PyAnsys Units cheat sheet

Version: main

Defining a quantity using unit strings

```
from ansys.units import Quantity, UnitRegistry

meter = Quantity(value=1, units="m")
```

Defining a quantity using base dimensions

```
from ansys.units import Quantity, BaseDimensions,
    Dimensions

dims = BaseDimensions
dimensions = Dimensions({dims.LENGTH: 1,
    dims.TIME: -2})

acceleration = Quantity(value=3,
    dimensions=dimensions)

acceleration.value  # 3.0
acceleration.units.name  # "m s^-2"
```

Defining a quantity using UnitRegistry

```
from ansys.units import Quantity, UnitRegistry

ureg = UnitRegistry()
meter = Quantity(value=1, units=ureg.m)
```

Defining a quantity using NumPy array

```
from ansys.units import Quantity, UnitRegistry

length_array_quantity = Quantity(value=[1.0, 6.0,
    7.0], units="m")
length_array_quantity[1] # Quantity (6.0, "m")

time = Quantity(value=2, units="s")

speed = length_array_quantity / time
speed  # Quantity ([0.5 3. 3.5], "m s^-1")
```

Defining a quantity using the quantity table

```
from ansys.units import Quantity

torque = Quantity(5, quantity_table={"Torque": 1})

torque.value  # 5.0
torque.units.name  # "N m"
torque.units.si_units  # "kg m^2 s^-2"
```

Arithmetic operations

```
from ansys.units import Quantity
import math

meter = Quantity(value=1, units="m")

m_ml = meter * 2  # (2.0, "m")
m_dv = meter / 2  # (0.5, "m")
m_sq = meter**2  # (1.0, "m^2")

meter = Quantity(value=1, units="m")
foot = Quantity(value=1, units="ft")

meter + foot  # Quantity (1.3048, "m")
foot + meter  # Quantity (4.2808398950131235, "ft")

deg = Quantity(90, "degree")
math.sin(deg)  # 1.0
```

Pre-defined unit systems

```
from ansys.units import UnitSystem

si = UnitSystem(system="SI")
si.LENGTH # "m"

cgs = UnitSystem(system="CGS")
cgs.LENGTH # "cm"

bt = UnitSystem(system="BT")
bt.LENGTH # "ft"
```

Custom unit systems

```
from ansys.units import BaseDimensions,
    UnitSystem, UnitRegistry

ureg = UnitRegistry()
dims = BaseDimensions

sys = UnitSystem(
    base_units={
        dims.MASS: ureg.slug,
        dims.LENGTH: ureg.ft,
        dims.TEMPERATURE: ureg.R,
        dims.TEMPERATURE_DIFFERENCE: ureg.delta_R,
        dims.CHEMICAL_AMOUNT: ureg.slugmol,
    }
)

sys.LENGTH
```

Convert quantity to other units

```
from ansys.units import Quantity

flbs = Quantity(1, "lb ft^-1 s^-1")
flbs.value  # 1

pas = flbs.to("Pa s")
pas.value  # 1.4881639435695542
pas.units.name  # 'Pa s'
```

Convert units using unit systems

```
from ansys.units import Unit, UnitSystem

si = Unit("m s^-1")
bt = UnitSystem(system="BT")
si.convert(bt)
```

Temperature and temperature difference

```
from ansys.units import Quantity, UnitRegistry

ureg = UnitRegistry()

K = Quantity(value=280, units=ureg.K)
K2 = Quantity(value=295, units=ureg.K)
delta_K = Quantity(value=5, units=ureg.delta_K)

K2_sb_K = K2 - K  # (15.0, "delta_K")
K_ad_delta_K = K + delta_K  # (285.0, "K")
K2_sb_delta_K = K2 - delta_K  # (290.0, "K")
delta_K_ad_delta_K = delta_K + delta_K  # (10.0,
    "delta_K")
delta_K_sb_delta_K = delta_K - delta_K  # (0.0,
    "delta_K")
```

Temperature values below absolute zero

```
from ansys.units import Quantity, UnitRegistry

ureg = UnitRegistry()

K = Quantity(value=-2, units=ureg.K) # (-2,
    delta_K)
C = Quantity(value=-2, units=ureg.C) # (-275,
    delta_C)
F = Quantity(value=-2, units=ureg.F) # (-500,
    delta_F)
R = Quantity(value=-2, units=ureg.R) # (-2,
    delta_R)
```

Angle as non-dimension

```
import ansys.units as pyunits

degree = pyunits.Quantity(1.0, "degree")
degree.is_dimensionless # True
degree + 1 # Quantity (58.29577951308232, "degree")
```

Angle as dimension

```
import os
os.environ["PYANSYS_UNITS_ANGLE_AS_DIMENSION"] =
    "1"
import ansys.units as pyunits

degree = pyunits.Quantity(1.0, "degree")
degree.is_dimensionless # False
degree + 1 # Not allowed
```