

NAME

codata - fundamental physical constants

LIBRARY

Codata (-libcodata, -lcodata)

SYNOPSIS

```
use codata
include "codata.h"
import pycodata
```

DESCRIPTION

codata is a Fortran library providing the fundamental physical constants according to CODATA <<https://www.nist.gov/programs-projects/codata-values-fundamental-physical-constants>>. A C API allows usage from C, or can be used as a basis for other wrappers. Python wrapper allows easy usage from Python.

The latest *codata* constants 2022 <<https://pml.nist.gov/cuu/Constants>> were integrated in stdlib <<https://github.com/fortran-lang/stdlib/releases/tag/v0.7.0>>. The constants are implemented as derived type which carries the name, the value, the uncertainty and the unit. This library is complementary to the constants defined in the stdlib by providing older values for the constants.

The latest values (2022) do not have the year as a suffix in their name. Older values can be used and they feature the year as a suffix in their name.

NOTES

To use *codata* within your fpm <<https://github.com/fortran-lang/fpm>> project, add the following lines to your file:

```
[dependencies]
codata = { git="https://github.com/MilanSkocic/codata.git" }
```

EXAMPLE

Example in Fortran

```
program example_in_f
use iso_fortran_env
use codata
implicit none

print '(A)', '# ##### EXAMPLE IN FORTRAN ######'

print '(A)', '# VERSION'
print *, "version = ", get_version()

print '(A)', '# CONSTANTS'
print *, "c = ", SPEED_OF_LIGHT_IN_VACUUM%value

print '(A)', '# UNCERTAINTY'
print *, "u(c) = ", SPEED_OF_LIGHT_IN_VACUUM%uncertainty

print '(A)', '# OLDER VALUES'
print '(A, F23.16)', "Mu_2022(latest) = ", MOLAR_MASS_CONSTANT%value
print '(A, F23.16)', "Mu_2018 = ", MOLAR_MASS_CONSTANT_2018%value
print '(A, F23.16)', "Mu_2014 = ", MOLAR_MASS_CONSTANT_2014%value
print '(A, F23.16)', "Mu_2010 = ", MOLAR_MASS_CONSTANT_2010%value
end program
```

Example in C:

```
#include <stdio.h>
#include "codata.h"

int main(void){

    printf("##### EXAMPLE IN C #####\n");

    printf("%s0,# VERSION");
    printf("version = %s0, codata_get_version());

    printf("%s0,# CONSTANTS");
    printf("c = %f0, SPEED_OF_LIGHT_IN_VACUUM.value);

    printf("%s0,# UNCERTAINTY");
    printf("u(c) = %f0, SPEED_OF_LIGHT_IN_VACUUM.uncertainty);

    printf("%s0,# OLDER VALUES");
    printf("Mu_2022(latest) = %23.16f0, MOLAR_MASS_CONSTANT.value);
    printf("Mu_2018 = %23.16f0, MOLAR_MASS_CONSTANT_2018.value);
    printf("Mu_2014 = %23.16f0, MOLAR_MASS_CONSTANT_2014.value);
    printf("Mu_2010 = %23.16f0, MOLAR_MASS_CONSTANT_2010.value);

    return 0;
}
```

Example in Python:

```
import sys
sys.path.insert(0, "../py/src/")
import pycodata

print("##### EXAMPLE IN PYTHON #####")
print("# VERSION")
print(f"version = {pycodata.__version__} ")

print("# Constants")
print(f"c = {pycodata.SPEED_OF_LIGHT_IN_VACUUM['value']}")

print("# UNCERTAINTY")
print(f"u(c) = {pycodata.SPEED_OF_LIGHT_IN_VACUUM['uncertainty']}")

print("# OLDER VALUES")
print(f"Mu_2022 = {pycodata.MOLAR_MASS_CONSTANT['value']}")
print(f"Mu_2018 = {pycodata.constants_2018.MOLAR_MASS_CONSTANT_2018['value']}")
print(f"Mu_2014 = {pycodata.constants_2014.MOLAR_MASS_CONSTANT_2014['value']}")
print(f"Mu_2010 = {pycodata.constants_2010.MOLAR_MASS_CONSTANT_2010['value']}
```

SEE ALSO

[gsl\(3\)](#)