PyAEDT cheat sheet

Version: main

Launch AEDT

Launch an AEDT application directly:

```
from ansys.aedt.core import Hfss
app = Hfss()
```

Open a Desktop session in graphical mode, open a project and connect to an active design:

```
from ansys.aedt.core import Desktop
desktop = Desktop(version="2025.2",
    new_desktop=True, non_graphical=False,
    close_on_exit=True)
app = desktop.load_project(
    project_file="project_path")
```

Open a Desktop session in non-graphical mode and create a new Maxwell 3D design:

```
from ansys.aedt.core import Desktop, Maxwell3d
desktop = Desktop(version="2025.2",
    new_desktop=True, non_graphical=True,
    close_on_exit=True)
app = Maxwell3d()
```

Connect to a running application with given project name and design name:

```
from ansys.aedt.core.generic.design_types import
    get_pyaedt_app
app = get_pyaedt_app(project_name="project_name",
    design="design_name")
```

Close the active AEDT session

```
app.release_desktop(close_projects=True,
    close_on_exit=True)
```

Work with variables

Create a variable that only applies to this design:

```
hfss["dim"] = "1mm"
```

Create a variable that applies at a project level:

```
hfss["$dim"] = "1mm"
```

Manage your variables:

```
hfss.variable_manager.variables
```

Handle your materials

Add a new material with custom properties:

```
material = hfss.materials.add_material("my_mat")
material.permittivity = 3.5
material.conductivity = 450000
material.permeability = 1.5
```

Create and manipulate geometry

Create a box and get object name:

```
box = hfss.modeler.create_box([1,1,1], [5,2,5],
    name="box", material="FR4_epoxy")
print(box.name)
```

Access edges or vertices data:

```
nb_edges = len(box.edges)
for edge in box.edges:
  print(edge.segment_info)
nb_vertices = len(box.vertices)
for vertex in box.vertices:
  print(vertex.position)
```

Define the solution setup

```
setup = hfss.create_setup("MySetup")
setup.props["Frequency"] = "50MHz"
setup.props["MaximumPasses"] = 10
```

Access the parametric sweep:

```
hfss.parametrics
```

Access the optimizations:

```
hfss.optimizations
```

Analyze the solution setup

```
hfss.analyze(cores=4)
```

Post processing

Post processing can be performed within and outside AEDT.

Report in AEDT

Create “Mag_E” report in a polyline:

```
test_points = [["0mm", "0mm", "0mm"], ["100mm",
    "20mm", "0mm"], ["71mm", "71mm", "0mm"],
    ["0mm", "100mm", "0mm"]]
p1 = hfss.modeler.create_polyline(test_points)
report = hfss.post.reports_by_category.fields(
  "Mag_E", setup.name + " : LastAdaptive", p1.name)
report.create()
```

Graphic operations

Visualize graphics objects and plot data within AEDT

```
field_plot = hfss.post.create_fieldplot_volume(
  ["box"], "Mag_E")
image_path = field_plot.export_image(
  r"C:\\workdir\\my_image.png")
```

Generate 2D/3D plots using third-party packages

```
hfss.post.plot_model_obj(objects=["box"],
    show_grid=True)
```

Get solution data

```
plot_data = hfss.get_traces_for_plot()
report = hfss.post.create_report(plot_data)
solution = report.get_solution_data()
plt = solution.plot(solution.expressions)
```

Generate PDF files using third-party packages

```
from ansys.aedt.core.visualization.plot.pdf import
    AnsysReport
pdf_report = AnsysReport(
  project_name=hfss.project_name,
  design_name=hfss.design_name)
pdf_report.create()
pdf_report.add_section()
pdf_report.add_chapter("HFSS Results")
pdf_report.add_text("This section contains plots.")
pdf_report.add_image(image_path)
pdf_report.save_pdf(file_path=r"C:\\workdir",
    file_name="report.pdf")
```