bpy.utils submodule (bpy.utils.units)

This module contains some data/methods regarding units handling.

bpy.utils.units.**categories**

Constant value bpy.utils.units.categories(NONE='NONE', LENGTH='LENGTH', AREA='AREA', VOLUME='VOLUME', MASS='MASS' ROTATION='ROTATION', TIME='TIME', TIME_ABSOLUTE='TIME_ABSOLUTE', VELOCITY='VELOCITY', ACCELERATION='ACCELERATION', CAMERA='CAMERA', POWER='POWER', TEMPERATURE='TEMPERATURE', WAVELENGTH='WAVELENGTH', COLOR_TEMPERATURE='COLOR_TEMPERATURE', FREQUENCY='FREQUENCY')

bpy.utils.units.**systems**

Constant value bpy.utils.units.systems(NONE='NONE', METRIC='METRIC', IMPERIAL='IMPERIAL')

bpy.utils.units.**to_string(unit_system, unit_category, value, precision=3, split_unit=False, compatible_unit=False)**

Convert a given input float value into a string with units.

PARAMETERS:

- **unit_system** (*str*) – The unit system, from `bpy.utils.units.systems`.
- **unit_category** (*str*) – The category of data we are converting (length, area, rotation, etc.), from `bpy.utils.units.categories`.
- **value** (*float*) – The value to convert to a string.
- **precision** (*int*) – Number of digits after the comma.
- **split_unit** (*bool*) – Whether to use several units if needed (1m1cm), or always only one (1.01m).
- **compatible_unit** (*bool*) – Whether to use keyboard-friendly units (1m2) or nicer utf-8 ones (1m²).

RETURNS:

The converted string.

RETURN TYPE:

str

RAISES:

ValueError – if conversion fails to generate a valid Python string.

bpy.utils.units.**to_value(unit_system, unit_category, str_input, str_ref_unit=None)**

Convert a given input string into a float value.

PARAMETERS:

- **unit_system** (*str*) – The unit system, from `bpy.utils.units.systems`.
- **unit_category** (*str*) – The category of data we are converting (length, area, rotation, etc.), from `bpy.utils.units.categories`.
- **str_input** (*str*) – The string to convert to a float value.
- **str_ref_unit** (*str* | *None*) – A reference string from which to extract a default unit, if none is found in `str_input`.

RETURNS:

The converted/interpreted value.

RETURN TYPE:

float

RAISES:

ValueError – if conversion fails to generate a valid Python float value.