report_example

August 12, 2024

[1]:
```
import forallpeople
forallpeople.environment('Structural',top_level=True)
#Useful extra units
cm = 1e-2*m #type: ignore
from IPython.display import Markdown, display
import handcalcs.render
```

Calculations Let the force be represented as:

$$F = ma$$

Where: - m is the mass in kg - a is the acceleration in $\mathrm{m/s^2}$

[30]:
```
# Example calculation
mass=100*kg
a = 9.81*m/s**2
force = mass * a
force=2*kN
display(Markdown(f"The result is: $F$={round(force,1)}"))
t = 3.75*mm #Espesor
H = 200*mm #Altura
B = 100*mm #Ancho
I_x = B*H**3/12 - (B - t)*(H - t)**3/12
I_y = H*B**3/12 - (H - t)*(B - t)**3/12
display(Markdown(f"The result is: $I_x$={round(I_x,0)}"))
display(Markdown(f"The result is: $I_y$={round(I_y,0)}"))
# Cambio de unidad
force.to(unit_name="kip")
```

The result is: F=2.0 kN

The result is: I_x=6042122 mm

The result is: I_y=2084212 mm

[30]:
0.450 kip

[16]:
```
%%capture
!jupyter nbconvert --to markdown --no-input report_example.ipynb
```

[33]:
```
%%capture
!jupyter nbconvert --no-input report_example.ipynb --to html
```

[38]:
```
%%capture
!jupyter nbconvert --to html  --no-input report_example.ipynb
```

[ ]:
```
%%render 2
mass=100*kg
a = 9.81*m/s**2
force = mass * a
```

[ ]:
```
%%render 2
mass=100*kg
a = (9.81*m/s**2)
force = (mass * a)
```

[ ]:
```
%%render params 2
mass=100*kg
a = 9.81*m/s**2
force = mass * a
```

[ ]:
```
%%render symbolic
mass=100*kg #Comment in line
a = 9.81*m/s**2
force = mass * a
```

[35]:
```
%%render params 2
force = mass * a
```

$$\text{force} = 981.00 \text{ N}$$

[36]:
```
%%render params 2
#Tubo 200x100x3.75mm
t = 3.75*mm #Espesor
H = 200*mm #Altura
B = 100*mm #Ancho
```

$$t = 3.75 \text{ mm} \ \text{(Espesor)} \quad H = 200.00 \text{ mm} \ \text{(Altura)} \quad B = 100.00 \text{ mm} \ \text{(Ancho)}$$

[37]:
```
%%render
I_x = B*H**3/12 - (B - t)*(H - t)**3/12
I_y = H*B**3/12 - (H - t)*(B - t)**3/12
```

$$
\begin{aligned}
I_x &= B \cdot \frac{(H)^3}{12} - (B-t) \cdot \frac{(H-t)^3}{12} \\
&= 100.000 \text{ mm} \cdot \frac{(200.000 \text{ mm})^3}{12} - (100.000 \text{ mm} - 3.750 \text{ mm}) \cdot \frac{(200.000 \text{ mm} - 3.750 \text{ mm})^3}{12} \\
&= 6042122.192 \text{ mm}^4
\end{aligned}
$$

$$
\begin{aligned}
I_y &= H \cdot \frac{(B)^3}{12} - (H-t) \cdot \frac{(B-t)^3}{12} \\
&= 200.000 \text{ mm} \cdot \frac{(100.000 \text{ mm})^3}{12} - (200.000 \text{ mm} - 3.750 \text{ mm}) \cdot \frac{(100.000 \text{ mm} - 3.750 \text{ mm})^3}{12} \\
&= 2084212.036 \text{ mm}^4
\end{aligned}
$$

[22]:
```
import numpy as np

from pylatex import Document, Math, Matrix, Section, Subsection, VectorName,␣
 ↪NoEscape, Figure
from pylatex.utils import NoEscape
import matplotlib
matplotlib.use("Agg")  # Not to use X server. For TravisCI.
import matplotlib.pyplot as plt  # noq

if __name__ == "__main__":

    geometry_options = {"right": "2cm", "left": "2cm"}
    doc = Document("pylatex_example", geometry_options=geometry_options)
    section = Section("Numpy tests")
    subsection = Subsection("Array")

    a = np.array([[100, 10, 20]]).T
    vec = Matrix(a)
    vec_name = VectorName("a")
    math = Math(data=[vec_name, "=", vec])

    subsection.append(math)
    section.append(subsection)

    subsection = Subsection("Matrix")
    M = np.matrix([[2, 3, 4], [0, 0, 1], [0, 0, 2]])
    matrix = Matrix(M, mtype="b")
    math = Math(data=["M=", matrix])
    math = Math(data=[force.value/1000," kN"])
```

```
    subsection.append(math)
    section.append(subsection)

    subsection = Subsection("Product")

    math = Math(data=["M", vec_name, "=", Matrix(M * a)])
    subsection.append(math)

    section.append(subsection)

    doc.append(section)

    doc.append(
        NoEscape(
            r"""
        The following is a demonstration of a custom \LaTeX{}
        command with a couple of parameters.\\
        """
        )
    )

    x = [0, 1, 2, 3, 4, 5, 6]
    y = [15, 2, 7, 1, 5, 6, 9]

    plt.plot(x, y)

    doc.append("Introduction.")

    with doc.create(Section("I am a section")):
        doc.append("Take a look at this beautiful plot:")

        with doc.create(Figure(position="h")) as plot:
            plot.add_plot(width=NoEscape(r"1\textwidth"))
            plot.add_caption("I am a caption.")

        doc.append("Created using matplotlib.")

    doc.append("Conclusion.")

    doc.generate_pdf(clean_tex=False)
```