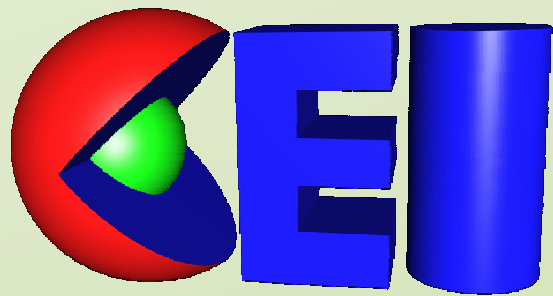


Scripting EnSight with Python



2010 Japan EnSight Forum

Presented by
Aric Meyer



What is Python?

- A programming language
 - Free
 - Open source
 - Cross platform
 - High-level
 - Optimized to save your time, not CPU time



Why does Python matter for EnSight?

- Python 2.5.1 is installed with every copy of EnSight
- Python is used by CEI to make parts of the EnSight GUI and perform other tasks
- EnSight can be scripted using Python



What can be done with Python scripting?

- Almost anything that can be done in EnSight GUI
- And many things that can't



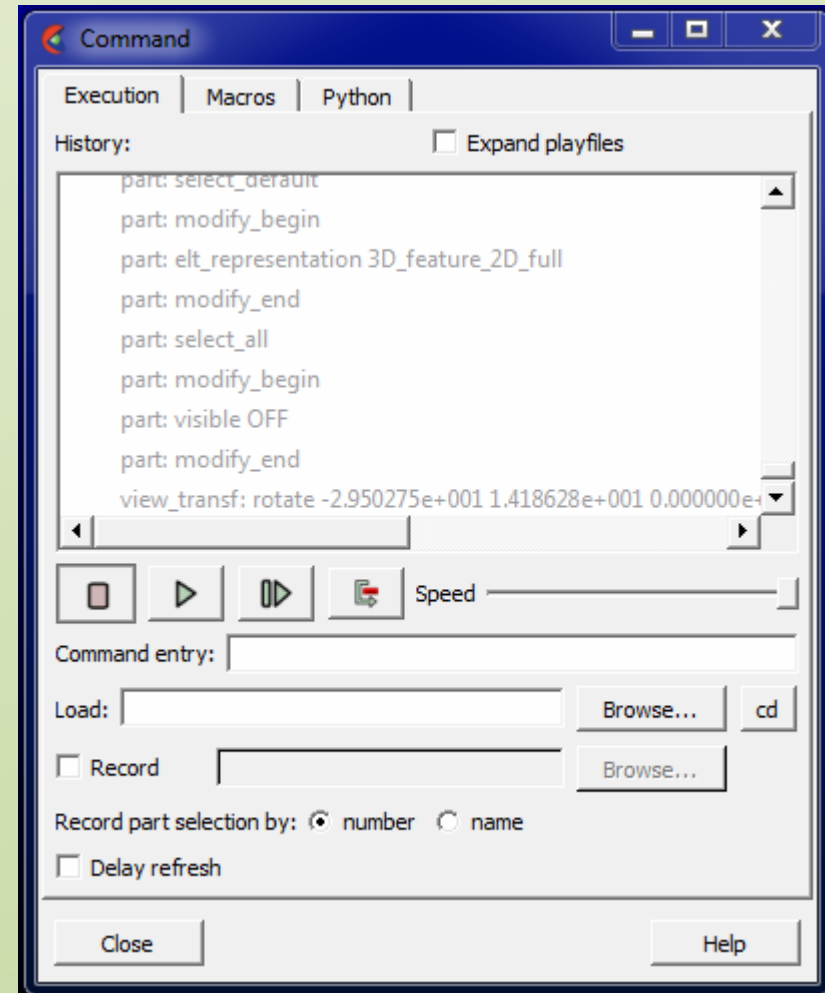
What do I need to know to write Python scripts?

- Some basic Python
 - You can ‘learn as you go’
- How to use EnSight’s resources
 - 90% of what you do can be copy-paste



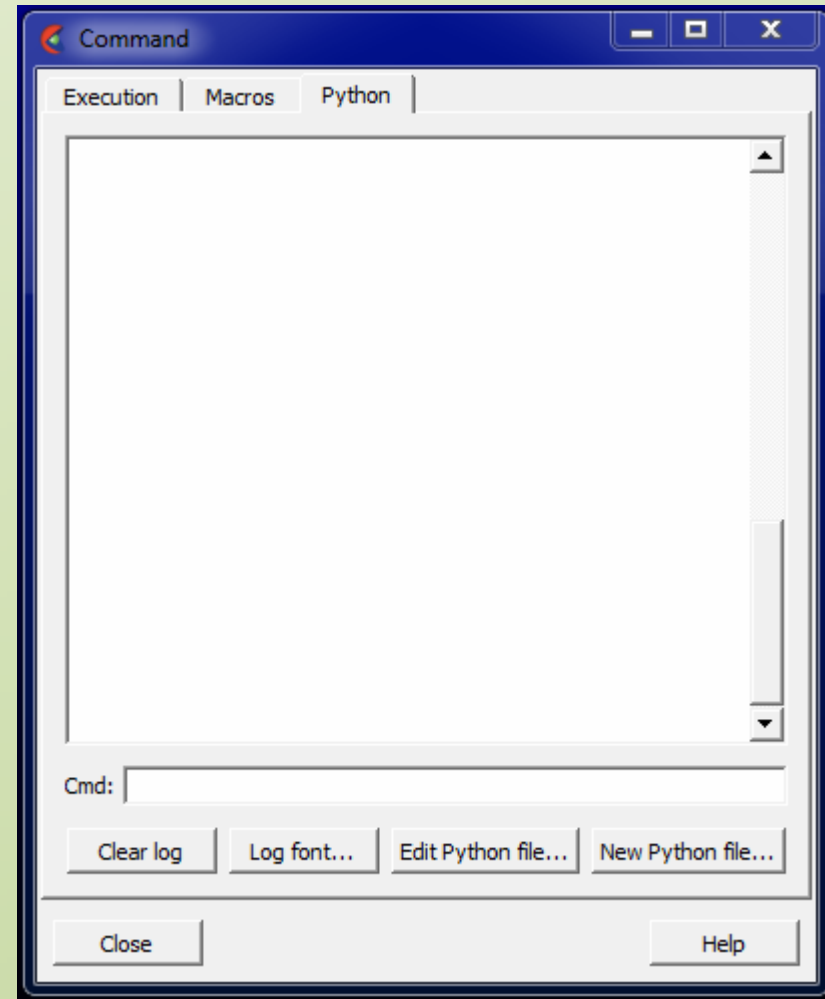
EnSight command language

- Everything you do in EnSight generates command language
- See the output in the command window
 - File->Command...



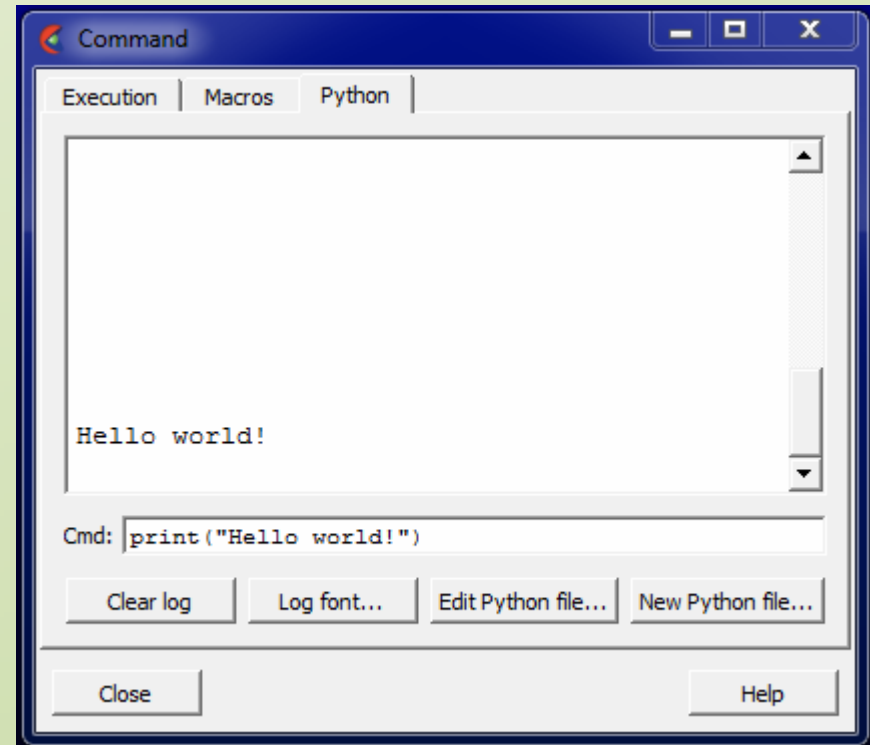
EnSight Python interpreter

- Python tab of the command window

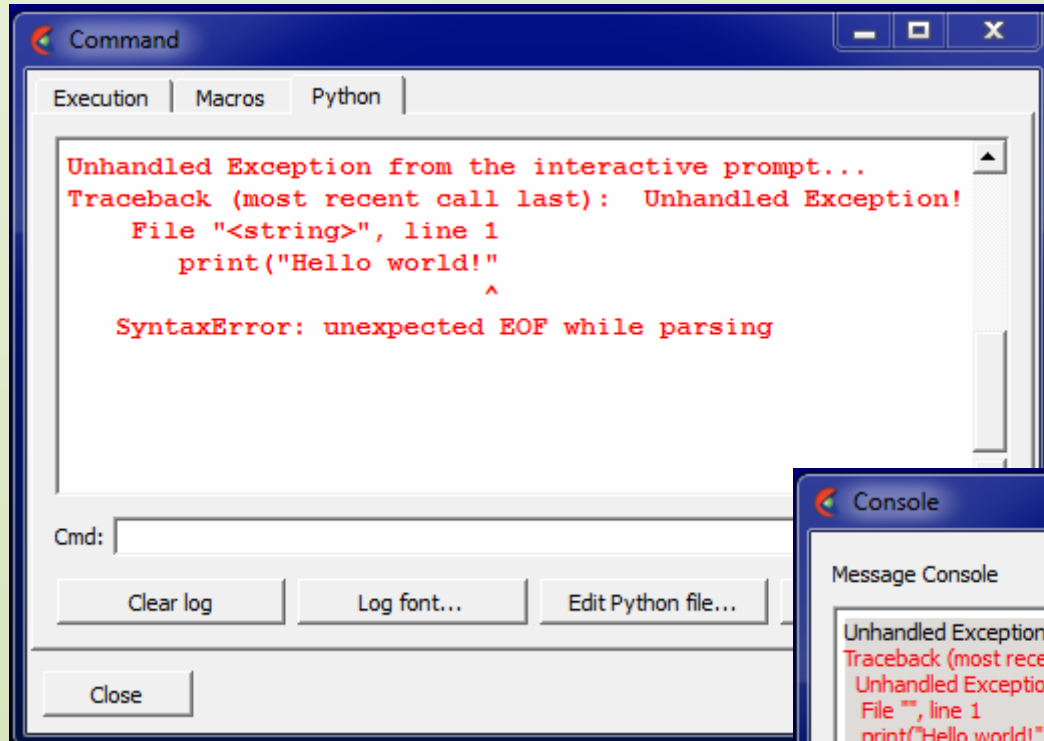


EnSight Python interpreter

- Type commands into the 'Cmd:' field and see immediate results
- Great for trying out new lines of code



EnSight Python interpreter



Command

Execution | Macros | Python

```
Unhandled Exception from the interactive prompt...
Traceback (most recent call last):  Unhandled Exception!
  File "<string>", line 1
    print("Hello world!")
    ^
SyntaxError: unexpected EOF while parsing
```

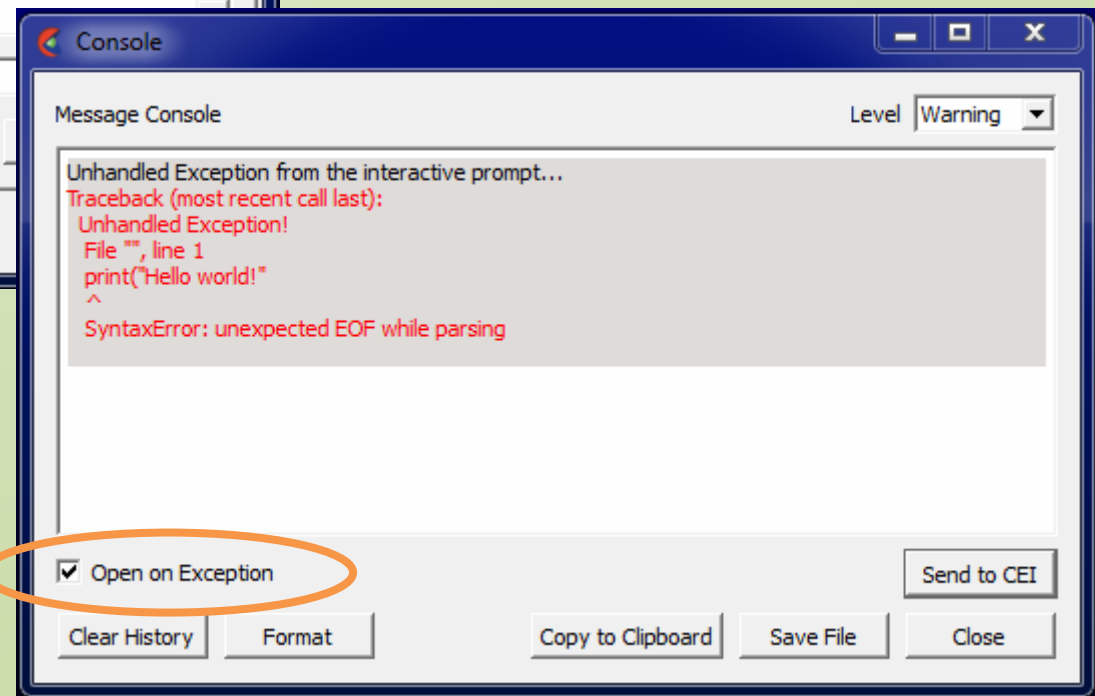
Cmd:

Clear log | Log font... | Edit Python file...

Close

Useful error messages make debugging easy

The console window will pop up when an Exception (error) occurs. I like to uncheck the 'Open on Exception' box and read the output in the command window.



Console

Message Console Level: Warning

```
Unhandled Exception from the interactive prompt...
Traceback (most recent call last):
  Unhandled Exception!
  File "", line 1
    print("Hello world!")
    ^
SyntaxError: unexpected EOF while parsing
```

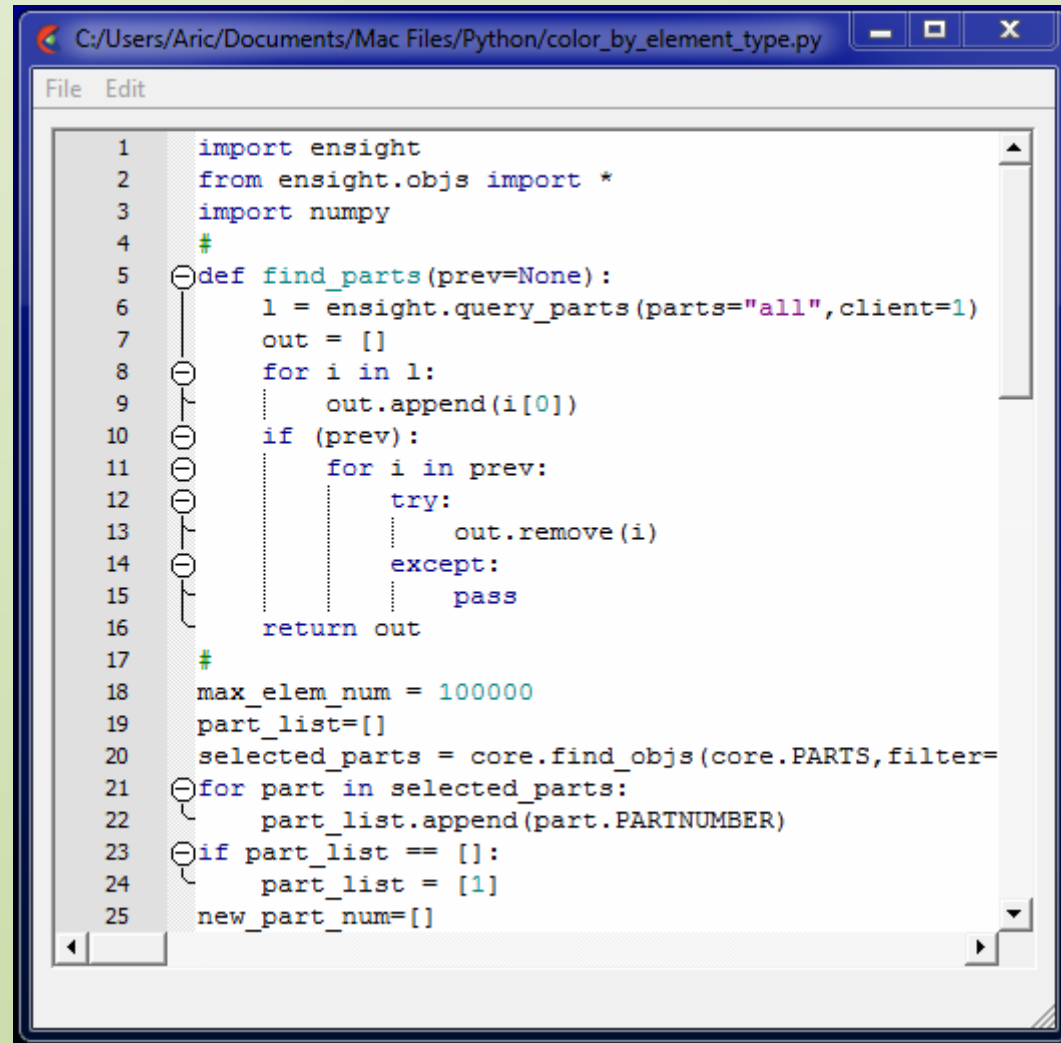
Open on Exception

Send to CEI

Clear History | Format | Copy to Clipboard | Save File | Close

Python editor

- EnSight contains a full-featured Python editor
- Click on 'Edit Python file...' or 'New Python file...' to open
- Edit scripts in here or copy-paste from your favorite editor



```
C:/Users/Aric/Documents/Mac Files/Python/color_by_element_type.py
File Edit
1 import ensight
2 from ensight.objs import *
3 import numpy
4 #
5 def find_parts(prev=None):
6     l = ensight.query_parts(parts="all", client=1)
7     out = []
8     for i in l:
9         out.append(i[0])
10        if (prev):
11            for i in prev:
12                try:
13                    out.remove(i)
14                except:
15                    pass
16        return out
17 #
18 max_elem_num = 100000
19 part_list=[]
20 selected_parts = core.find_objs(core.PARTS, filter=
21 for part in selected_parts:
22     part_list.append(part.PARTNUMBER)
23 if part_list == []:
24     part_list = [1]
25     new_part_num=[]
```



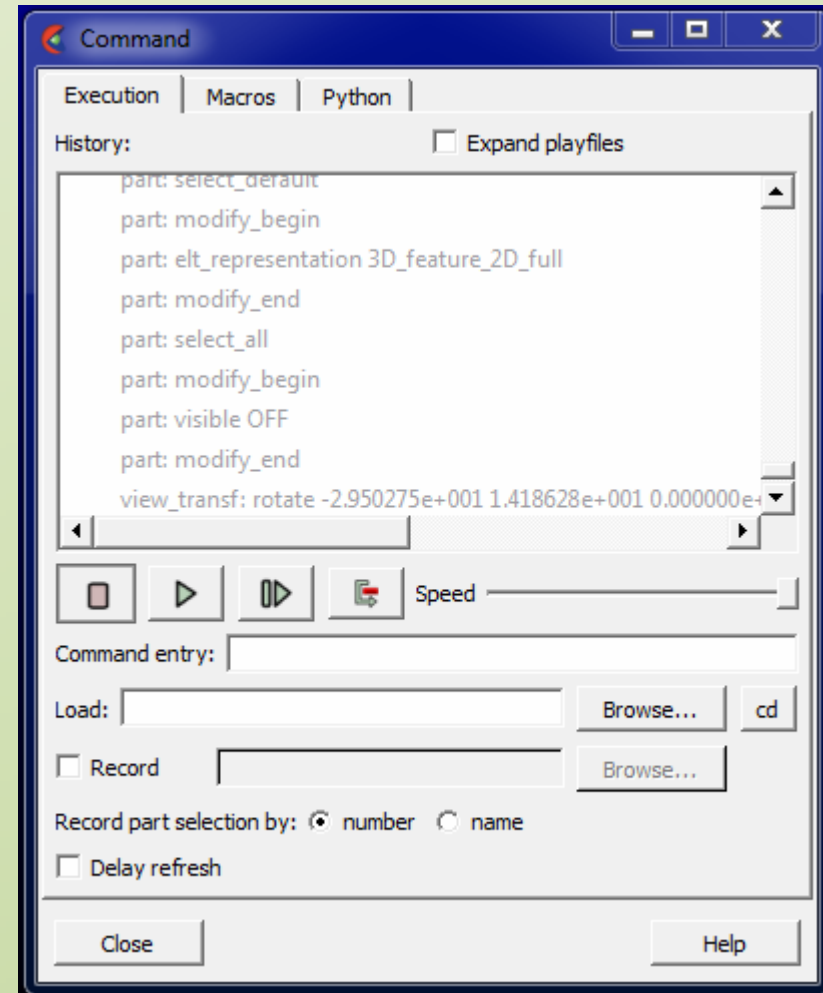
Scripting made easy

Step 1

Execute an action in the EnSight GUI

Step 2

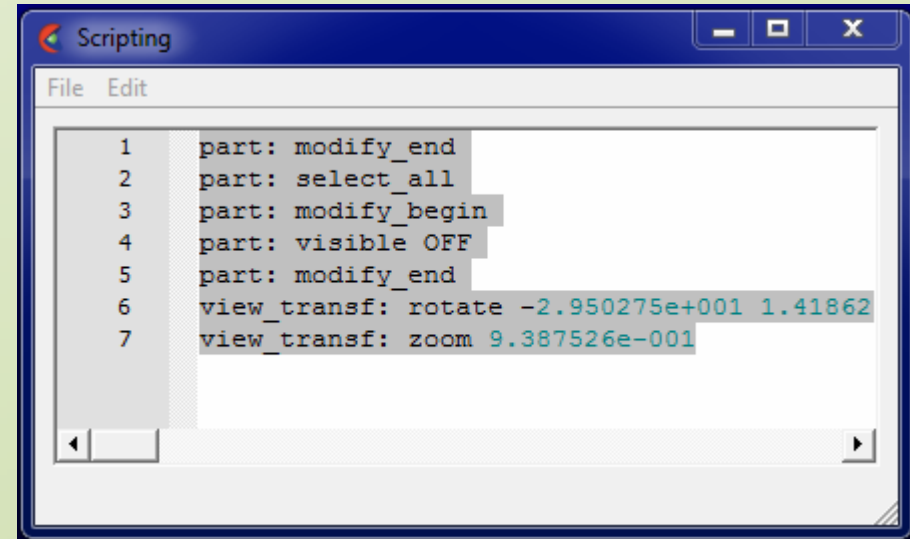
Copy desired commands from the command window



Scripting made easy

Step 3

Paste into the Python editor



```
Scripting
File Edit
1 part: modify_end
2 part: select_all
3 part: modify_begin
4 part: visible OFF
5 part: modify_end
6 view_transf: rotate -2.950275e+001 1.41862
7 view_transf: zoom 9.387526e-001
```

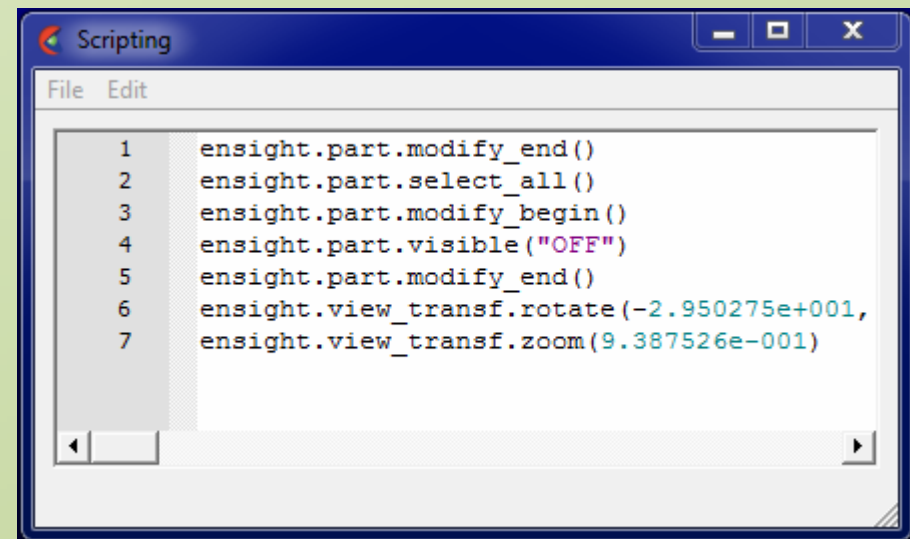
Step 4

Convert EnSight command language to Python automatically

Edit->Convert selection to native Python

Add variables, loops, queries, functions...

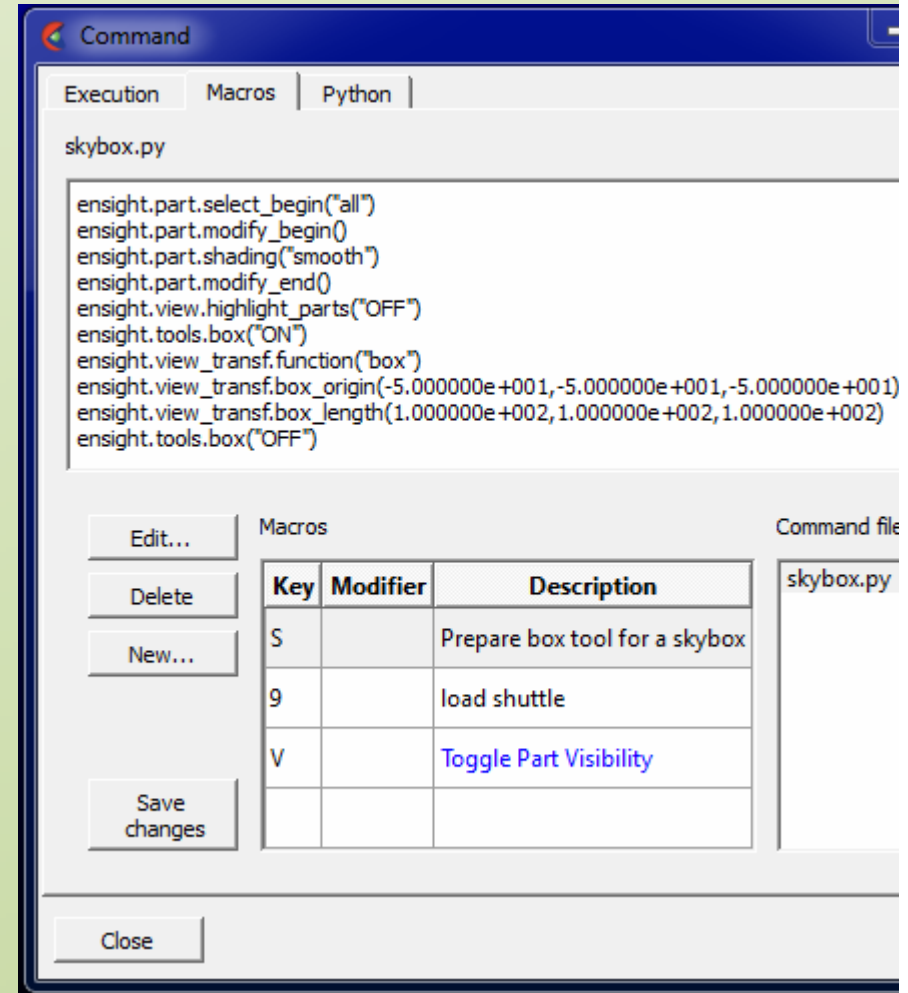
Use the full power of Python



```
Scripting
File Edit
1 ensight.part.modify_end()
2 ensight.part.select_all()
3 ensight.part.modify_begin()
4 ensight.part.visible("OFF")
5 ensight.part.modify_end()
6 ensight.view_transf.rotate(-2.950275e+001,
7 ensight.view_transf.zoom(9.387526e-001)
```

Using your scripts

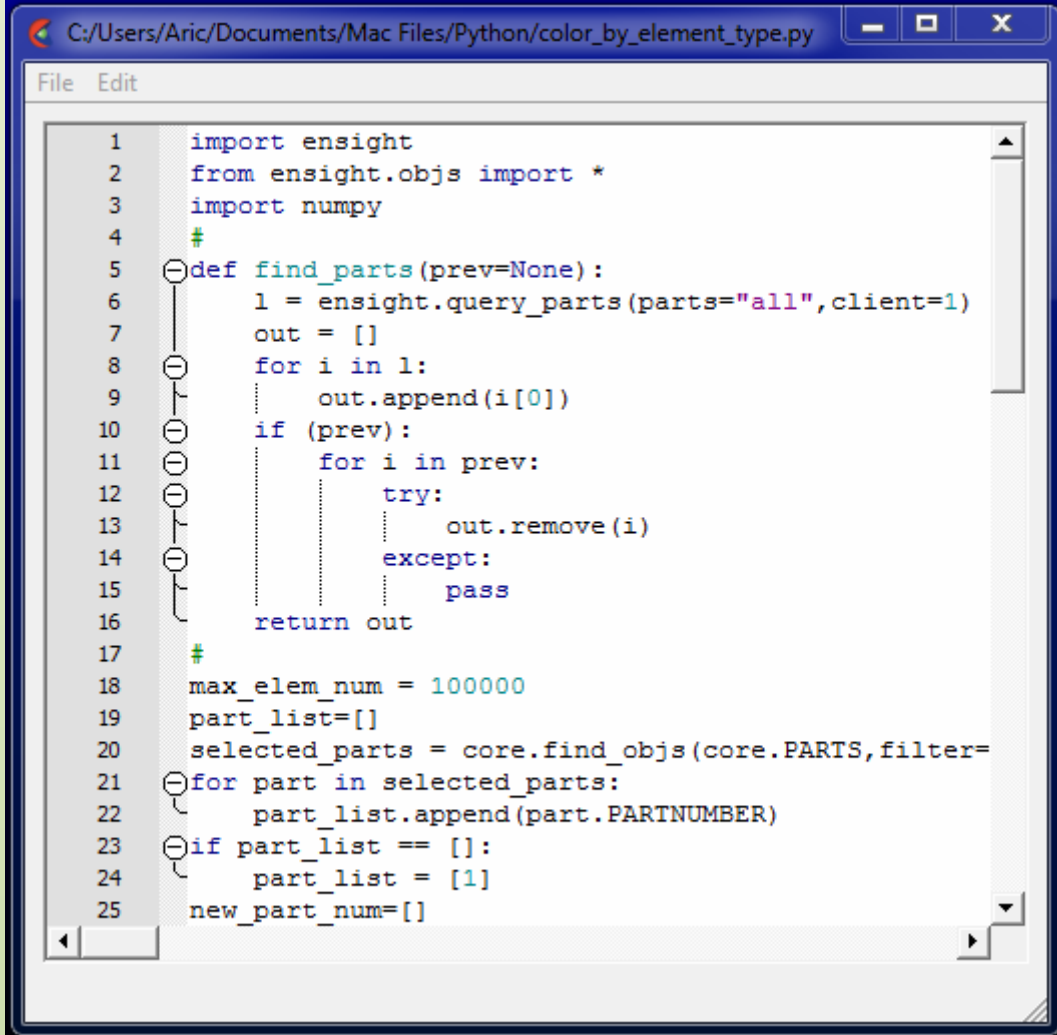
- Ways to execute scripts
 - In the Python editor
 - As a macro
 - As a user defined tool
 - In a right-click menu
 - In batch mode
 - In a keyframe animation
 - From another script



Using your scripts

In the Python editor

File->Run script (execfile)



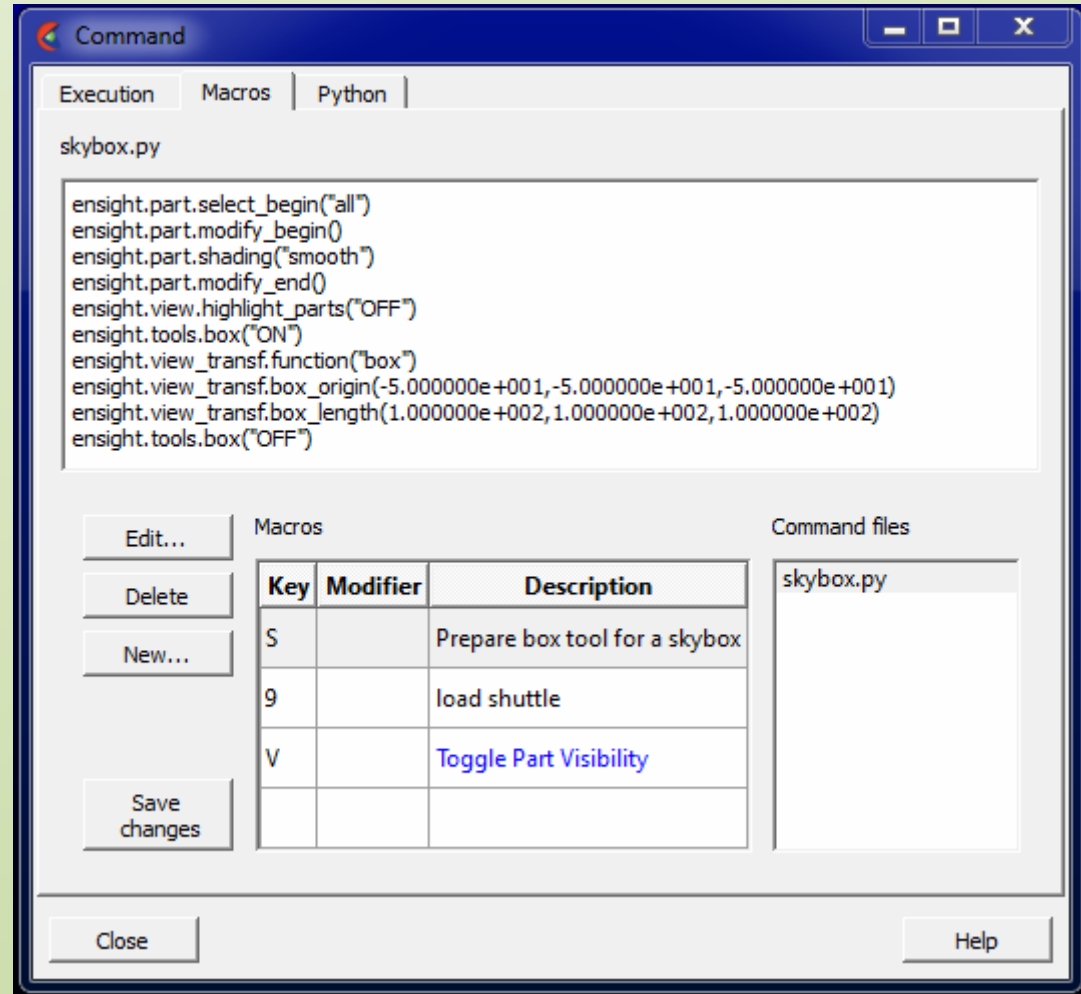
```
C:/Users/Aric/Documents/Mac Files/Python/color_by_element_type.py
File Edit
1 import ensight
2 from ensight.objs import *
3 import numpy
4 #
5 def find_parts(prev=None):
6     l = ensight.query_parts(parts="all", client=1)
7     out = []
8     for i in l:
9         out.append(i[0])
10        if (prev):
11            for i in prev:
12                try:
13                    out.remove(i)
14                except:
15                    pass
16        return out
17 #
18 max_elem_num = 100000
19 part_list=[]
20 selected_parts = core.find_objs(core.PARTS, filter=
21 for part in selected_parts:
22     part_list.append(part.PARTNUMBER)
23 if part_list == []:
24     part_list = [1]
25 new_part_num=[]
```



Using your scripts

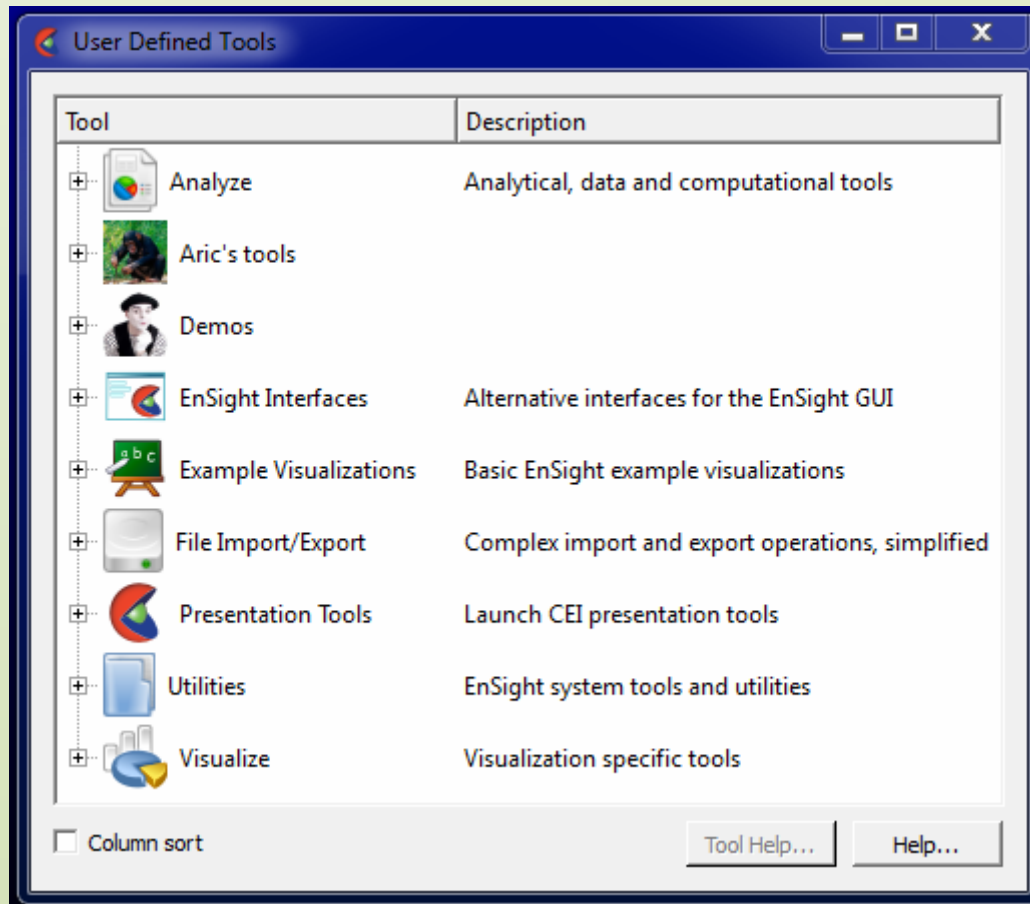
As a macro

- Command window, Macros tab
- Assign a script to a keyboard shortcut



Using your scripts

Add to the user defined tools menu



How to create a user defined tool

- Step 1: Add a header to your script
- Step 2: Place your script in the correct directory
- Step 3: Launch EnSight by typing:
`ensight91 -no_prefs`

Your tool should now show up in the menu

- User defined tools are located in:
 - CEI_HOME¥ensight91¥ site_preferences¥extensions¥user_defined¥Tools
 - Add them here or in your .ensight91 user directory:
 - Windows Vista and 7: C:¥Users¥username¥.ensight91
 - Windows (XP and older): C:¥Documents and Settings¥username¥.ensight91
 - Linux: ~/.ensight91
 - Mac: ~/Library/Application Support/EnSight91



How to create a user defined tool

Header lines

- Add the following lines to the beginning of your script
- Edit the green items to say whatever you want
- The icon and tooltip are optional

```
#ENSIGHT_USER_DEFINED_BEGIN
```

```
#TEXT=GUI Demo
```

```
#TOOLTIP=Demonstrate the GUI features available to EnSight python
```

```
#ICON=gui_demo.png
```

```
#TYPE=TOOL
```




```
#ENSIGHT_USER_DEFINED_END
```



Create your own UDT directory

My UDT directory is located in:

C:\Users\Aric\.ensight91\extensions\user_defined\Tools\Aric's tools

Name	Date modified	Type	Size
 Aric's tools	9/8/2010 9:50 PM	File folder	
 chimp.png	10/7/2009 4:19 AM	PNG File	171 KB
 tools.define	10/19/2010 5:37 PM	DEFINE File	1 KB

This is the entire tools.define file:

```
#ENSIGHT_USER_DEFINED_BEGIN  
#NAME=Aric's tools  
#TEXT=Aric's tools  
#TOOLTIP=User defined tools made by Aric  
#ICON=chimp.png  
#TYPE=TOOLDIR  
#ENSIGHT_USER_DEFINED_END
```










Other Python resources

- EnSight's How To Manual, How To...
 - Produce Customized Access to Tools & Features
 - Produce Customized Pop-up Menus
- EnSight's Interface Manual
 - Chapter 6
- User defined tool:
 - Utilities -> Development -> Python Object API Attributes
- CEI Python API webpage:
 - <https://sites.google.com/a/ensight.com/ensight-python-api/>
- Your distributor and CEI support
 - Help developing your own scripts is included in CEI support!



Python Examples

Or “What a sales guy has been able to do”

Aric's tools		
	Crash fail test	Identify failed parts vs. time
	Cylinder tool length	Set the length of the cylinder tool
	Display seed points	Shows particle trace seed points without creating particle traces
	Find part centerline	Creates a spline at the center of a part or group or parts
	Fit view every timestep	Executes the 'fit' command whenever the timestep is changed
	Rainbow streamlines	Emit a rake of streamlines each colored by a different solid color
	Rotating tire animation	Makes an animated texture

- Also data translators into EnSight case format using the Python API

