     import pythonloader
10    pythonscript = None
11    for url, module in pythonloader.g_loadedComponents.iteritems():
12        if url.endswith("script-provider-for-python/pythonscript.py"):
13            pythonscript = module
14    if pythonscript is None:
15        raise Exception("Failed to find pythonscript module.")
16
17 from com.sun.star.awt import XActionListener, XMouseListener, \
18     XKeyListener, Rectangle, Selection
19 from com.sun.star.awt.tree import XTreeExpansionListener
20 from com.sun.star.uno import Exception as UNOException
21
22 # Default content of the new file.
23 TEMPLATE = """
24
25
26 class DialogBase(object):
27     """ Base class for dialog. """
28     def __init__(self, ctx):
29         self.ctx = ctx
30
31     def create(self, name):
32         """ Create service instance. """
33         return self.ctx.getServiceManager().createInstanceWithContext(
```

The status bar at the bottom shows the following messages:

```
20:54:46: Voici Geany 1.28.
20:54:46: Projet « LibreOffice » ouvert.
20:54:46: Fichier C:\Users\LibreOfficiant\Dropbox\LibreOffice 5\Apso\src\apso-0.8.7.py ouvert (1)
```



PyCharm: various explorers, syntax highlighting, code folding

The screenshot displays the PyCharm IDE interface. At the top, the title bar reads "apso - [C:\Users\LibreOfficiant\PycharmProjects\apso] - ...\apso.py - PyCharm Community Edition 2016.3.2". The menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The main workspace is divided into three panes:

- Structure:** Shows a tree view of the project's classes and methods. The "OrganizerDialog" class is expanded, listing methods such as `__init__`, `_get_current_document`, `execute`, `button_pushed`, `exec_execute`, `exec_menu`, `exec_create_file`, `exec_create_dir`, `exec_substitute`, `exec_rename`, `exec_copypatodoc`, `exec_export`, `exec_delete`, `exec_edit`, `exec_debug`, `_create_tempfile`, `ListenerBase`, `ActionListener`, `KeyListener`, `MouseListener`, and `TreeExpansionListener`.
- Code Editor:** Displays the source code for `apso.py`. The `DialogBase` class is highlighted in yellow. The code includes a `resolvestring` method, a `loadResourceResolver` method, and the `DialogBase` class definition with its `__init__` and `create` methods.
- Tooltip:** A yellow tooltip is positioned over the `DialogBase` class name, listing its subclasses: `NameInput`, `ErrorMessageDialog`, `SyntaxErrorMessageDialog`, and `OrganizerDialog`.

The bottom status bar shows "6: TODO", "Python Console", and "Terminal".



Geany: code completion

The screenshot shows the Geany IDE interface. The title bar reads: *©Hanya.apso.py - C:\Users\LibreOfficiant\Dropbox\LibreOffice 5\Apso\src - [LibreOffice] - Geany. The menu bar includes: Fichier, Éditer, Rechercher, Affichage, Document, Projet, Construire, Outils, Aide. The toolbar contains icons for Nouveau, Ouvrir, Enregistrer, Tout enregistrer, Rétablir, Fermer, Précédent, and Suivant. The left sidebar shows a 'Symboles' pane with a tree view of the project structure. The 'Documents' folder is expanded, showing a 'MouseListener' class at line 659. The main editor window shows the following code:

```
632  
633  
634 ml = MouseListener()  
635 ml.mous|
```

A code completion popup menu is visible, listing the following methods:

- mouseEntered
- mouseExited
- mousePressed
- mouseReleased



PyCharm: code completion

PC apso - [C:\Users\LibreOfficiant\PycharmProjects\apso] - ...\apso.py - PyCharm Community Edition 2016.3.2

File Edit View Navigate Code Refactor Run Tools VCS Window Help

apso apso.py

Structure

- 1: Project
 - 1: Structure
 - treeExpansionListener(ListenerBase, XTreeExpansionListener)
 - _mouse_pressed(self, ev)
 - treeExpanding(self, ev)
 - treeCollapsing(self, ev)
 - treeExpanded(self, ev)
 - treeCollapsed(self, ev)
 - requestChildNodes(self, ev)
 - SelectionChangeListener(ListenerBase, XSelectionChangeListener)

apso.py x

616
617
618
619
620
621
622
623
624
625
626

tel = Tre

TreeExpansionListener	OrganizerDialog
TREE_HEIGHT	OrganizerDialog
TREE_NAME	OrganizerDialog
XTreeExpansionListener	
UnicodeTranslateError	builtins

Press Ctrl+Point to choose the selected (or first) suggestion and insert a dot afterwards >> π

PyZo: debugging

IDE_utils.py (C:\Users\LibreOfficiant\Documents\ide_utils\IDE_utils.py) - Interactive Editor for Python

File Edit View Settings Shell Run Tools Help

```
501
502 sOffice = os.path.join(sOffice, "soffice")
503 if sys.platform.startswith("win"):
504     sOffice += ".exe"
505
506 options = ['-accept=pipe,name=OfficeHelper;urp;',
507            '-nodefault', '-nologo']
508 if sys.version_info.major == 3: # OpenOffice uses 2.x
509     options = ['--accept=pipe,name=OfficeHelper;urp;',
510              '--nodefault', '--nologo']
511
512 oh = {sOffice: options}
513 start(soffice=oh)
514 ctx = connect(pipe='OfficeHelper')
515
516 return ctx
517
518 logging.info('BOOTSTRAPPING (Libre|Open)Office instance..')
519 if _EMULATE_OFFICEHELPER:
520     _ctx = _bootstrap()
521 else:
522     _ctx = officehelper.bootstrap()
523
524 ScriptContext.pool['officehelper'] = _ctx # Force service termination
525 XSCRIPTCONTEXT = ScriptContext(_ctx)
526 ''' Substitute XSCRIPTCONTEXT built-in '''
527
528 # =====
529 # TERMINATE
530 # =====
531
532 # atexit.register(stop)
533 ''' Make sure services are released
534
535 Whenever ScriptContext() is used but Runner() isn't, all services
536 that were connected to, including officehelper random pipe, have to
537 be terminated using stop() routine.
538
539 ''' # STOP 'officehelper' (Libre|Open)Office instances
540 @atexit.register
541 def stop():
542     """ STOP all (Libre|Open)sOffice instances """
543     logging.info('EXITing '+_name_)
544     try:
545         if ScriptContext.pool: # non-empty pool
546             terminate_desktops(ScriptContext.pool)
547     except (DisposedException) as e:
548         ''' URP bridge already released '''
```

Shells

Python 3.5.5 (default, Jul 28 2018, 07:26:52) on Windows (64 bits).
This is the Pyzo interpreter.
Type 'help' for help, type '?' for a list of *magic* commands.

Kernel process terminated for restart. (0)

Python 3.5.5 (default, Jul 28 2018, 07:26:52) on Windows (64 bits).
This is the Pyzo interpreter.
Type 'help' for help, type '?' for a list of *magic* commands.
Running script: "C:\Users\LibreOfficiant\Documents\ide_utils\IDE_utils.py"

bootstrap

- _DEBUG
- _DELAYS
- _EMULATE_OFFICEHELPER
- _INFO
- _SECONDS

Workspace

Name	Type	Repr
BootstrapExcep...	type	<class 'officehelper.BootstrapException'>
CONNECT_DELA...	tuple	<tuple with 7 elements>
CONNECT_EXC...	type	<class 'uno.com.sun.star.connection.NoConnectException'>
CONNECT_REP...	function	<function <lambda> at 0x00000255E6899378>
DisposedExcept...	type	<class 'uno.com.sun.star.lang.DisposedException'>
NoConnectExce...	type	<class 'uno.com.sun.star.connection.NoConnectException'>
RUNNERS	str	'Runners.json'
Runner	type	<class '__main__.Runner'>
ScriptContext	type	<class '__main__.ScriptContext'>

Source structure

```
__enter__0
__exit__0
_accept2Uno0
_isOfficeBinary0
_read_service0
connect0
start0
stop0
```

File Browser

Click star to bookmark current dir

C:\Users\LibreOfficiant\Documents\ide_utils

- .git
- .idea
- .vscode
- docs
- Guides



PyCharm: debugging

The screenshot shows the PyCharm IDE with a Python script named 'Bridge.py' open. The script is being debugged, and the debugger is currently paused at line 199. The code in the script is as follows:

```
192 # Quick n Dirty Testing
193 if __name__ == '__main__':
194
195     ctx = XSCRIPTCONTEXT.ComponentContext ctx: None: pyuno object (com.sun.star.uno.XInterface)0x2f902bc{, ...
196     dsk = XSCRIPTCONTEXT.getDesktop() dsk: None: pyuno object (com.sun.star.uno.XInterface)0x2fb361c(implemen
197     doc = XSCRIPTCONTEXT.getDocument() # May return None doc: None
198     obj = createObject("com.sun.star.frame.Desktop") obj: None: pyuno object (com.sun.star.uno.XInterface)0
199     print(ctx)
200     print(dsk) # com.sun.star.uno.XInterface
201     print(doc) # com.sun.star.lang.XComponent - base, calc, draw, basicIDE, impress, math, writer, ...
```

The debugger interface shows the following information:

- Debugger:** Console, Frames, Variables
- Frames:** MainThread, <module>, Bridge.py:199, execfile, _pydev_execfile.py:18, run, pydevd.py:974, <module>, pydevd.py:1596
- Variables:**
 - Special Variables
 - NOCONNECT_EXCEPTION = {str} 'Failed to connect to %s on host=%s,port= %d,pipe= %s'
 - NODESKTOP_EXCEPTION = {str} 'Failed to create % desktop on host=%s,port= %d,pipe= %s'
 - XSCRIPTCONTEXT = {OfficeSession} <__main___.OfficeSession object at 0x02D05DB0>
 - _MY_HOST = {str} 'localhost'
 - _MY_OFFICE = {str} 'LibreOffice'
 - _MY_PIPE = {str} 'LibreOffice'
 - _MY_PORT = {int} 2017
 - _NOSTARTUP_EXCEPTION_MSG = {str} 'Failed to start %s on host= %s,port= %d,pipe= %s'
 - _OfficeSession = {Singleton} <class '.__main___.OfficeSession'>
 - _UNSUPPORTED_PLATFORM_MSG = {str} '\\'%s\\' platform is not supported yet.'
 - _VERBOSE = {bool} False
 - ctx = {pyuno} None: pyuno object (com.sun.star.uno.XInterface)0x2f902bc{, supportedInterfaces={com.sun.star.uno.XComponentCo... View
 - doc = {NoneType} None
 - dsk = {pyuno} None: pyuno object (com.sun.star.uno.XInterface)0x2fb361c(implementationName=com.sun.star.comp.framework.De... View
 - obj = {pyuno} None: pyuno object (com.sun.star.uno.XInterface)0x2fb361c(implementationName=com.sun.star.comp.framework.De... View
 - pyuno_runtime = {pyuno_runtime} <pyuno_runtime object at 0x02EDC1C0>

The status bar at the bottom indicates line 199:1, CRLF, UTF-8.



Geany, PyCharm: Unit Testing





Pycharm: PEP compliance

```
PC IDEs [C:\Users\LibreOfficiant\PycharmProjects\IDEs] - C:\Program Files\LibreOffice\program\pythonscript.py - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
C:\Program Files> LibreOffice> program> pythonscript.py test
pythonloader.py x pythonscript.py x
IDEs C:\Users\LibreOfficiant\PycharmProjects\IDEs
test.py
External Libraries
Scratches and Consoles
485     entry.module.__file__ = url
486     self.modules[url] = entry
487     log.debug("mapped " + url + " to " + str(entry.module))
488     return entry.module
489
490     -----
491     def isScript( candidate ):
492         ret = False
493         if isinstance( candidate, type(isScript) ):
494             ret = True
495         return ret
496
497     -----
498     class ScriptBrowseNode( unohelper.Base, XBrowseNode , XPropertySet, XInvocation, XAct
499     def __init__( _self, provCtx, uri, fileName, funcName ):
500         self.fileName = fileName
501
502         raise
503         return ret
504
505     def hasChildNodes( self ):
506         try:
507             return len(self.getChildNodes()) > 0
508         except:
509             return False
510
511     def getType( _self ):
512
513     def hasChildNodes( self ):
514         return False
515
516     def getType( _self ):
517         return SCRIPT
518
519     -----
520     def getType( _self ):
521         return SCRIPT
522
523     -----
524     def getType( _self ):
525         return SCRIPT
526
527     -----
528     def getType( _self ):
529         return SCRIPT
530
531     -----
532     def getType( _self ):
533         return SCRIPT
534
535     -----
536     def getType( _self ):
537         return SCRIPT
538
539     -----
540     def getType( _self ):
541         return SCRIPT
542
543     -----
544     def getType( _self ):
545         return SCRIPT
546
547     -----
548     def getType( _self ):
549         return SCRIPT
550
551     -----
552     def getType( _self ):
553         return SCRIPT
554
555     -----
556     def getType( _self ):
557         return SCRIPT
558
559     -----
560     def getType( _self ):
561         return SCRIPT
562
563     -----
564     def getType( _self ):
565         return SCRIPT
566
567     -----
568     def getType( _self ):
569         return SCRIPT
570
571     -----
572     def getType( _self ):
573         return SCRIPT
574
575     -----
576     def getType( _self ):
577         return SCRIPT
578
579     -----
580     def getType( _self ):
581         return SCRIPT
582
583     -----
584     def getType( _self ):
585         return SCRIPT
586
587     -----
588     def getType( _self ):
589         return SCRIPT
590
591     -----
592     def getType( _self ):
593         return SCRIPT
594
595     -----
596     def getType( _self ):
597         return SCRIPT
598
599     -----
600     def getType( _self ):
601         return SCRIPT
602
603     -----
604     def getType( _self ):
605         return SCRIPT
606
607     -----
608     def getType( _self ):
609         return SCRIPT
610
611     -----
612     def getType( _self ):
613         return SCRIPT
614
615     -----
616     def getType( _self ):
617         return SCRIPT
618
619     -----
620     def getType( _self ):
621         return SCRIPT
622
623     -----
624     def getType( _self ):
625         return SCRIPT
626
627     -----
628     def getType( _self ):
629         return SCRIPT
630
631     -----
632     def getType( _self ):
633         return SCRIPT
634
635     -----
636     def getType( _self ):
637         return SCRIPT
638
639     -----
640     def getType( _self ):
641         return SCRIPT
642
643     -----
644     def getType( _self ):
645         return SCRIPT
646
647     -----
648     def getType( _self ):
649         return SCRIPT
650
651     -----
652     def getType( _self ):
653         return SCRIPT
654
655     -----
656     def getType( _self ):
657         return SCRIPT
658
659     -----
660     def getType( _self ):
661         return SCRIPT
662
663     -----
664     def getType( _self ):
665         return SCRIPT
666
667     -----
668     def getType( _self ):
669         return SCRIPT
670
671     -----
672     def getType( _self ):
673         return SCRIPT
674
675     -----
676     def getType( _self ):
677         return SCRIPT
678
679     -----
680     def getType( _self ):
681         return SCRIPT
682
683     -----
684     def getType( _self ):
685         return SCRIPT
686
687     -----
688     def getType( _self ):
689         return SCRIPT
690
691     -----
692     def getType( _self ):
693         return SCRIPT
694
695     -----
696     def getType( _self ):
697         return SCRIPT
698
699     -----
700     def getType( _self ):
701         return SCRIPT
702
703     -----
704     def getType( _self ):
705         return SCRIPT
706
707     -----
708     def getType( _self ):
709         return SCRIPT
710
711     -----
712     def getType( _self ):
713         return SCRIPT
714
715     -----
716     def getType( _self ):
717         return SCRIPT
718
719     -----
720     def getType( _self ):
721         return SCRIPT
722
723     -----
724     def getType( _self ):
725         return SCRIPT
726
727     -----
728     def getType( _self ):
729         return SCRIPT
730
731     -----
732     def getType( _self ):
733         return SCRIPT
734
735     -----
736     def getType( _self ):
737         return SCRIPT
738
739     -----
740     def getType( _self ):
741         return SCRIPT
742
743     -----
744     def getType( _self ):
745         return SCRIPT
746
747     -----
748     def getType( _self ):
749         return SCRIPT
750
751     -----
752     def getType( _self ):
753         return SCRIPT
754
755     -----
756     def getType( _self ):
757         return SCRIPT
758
759     -----
760     def getType( _self ):
761         return SCRIPT
762
763     -----
764     def getType( _self ):
765         return SCRIPT
766
767     -----
768     def getType( _self ):
769         return SCRIPT
770
771     -----
772     def getType( _self ):
773         return SCRIPT
774
775     -----
776     def getType( _self ):
777         return SCRIPT
778
779     -----
780     def getType( _self ):
781         return SCRIPT
782
783     -----
784     def getType( _self ):
785         return SCRIPT
786
787     -----
788     def getType( _self ):
789         return SCRIPT
790
791     -----
792     def getType( _self ):
793         return SCRIPT
794
795     -----
796     def getType( _self ):
797         return SCRIPT
798
799     -----
800     def getType( _self ):
801         return SCRIPT
802
803     -----
804     def getType( _self ):
805         return SCRIPT
806
807     -----
808     def getType( _self ):
809         return SCRIPT
810
811     -----
812     def getType( _self ):
813         return SCRIPT
814
815     -----
816     def getType( _self ):
817         return SCRIPT
818
819     -----
820     def getType( _self ):
821         return SCRIPT
822
823     -----
824     def getType( _self ):
825         return SCRIPT
826
827     -----
828     def getType( _self ):
829         return SCRIPT
830
831     -----
832     def getType( _self ):
833         return SCRIPT
834
835     -----
836     def getType( _self ):
837         return SCRIPT
838
839     -----
840     def getType( _self ):
841         return SCRIPT
842
843     -----
844     def getType( _self ):
845         return SCRIPT
846
847     -----
848     def getType( _self ):
849         return SCRIPT
850
851     -----
852     def getType( _self ):
853         return SCRIPT
854
855     -----
856     def getType( _self ):
857         return SCRIPT
858
859     -----
860     def getType( _self ):
861         return SCRIPT
862
863     -----
864     def getType( _self ):
865         return SCRIPT
866
867     -----
868     def getType( _self ):
869         return SCRIPT
870
871     -----
872     def getType( _self ):
873         return SCRIPT
874
875     -----
876     def getType( _self ):
877         return SCRIPT
878
879     -----
880     def getType( _self ):
881         return SCRIPT
882
883     -----
884     def getType( _self ):
885         return SCRIPT
886
887     -----
888     def getType( _self ):
889         return SCRIPT
890
891     -----
892     def getType( _self ):
893         return SCRIPT
894
895     -----
896     def getType( _self ):
897         return SCRIPT
898
899     -----
900     def getType( _self ):
901         return SCRIPT
902
903     -----
904     def getType( _self ):
905         return SCRIPT
906
907     -----
908     def getType( _self ):
909         return SCRIPT
910
911     -----
912     def getType( _self ):
913         return SCRIPT
914
915     -----
916     def getType( _self ):
917         return SCRIPT
918
919     -----
920     def getType( _self ):
921         return SCRIPT
922
923     -----
924     def getType( _self ):
925         return SCRIPT
926
927     -----
928     def getType( _self ):
929         return SCRIPT
930
931     -----
932     def getType( _self ):
933         return SCRIPT
934
935     -----
936     def getType( _self ):
937         return SCRIPT
938
939     -----
940     def getType( _self ):
941         return SCRIPT
942
943     -----
944     def getType( _self ):
945         return SCRIPT
946
947     -----
948     def getType( _self ):
949         return SCRIPT
950
951     -----
952     def getType( _self ):
953         return SCRIPT
954
955     -----
956     def getType( _self ):
957         return SCRIPT
958
959     -----
960     def getType( _self ):
961         return SCRIPT
962
963     -----
964     def getType( _self ):
965         return SCRIPT
966
967     -----
968     def getType( _self ):
969         return SCRIPT
970
971     -----
972     def getType( _self ):
973         return SCRIPT
974
975     -----
976     def getType( _self ):
977         return SCRIPT
978
979     -----
980     def getType( _self ):
981         return SCRIPT
982
983     -----
984     def getType( _self ):
985         return SCRIPT
986
987     -----
988     def getType( _self ):
989         return SCRIPT
990
991     -----
992     def getType( _self ):
993         return SCRIPT
994
995     -----
996     def getType( _self ):
997         return SCRIPT
998
999     -----
1000    def getType( _self ):
1001        return SCRIPT
1002
1003    -----
1004    def getType( _self ):
1005        return SCRIPT
1006
1007    -----
1008    def getType( _self ):
1009        return SCRIPT
1010
1011    -----
1012    def getType( _self ):
1013        return SCRIPT
1014
1015    -----
1016    def getType( _self ):
1017        return SCRIPT
1018
1019    -----
1020    def getType( _self ):
1021        return SCRIPT
1022
1023    -----
1024    def getType( _self ):
1025        return SCRIPT
1026
1027    -----
1028    def getType( _self ):
1029        return SCRIPT
1030
1031    -----
1032    def getType( _self ):
1033        return SCRIPT
1034
1035    -----
1036    def getType( _self ):
1037        return SCRIPT
1038
1039    -----
1040    def getType( _self ):
1041        return SCRIPT
1042
1043    -----
1044    def getType( _self ):
1045        return SCRIPT
1046
1047    -----
1048    def getType( _self ):
1049        return SCRIPT
1050
1051    -----
1052    def getType( _self ):
1053        return SCRIPT
1054
1055    -----
1056    def getType( _self ):
1057        return SCRIPT
1058
1059    -----
1060    def getType( _self ):
1061        return SCRIPT
1062
1063    -----
1064    def getType( _self ):
1065        return SCRIPT
1066
1067    -----
1068    def getType( _self ):
1069        return SCRIPT
1070
1071    -----
1072    def getType( _self ):
1073        return SCRIPT
1074
1075    -----
1076    def getType( _self ):
1077        return SCRIPT
1078
1079    -----
1080    def getType( _self ):
1081        return SCRIPT
1082
1083    -----
1084    def getType( _self ):
1085        return SCRIPT
1086
1087    -----
1088    def getType( _self ):
1089        return SCRIPT
1090
1091    -----
1092    def getType( _self ):
1093        return SCRIPT
1094
1095    -----
1096    def getType( _self ):
1097        return SCRIPT
1098
1099    -----
1100    def getType( _self ):
1101        return SCRIPT
1102
1103    -----
1104    def getType( _self ):
1105        return SCRIPT
1106
1107    -----
1108    def getType( _self ):
1109        return SCRIPT
1110
1111    -----
1112    def getType( _self ):
1113        return SCRIPT
1114
1115    -----
1116    def getType( _self ):
1117        return SCRIPT
1118
1119    -----
1120    def getType( _self ):
1121        return SCRIPT
1122
1123    -----
1124    def getType( _self ):
1125        return SCRIPT
1126
1127    -----
1128    def getType( _self ):
1129        return SCRIPT
1130
1131    -----
1132    def getType( _self ):
1133        return SCRIPT
1134
1135    -----
1136    def getType( _self ):
1137        return SCRIPT
1138
1139    -----
1140    def getType( _self ):
1141        return SCRIPT
1142
1143    -----
1144    def getType( _self ):
1145        return SCRIPT
1146
1147    -----
1148    def getType( _self ):
1149        return SCRIPT
1150
1151    -----
1152    def getType( _self ):
1153        return SCRIPT
1154
1155    -----
1156    def getType( _self ):
1157        return SCRIPT
1158
1159    -----
1160    def getType( _self ):
1161        return SCRIPT
1162
1163    -----
1164    def getType( _self ):
1165        return SCRIPT
1166
1167    -----
1168    def getType( _self ):
1169        return SCRIPT
1170
1171    -----
1172    def getType( _self ):
1173        return SCRIPT
1174
1175    -----
1176    def getType( _self ):
1177        return SCRIPT
1178
1179    -----
1180    def getType( _self ):
1181        return SCRIPT
1182
1183    -----
1184    def getType( _self ):
1185        return SCRIPT
1186
1187    -----
1188    def getType( _self ):
1189        return SCRIPT
1190
1191    -----
1192    def getType( _self ):
1193        return SCRIPT
1194
1195    -----
1196    def getType( _self ):
1197        return SCRIPT
1198
1199    -----
1200    def getType( _self )
```



PyCharm: Versioning

C:\Users\LibreOfficiant\PycharmProjects\apso\apso

Side-by-side viewer | Do not ignore | Highlight words | 145 differences

Last 12 Hours	19 minutes ago - apso (Read-only)	Current
17 minutes ago apso 0.8.7	<pre>import uno import unohelper import traceback import sys try: import pythonscript except: import pythonloader pythonscript = None for url, module in pythonloader.g_loadedComponents.iteritems(): if url.endswith("script-provider-for-python/pythonscript.py"): pythonscript = module if pythonscript is None: raise Exception("Failed to find pythonscript module.")</pre>	<pre>from sys import getfilesystemencoding, version_info from threading import Thread from subprocess import call as sub_call try: from urllib.parse import unquote except ImportError: from urllib import unquote try: import pythonscript except: import pythonloader pythonscript = None for url, module in pythonloader.g_loadedComponents.iteritems(): if url.endswith("script-provider-for-python/pythonscript.py"): pythonscript = module if pythonscript is None: raise Exception("Impossible de trouver le module pythonscript.")</pre>
17 minutes ago apso 0.8.3	<pre>from com.sun.star.awt import XActionListener, XMouseListener, XKeyListener, Rectangle, Selection from com.sun.star.awt.tree import XTreeExpansionListener from com.sun.star.uno import Exception as UNOException</pre>	<pre>from com.sun.star.task import XJobExecutor from com.sun.star.awt import (XActionListener, XMouseListener, XKeyListener, XContainerWindowEventHandler, Rectangle, Selection) from com.sun.star.awt.tree import XTreeExpansionListener from com.sun.star.view import XSelectionChangeListener from com.sun.star.beans import PropertyValue from com.sun.star.uno import Exception as UNOException from com.sun.star.awt.MessageBoxType import MESSAGEBOX, INFOBOX, ERRORBOX, from com.sun.star.awt.MessageBoxResults import YES from com.sun.star.ui.dialogs.ExecutableDialogResults import OK from com.sun.star.ui.dialogs.TemplateDescription import FILESAVE_AUTOEXTENS</pre>
18 minutes ago apso	<pre># Default content of the new file. TEMPLATE = """</pre>	<pre>#----- # GENERICS #----- # Addon ID EXTID = 'apso.python.script.organizer'</pre>
18 minutes ago apso	<pre>class DialogBase(object): """ Base class for dialog. """ def __init__(self, ctx): self.ctx = ctx def create(self, name): """ Create service instance. """ return self.ctx.getServiceManager().createInstanceWithContext(name, self.ctx)</pre>	<pre># lib or 300.2</pre>
19 minutes ago apso 0.0.0 © Hanya	<pre>class RuntimeDialogBase(DialogBase):</pre>	



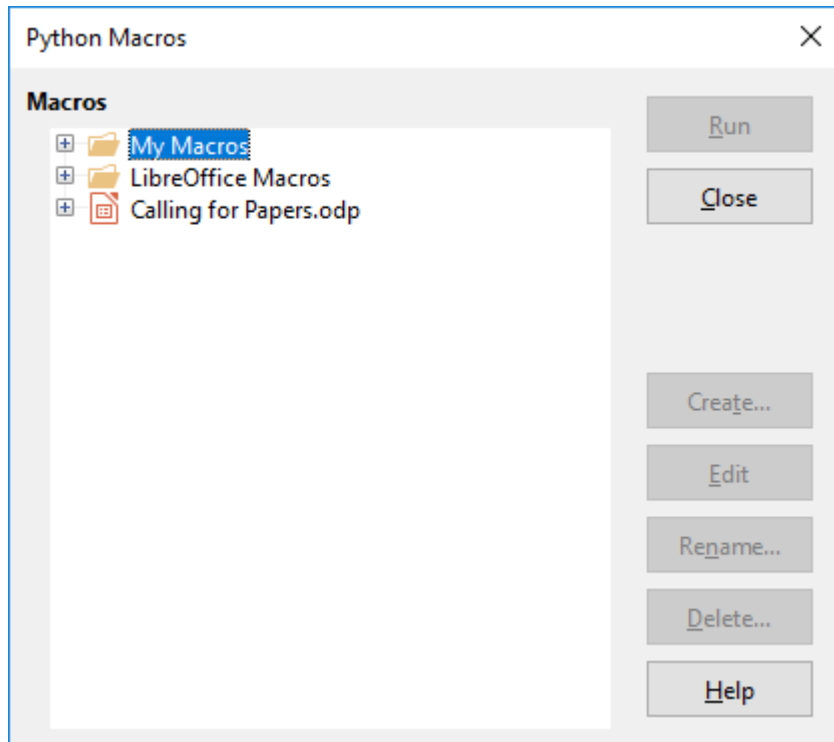
Agenda

- Scripting Framework
- Why Python ?
 - Srce Explorer, Syntax Hiliting, Code Completion, Debugging,
 - Coding Guidelines, Test-Driven Dev, Version Control
- Walkthrough Geany, PyZo, PyCharm
- Extension: APSO - JMZ
 - Xray by BM, Mri by Hanya
- IDEs
 - Projects Setup
- Scripting Python Macros
 - Help, Libraries & Modules
 - Programming Basics
 - Dialogs, Listeners, Monitors
 - X-Scripting
- Q & A



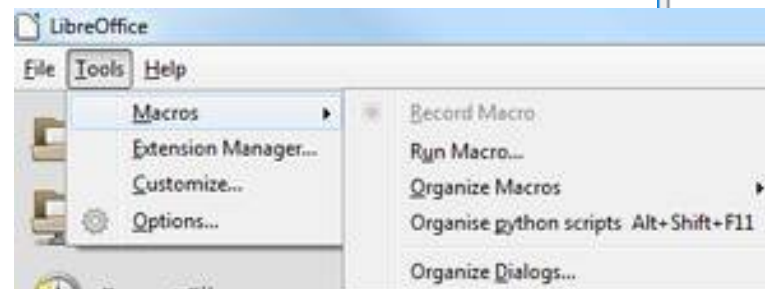
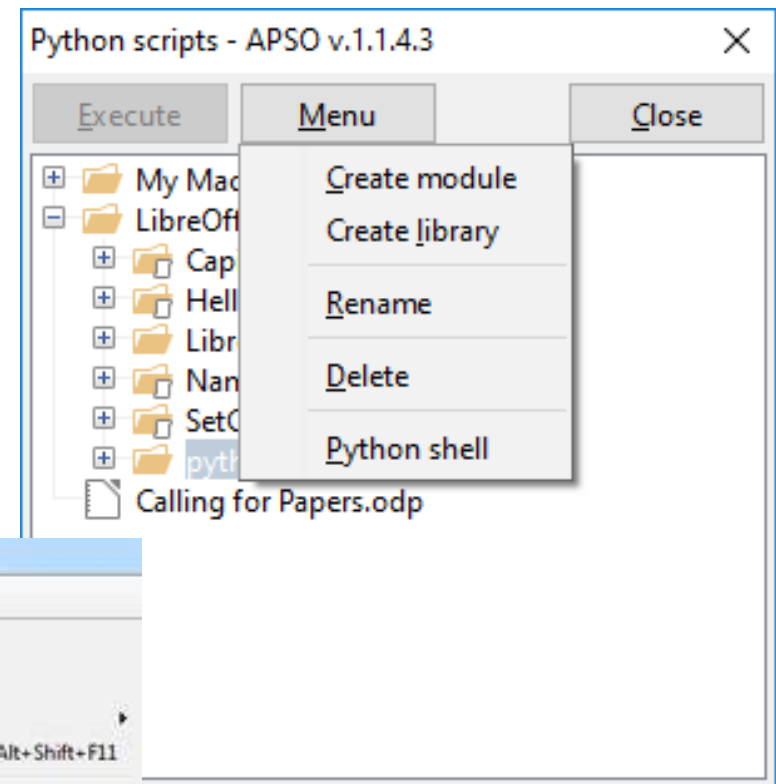


Altern. Python Script Org^{zer}



More although..

..Less





APSO: interactive console

```
APSO console
APSO python console [LibreOffice]
3.5.4 (default, May  9 2018, 21:39:23) [MSC v.1900 64 bit (AMD64)]
Type "help", "copyright", "credits" or "license" for more information.
>>> ctx = XSCRIPTCONTEXT.getComponentContext()
>>> smgr = ctx.ServiceManager
>>> geb = smgr.createInstanceWithContext("com.sun.star.frame.GlobalEventBroadcaster", ctx)
>>> dir(geb.Events)
['ElementNames', 'ElementType', 'Events', 'ImplementationId', 'OnCloseApp', 'OnCopyTo',
'OnCopyToDone', 'OnCopyToFailed', 'OnCreate', 'OnFocus', 'OnLoad', 'OnLoadFinished',
'OnModeChanged', 'OnModifyChanged', 'OnNew', 'OnPrepareUnload', 'OnPrepareViewClosing', 'OnPrint',
'OnSave', 'OnSaveAs', 'OnSaveAsDone', 'OnSaveAsFailed', 'OnSaveDone', 'OnSaveFailed',
'OnStartApp', 'OnStorageChanged', 'OnTitleChanged', 'OnUnfocus', 'OnUnload', 'OnViewClosed',
'OnViewCreated', 'OnVisAreaChanged', 'Types', 'getByName', 'getElementNames', 'getElementType',
'getEvents', 'getImplementationId', 'getTypes', 'hasByName', 'hasElements', 'queryAdapter',
'queryInterface', 'replaceByName']
>>> help(geb.Events.getElementNames)
Help on PyUNO_callable object:

class PyUNO_callable(object)
|   Methods defined here:
|
|   __call__(self, /, *args, **kwargs)
|       Call self as a function.
>>> |
```

Read Evaluate Print Loop



APSO: debug window

The screenshot shows the APSO_DEBUG window with the following components:

- Toolbar:** NEXT, EXIT LOOP, STEP IN (highlighted), RETURN, RESTART, EDIT FILE, CONSOLE, QUIT.
- Code Editor:** Displays Python code from `c:\program files\libreoffice\program\python-core-3.5.7\lib\subprocess.py`. The current line is 563: `def __init__(self, args, bufsize=-1, executable=None,`. The code continues with parameters for `stdin`, `stdout`, `stderr`, `preexec_fn`, `close_fds`, `shell`, `cwd`, `env`, `universal_newlines`, `startupinfo`, `creationflags`, `restore_signals`, `start_new_session`, and `pass_fds`. It includes a docstring: `"""Create new Popen instance."""` and a `_cleanup()` method. Comments describe the `waitpid` lock and the `self._input` attribute. The code ends with `self._communication_started = False` and a conditional `bufsize` assignment.
- Stack (double click a frame to display its state):** Shows `interpreter_console()` and `__init__(self=<subprocess.Popen object at 0x00000288016AAF98>, args='C:\\\\`.
- Local scope (double click to inspect a variable with MRI or Xray):** Shows the following variables:
 - `any_stdio_set :`
 - `args : 'C:\\\\Program Files\\LibreOffice\\program`
 - `bufsize : -1`
 - `c2pread :`
 - `c2pwrite :`
 - `close_fds : <object object at 0x000002887ADC00F`
 - `creationflags : 0`
 - `cwd : None`
 - `env : None`
 - `errread :`

see also Mri, Xray



Agenda

- Scripting Framework
- Why Python ?
 - Srce Explorer, Syntax Hiliting, Code Completion, Debugging,
 - Coding Guidelines, Test-Driven Dev, Version Control
- Walkthrough Geany, PyZo, PyCharm
- Extension: APSO - JMZ
 - Xray by BM, Mri by Hanya
- IDEs
 - Projects Setup
- Scripting Python Macros
 - Help, Libraries & Modules
 - Programming Basics
 - Dialogs, Listeners, Monitors
 - X-Scripting
- Q & A





IDEs: Setup

- Geany
 - Pyzo
- PyCharm



IDEs: Geany projects_ setup

Project Properties

Project Indentation Editor Files **Build**

#	Label	Command	Working directory	Reset
Python commands				
1.	Compile	s/python -m py_compile "%f"		✕
2.				✕
3.	Lint	pep8 --max-line-length=80 "%f"		✕
Error regular expression:		(.+) : ([0-9]+) : ([0-9]+)		✕
Independent commands				
1.	Make	make		✕
2.	Make Custom Target...	make		✕
3.	Make Object	make %e.o		✕
4.				✕
Error regular expression:				✕
<i>Note: Item 2 opens a dialog and appends the response to the command.</i>				
Execute commands				
1.	Execute	tents/Resources/python "%f"		✕
2.				✕

%d, %e, %f, %p, %l are substituted in command and directory fields, see manual for details.

Cancel OK

Project Properties

Project Indentation Editor Files **Build**

#	Label	Command	Working directory	Reset
Python commands				
1.	Compile	"D:\Program Files\LibreOffice"		📁
2.				📁
3.	Lint	pep8 --max-line-length=80 "%f"		📁
Error regular expression:		((^:)+):([0-9]+):([0-9:]+)?.*		📁
Independent commands				
1.	Make	make		📁
2.	Make Custom Target...	make		📁
3.	Make Object	make %e.o		📁
4.				📁
Error regular expression:		((^:)+):([0-9]+):([0-9:]+)?.*		📁
<i>Note: Item 2 opens a dialog and appends the response to the command.</i>				
Execute commands				
1.	Execute	"D:\Program Files\LibreOffice"		📁
2.				📁

%d, %e, %f, %p, %l are substituted in command and directory fields, see manual for details.

Cancel OK



IDEs: PyZo shell configurations

Shell configurations

Python Add config

name

exe

ipython Use IPython shell if available.

gui

pythonPath

Use system default

startupScript

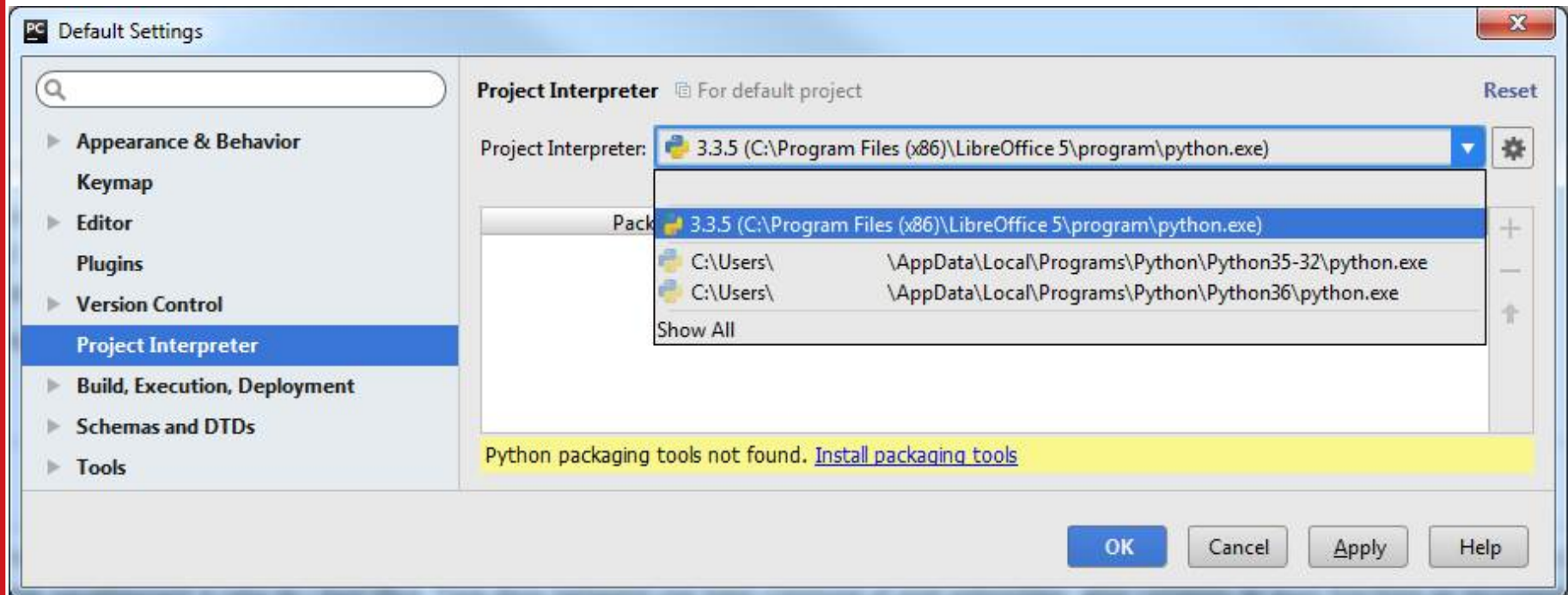
Use system default
 File to run at startup
 Code to run at startup

startDir

Cancel Done



IDEs: PyCharm project interpreters



- Debugging Python & Basic altogether



Agenda

- Scripting Framework
- Why Python ?
 - Srce Explorer, Syntax Hiliting, Code Completion, Debugging,
 - Coding Guidelines, Test-Driven Dev, Version Control
- Walkthrough Geany, PyZo, PyCharm
- Extension: APSO - JMZ
 - Xray by BM, Mri by Hanya
- IDEs
 - Projects Setup
- Scripting Python Macros
 - Help, Libraries & Modules
 - Programming Basics
 - Dialogs, Listeners, Monitors
 - X-Scripting
- Q & A





Scripting: API help

- LibreOffice Help
local / [online](#)
- [API documentation](#)
[Software Developer Kit](#)
- Python Interactive Console (REPL)
 - [Forums](#)



Scripting: module/library

- Programming Python Scripts
- Standalone module: XSCRIPTCONTEXT
- Shared module: import uno
- Xlating Basic 2 Python



Scripting: Basics

- Input/Output to Screen
- Getting session information
- Identifying the operating system
 - Importing Python modules



Scripting: Dialogs, Listeners

- Std dialogs, Listeners
 - Dialog handlers
in 6.4
- Monitoring doc. Events
demo



Scripting: X-Languages

- Python to Basic
MsgBox, InputBox
other use case?
- Basic to Python
Std modules
cmath, datetime, json, math, net, re,
unittest
e.g. python.os.Filelen - demo
- Python/Basic to JavaScript
..you name it..



e-References

- Altern. Python Script Org. (APSO)
(Mri, Xray)

- local/online **Help**
Macros and Scripting

- **Wikis**

Designing & Developing Python Apps
Python Apache OpenOffice

Thanks

Q & A

come to discuss

- Python IDE simple enablement (for *Office) using IDE_utils
- LibreOffice Basic Hidden Gems

