PyFluent cheat sheet



Solver settings object interface Version: 0.13 (stable)

/ Launch Fluent locally	/ Define materials Use solver settings objects to define materials:	Apply solution settings
<pre>import ansys.fluent.core as pyfluent</pre>	<pre>solver.setup.materials.copy_database_material_by_name(turner#fluid#</pre>	Use solver settings objects to apply solution settings, initialize, and solve.
<pre>solver = pyfluent.launch_fluent(mode="solver", show_gui=True)</pre>	<pre>type="fluid", name="water-liquid") solver.setup.cell_zone_conditions.fluid["elbow-fluid"</pre>	<pre>solver.solution.initialization.hybrid_initialize() solver.solution.run_calculation.iterate(number_of_iterations=150)</pre>
/ Import mesh in launched session].material = "water-liquid"	
Read the available mesh file in the Fluent session:	/ Define boundary conditions	/ Postprocessing
<pre>mesh_filename = "example_file.msh.h5" solver.file.read(file_type="mesh", file_name= mesh_filename)</pre>	<pre>solver.setup.boundary_conditions.velocity_inlet["cold-inlet"</pre>	Postprocess data with the results object. For example, create and display contours on a plane:
Use specific methods to read case files and case data files:	<pre>_].vmag = { "option": "constant or expression", "constant": 0.4,</pre>	<pre>solver.results.graphics.contour["contour"] = {} solver.results.graphics.contour["contour"].print_state ()</pre>
<pre># e.g., read_case(), read_case_data() case_filename = "example_file.cas.h5" solver.file.read_case(file_type="case", file_name=</pre>	<pre>} solver.setup.boundary_conditions.velocity_inlet["cold-inlet"].ke_spec = "Intensity and Hydraulic Diameter"</pre>	<pre>solver.results.graphics.contour["contour"].field = " temperature" solver.results.graphics.contour["contour"]. surfaces_list = ["symmetry-xyplane"]</pre>
/ Enable heat transfer physics Enable heat transfer by activating the energy equation:	<pre>solver.setup.boundary_conditions.velocity_inlet["cold-inlet"].turb_intensity = 5 solver.setup.boundary_conditions.velocity_inlet["cold-inlet"</pre>	/ Temperature contour
solver.setup.models.energy.enabled = True].turb_hydraulic_diam = "4 [in]"	
<pre>/ Access the object state using pprint # >>> from pprint import pprint # >>> pprint(solver.setup.models.energy())</pre>	<pre>solver.setup.boundary_conditions.velocity_inlet["cold-inlet"].t = { "option": "constant or expression", "constant": 293.15, }</pre>	100 1000 11043 11042 11042 110424 11042
<pre>{ "enabled": True, "inlet_diffusion": True,</pre>	/ Modify cell zone conditions Use solver settings objects to modify cell zone conditions.	References from PyFluent documentation
"kinetic_energy": False, "pressure_work": False, "viscous_dissipation": False, }	<pre>solver.setup.cell_zone_conditions.fluid["elbow-fluid"]</pre>	Getting started Solver settings objects
		• Examples
	Getting started with PyEluent / PyEluent on GitHub / Visit	