Cheatsheet for PyMAPDL

Launching PyMAPDL

本地启动PyMAPDL实例并退出

```
# To launch an instance
from ansys.mapdl.core import launch_mapdl
mapdl=launch_mapdl()
# To exit the instance
mapdl.exit()
```

指定作业名称、处理器数量和工作目录

```
jname='user_jobname'
path = <path of directory>
mapdl=launch_mapdl(nproc=2,run_location=path,
    jobname=jname)
```

创建和退出实例池

```
# To create a pool of 10 instances
from ansys.mapdl.core import LocalMapdlPool
pool=mapdl.LocalMapdlPool(10)
# To exit the pool
pool.exit()
```

PyMAPDL Language

PyMAPDL 命令是对 APDL 命令进行包装的 Python 语句。例如，*ESEL,S,,TYPE1* 翻译为

```
mapdl.esel('s','type',vmin=1)
```

以 * 或 / 开头的命令会删除这些字符

```
mapdl.prep7()    # /PREP7
mapdl.get()      # *GET
```

若删除 * 或 / 会导致与其他命令冲突，则会添加前缀 "slash "或 "star"

```
mapdl.solu()             # SOLU
mapdl.slashsolu()        # /SOLU

mapdl.vget()             # VGET
mapdl.starvget()         # *VGET
```

将现有 APDL 脚本转换为 PyMAPDL 格式

```
inputfile='ansys_inputfile.inp'
pyscript='pyscript.py'
mapdl.convert_script(inputfile,pyscript)
```

MAPDL Class

将表格从 Python 加载到 MAPDL

```
mapdl.load_table(name,array,var1='', var2='',
    var3='', csysid='')
```

从 MAPDL 数据库访问或向其写入参数

```
# To save a parameter named 'displ_load' to a
    NumPy array nparray
nparray=mapdl.parameters['displ_load']
# To create a parameter named 'exp_disp' from a
    NumPy array nparray
mapdl.parameters['exp_disp']=nparray
```

使用 *GET 和 *VGET 直接访问 NumPy 数组的信息

```
# Runs the *GET command and returns a Python
    value.
mapdl.get_value(entity='', entnum='', item1='',
    it1num='', item2='', it2num='', **kwargs)

# Runs *VGET command and returns a Python array
    .
mapdl.get_array(entity='', entnum='', item1='',
    it1num='', item2='', it2num='', kloop='',
    **kwargs)
```

Mesh Class

将有限元网格存储为 VTK UnstructuredGrid 数据对象。

```
grid = mapdl.mesh.grid
```

将单元和节点编号保存到 Python 数组中。

```
# Array of nodal coordinates
nodes=mapdl.mesh.nodes

# Save node numbers of selected nodes to array
node_num=mapdl.mesh.nnum
# Save node numbers of selected nodes to array
node_num_all=mapdl.mesh.nnum_all

# Element numbers of currently selected
    elements
elem_num=mapdl.mesh.enum
# Array of all element numbers, even those not
    selected.
elem_num_all=mapdl.mesh.enum_all
```

Post-processing Class

绘制结果的一般形式是

```
mapdl.postprocessing.result_name
```

```
mapdl.post1()
mapdl.set(1, 2)
# To plot the nodal equivalent stress
mapdl.post_processing.plot_nodal_eqv_stress()
# To save nodal eqv. stresses to a Python array
nod_eqv_stress=mapdl.post_processing.
    nodal_eqv_stress()
# To plot the contour legend or Scalar bar
    using python data structure dictionary
mapdl.allsel()
sbar_kwargs = {"color": "black", "title": "1st␣
    Principal␣Stress␣(psi)", "vertical": False,
     "n_labels": 6}
mapdl.post_processing.
    plot_nodal_principal_stress('1', cpos='xy',
     background='white', edge_color='black',
    show_edges=True, scalar bar_args=
    sbar_kwargs, n_colors=9)
```

Plotting Class

用于 APDL 几何图形和网格的通用 PyMAPDL 绘图仪是：

```
plotting.general_plotter(meshes,points,labels)
```

```
# To plot the currently selected elements
mapdl.eplot([show_node_numbering, vtk])
# To plot the selected volumes
mapdl.vplot([nv1, nv2, ninc, degen, scale,
    ...])
# To display the selected areas
mapdl.aplot([na1, na2, ninc, degen, scale,
    ...])
# To display the selected lines without MAPDL
    plot symbols
mapdl.lplot(vtk=True, cpos='xy', line_width=10)
# To save a '.png' file of the line plot with
    MAPDL coordinate symbol
mapdl.psymb('CS', 1)
mapdl.lplot(vtk=False)
```

References from PyMAPDL Documentation

- Getting Started
- MAPDL Commands
- API Reference

Translate by ff