

### Open the output database (.odb)

```

import sys
from odbAccess import *

# Open the output database
myOdb = openOdb('path\my file.odb')
    
```

Once you've created your odb object, you can access all the variables through the different attributes of the class. In the following cheatsheet, we only point out the way to access the results that mostly interest us such as **Field Outputs** and **History Outputs**

### Accessing to Field Outputs

In the **odb object** we will access the **steps** attribute, then the **frames** attribute to get our field output.

```

>>> steps = myOdb.steps
>>> frames = steps['my_step_name'].frames
    
```

### List of the main attributes in a step

- description
- domain
- frames
- historyRegion
- inertiaAboutCenter
- inertiaAboutOrigin
- loadCases
- mass
- massCenter
- name
- number
- previousStepName
- procedure
- timePeriod
- totalTime

### List of the main attributes in a frame

frames.description	Return description if given
frames.domain	TIME or FREQUENCY
frames.fieldOutputs	<b>return all the different outputs you specified in the simulation for each node and element</b>
frames.frameValue	return the simulation time as a float
frames.incrementNumber	return the increment number as an integer

### Which field output do we usually encounter

#### NODAL OUTPUTS

U	displacement
UR	rotational displacement
COORD	coordinates
RF	Reaction force

#### ELEMENT OUTPUT

S  
SMISES

Exemple to access the displacement along the x axis (U1), for a given node:

```

>>> myOdb = openOdb('path\my odb.odb')
>>> frame1 = myOdb.steps['Step-1'].frames[-1]
>>> t = frame1.frameValue
>>> u1 = frame1.fieldOutput['U'].values[27].data[0]
    
```

For all possible outputs, see the official Abaqus documentation

### Main attributes in a fieldOutput object

componentLabels	
description	
name	
type	
values	all the main values are stored in values [my Node].data

### What about the instance

**the Root Assembly** The odb object has an attribute rootAssembly, which gives informations about assembly such as:

elementSets

elements

instances

name

nodeSets

nodes

rigidBodies

sectionAssignment

surfaces

### Access History Output results

In your simulation, you may have specify **HistoryRegion** in which you historyOutput Objects. In such objects are store

```

region = myOdb.steps['stepName'].historyRegions['regionName'].historyOutputs['CVOL1']
    
```

Here is a quick exemple to access to time and

```

>>> myStep.historyRegions['regionName'].historyOutputs['CVOL1'].data
    
```

returns a tuple organized as:

```

((t1,CVOL1), (t2,CVOL1)...)
    
```

