handcalcs

September 20, 2020

[2]:
```
1.0
```

isso não é um input! isso é um markdown.

$$peso_{andre} = 73.000 \text{ kg} \quad \text{(pesei ontem)}$$

$$peso_{vitoria} = 58.000 \text{ kg}$$

$$altura_{andre} = 1.750 \text{ m}$$

$$altura_{vitoria} = 1.580 \text{ m}$$

$$IMC_{andre} = \frac{peso_{andre}}{altura_{andre}} = \frac{73.000 \text{ kg}}{1.750 \text{ m}} = 41.714 \text{ kg} \cdot \text{m}^{-1}$$

```
␣
↪---------------------------------------------------------------------------

NameError                                 Traceback (most recent call␣
↪last)

<ipython-input-4-9380ca62b029> in <module>
  1 # short
  2
----> 3 g = integrate.quad(f, 11, 12)


NameError: name 'f' is not defined
```

```
␣
↪---------------------------------------------------------------------------

KeyError                                  Traceback (most recent call␣
↪last)

<ipython-input-4-76e4fd08c2bc> in <module>
```

```
   ----> 1 get_ipython().run_cell_magic('render', '', '# short\n\ng = integrate.
↪quad(f, 11, 12)\n')


␣
↪~\AppData\Roaming\Python\Python38\site-packages\IPython\core\interactiveshell.
↪py in run_cell_magic(self, magic_name, line, cell)
      2369             with self.builtin_trap:
      2370                 args = (magic_arg_s, cell)
   -> 2371                 result = fn(*args, **kwargs)
      2372             return result
      2373


      ~\anaconda3\lib\site-packages\handcalcs\render.py in render(line, cell)
       48     # Do the handcalc conversion
       49     renderer = hand.LatexRenderer(cell, var_dict)
   ---> 50     latex_code = renderer.render()
       51
       52     # Display, but not as an "output"


      ~\anaconda3\lib\site-packages\handcalcs\handcalcs.py in render(self)
      151
      152     def render(self):
   --> 153         return latex(self.source, self.results, self.precision)
      154
      155


      ~\anaconda3\lib\site-packages\handcalcs\handcalcs.py in␣
↪latex(raw_python_source, calculated_results, precision)
      162     cell = categorize_raw_cell(source, calculated_results)
      163     cell = categorize_lines(cell)
   --> 164     cell = convert_cell(cell)
      165     cell = format_cell(cell)
      166     return cell.latex_code


      ~\anaconda3\lib\functools.py in wrapper(*args, **kw)
      873                             '1 positional argument')
      874
   --> 875         return dispatch(args[0].__class__)(*args, **kw)
      876
      877     funcname = getattr(func, '__name__', 'singledispatch function')
```

```
      ~\anaconda3\lib\site-packages\handcalcs\handcalcs.py in␣
↪convert_calc_cell(cell)
      449     incoming = deque([])
      450     for line in outgoing:
  --> 451         incoming.append(convert_line(line, calculated_results))
      452     cell.lines = incoming
      453     return cell


      ~\anaconda3\lib\functools.py in wrapper(*args, **kw)
      873                             '1 positional argument')
      874
  --> 875         return dispatch(args[0].__class__)(*args, **kw)
      876
      877     funcname = getattr(func, '__name__', 'singledispatch function')


      ~\anaconda3\lib\site-packages\handcalcs\handcalcs.py in␣
↪convert_calc(line, calculated_results)
      513         result,
      514     ) = line.line  # Unpack deque of form [[calc_line, …], ['=',␣
↪'result']]
  --> 515     symbolic_portion, numeric_portion = swap_calculation(line_deque,␣
↪calculated_results)
      516     line.line = symbolic_portion + numeric_portion + result
      517     return line


      ~\anaconda3\lib\site-packages\handcalcs\handcalcs.py in␣
↪swap_calculation(calculation, calc_results)
     1325     """Returns the python code elements in the deque converted into
     1326     latex code elements in the deque"""
  -> 1327     calc_w_integrals_preswapped = swap_integrals(calculation,␣
↪calc_results)
     1328     symbolic_portion =␣
↪swap_symbolic_calcs(calc_w_integrals_preswapped)
     1329     calc_drop_decl = deque(


      ~\anaconda3\lib\site-packages\handcalcs\handcalcs.py in␣
↪swap_integrals(calculation, calc_results)
     1394                 skip_next = True
     1395                 function_name = next_item[0]
  -> 1396                 function = calc_results[function_name]
     1397                 function_source = (
     1398                     inspect.getsource(function).split("\n")[1].
↪replace("return", "")
```

```
        KeyError: 'f'
```

```
\[
\begin{aligned}
g &= \int_{ 0 } ^ { 12 } \left( x \right) ^{ 2 } + 10 \cdot x \; dx = \int_{ 0 }
^ { 12 } \left( x \right) ^{ 2 } + 10 \cdot x \; dx &= (1295.9999999999995,
1.4388490399142024e-11)
\end{aligned}
\]
```

$$\Sigma = \text{sum}\left(x\right) = \text{sum}\left(\left[1,\ 2,\ 3\right]\right) \qquad = 6$$

$$Len = \text{len}\left(x\right) = \text{len}\left(\left[1,\ 2,\ 3\right]\right) \qquad = 3$$

$$Mean = \frac{\Sigma}{Len} = \frac{6}{3} \qquad = 2.0$$