

# GRIB and ecCodes – a very short introduction

A hands-on introduction to Numerical Weather Prediction Models:  
Understanding and Experimenting 13-17 November 2023

Paul Dando

Computing and Software Support Team – User Services Section



# What is GRIB ?

- GRIB – “**G**eneral **R**egularly-distributed **I**nformation in **B**inary form”
  - Code defined by the WMO / CBS in 1985 for the exchange of large volumes of gridded data
  - Machine independent
  - Requires software for encoding and decoding

47 52 49 42	00 00 66 01	00 00 1C 01	62 01 FF 80	33 6D 00 01	06 0C	GRIB f b ~Ä3m
05 0C 00 0C	00 C8 05 00	00 00 15 00	00 00 00 00	32 02 2B 0A	00 F8	» 2 + ~
01 90 80 33	C2 00 16 76	88 00 68 1A	00 76 F2 00	64 00 64 40	00 00	éÄ3- v à h vÚ d d@
00 00 80 55	F0 80 9C 40	00 00 00 00	43 3E B0 71	00 00 00 00	00 00	ÄU•Äú@ C>∞q
0C 08 80 11	3C 1F 09 7C	00 00 37 37	37 37			Ä <   7777

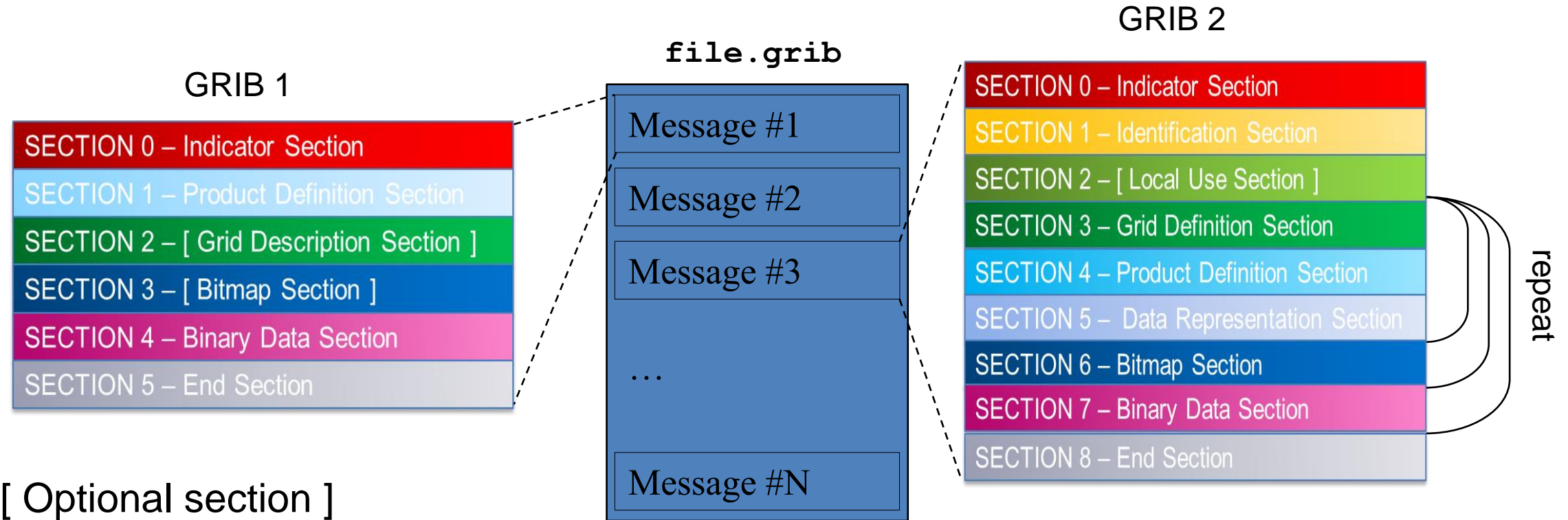
- Currently there are two different coding standards
  - GRIB edition 1
    - Used currently for ECMWF operational pressure level data, (most) surface level data and all wave data
  - GRIB edition 2
    - Used for ECMWF operational model level data since 11 May 2011 and some new surface data

**ECMWF plans to migrate fully to GRIB edition 2 by 2026 !**

# GRIB structure

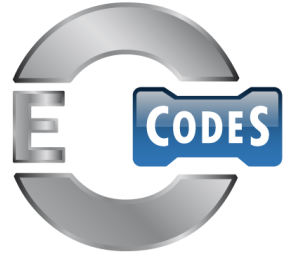
- Each message contains several sections
- A file can contain a mix of editions 1 and 2
- A file may contain one or more GRIB messages

47 52 49 42 00 00 66 01 00 00 1C 01 62 01 FF 80 33 6D 00 01 06 0C	GRIB f b ~Ä3m
05 0C 00 0C 00 C8 05 00 00 00 15 00 00 00 00 00 32 02 2B 0A 00 F8	» 2 + -
01 90 80 33 C2 00 16 76 88 00 68 1A 00 76 F2 00 64 00 64 40 00 00	éÄ3- vâ h vÚ d d@
00 00 80 55 F0 80 9C 40 00 00 00 00 43 3E B0 71 00 00 00 00 00 00	ÄUÄú@ C>∞q
0C 08 80 11 3C 1F 09 7C 00 00 37 37 37 37	Ä <   7777

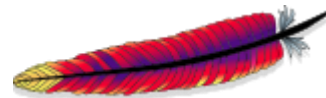


[ Optional section ]

# What is ecCodes ?



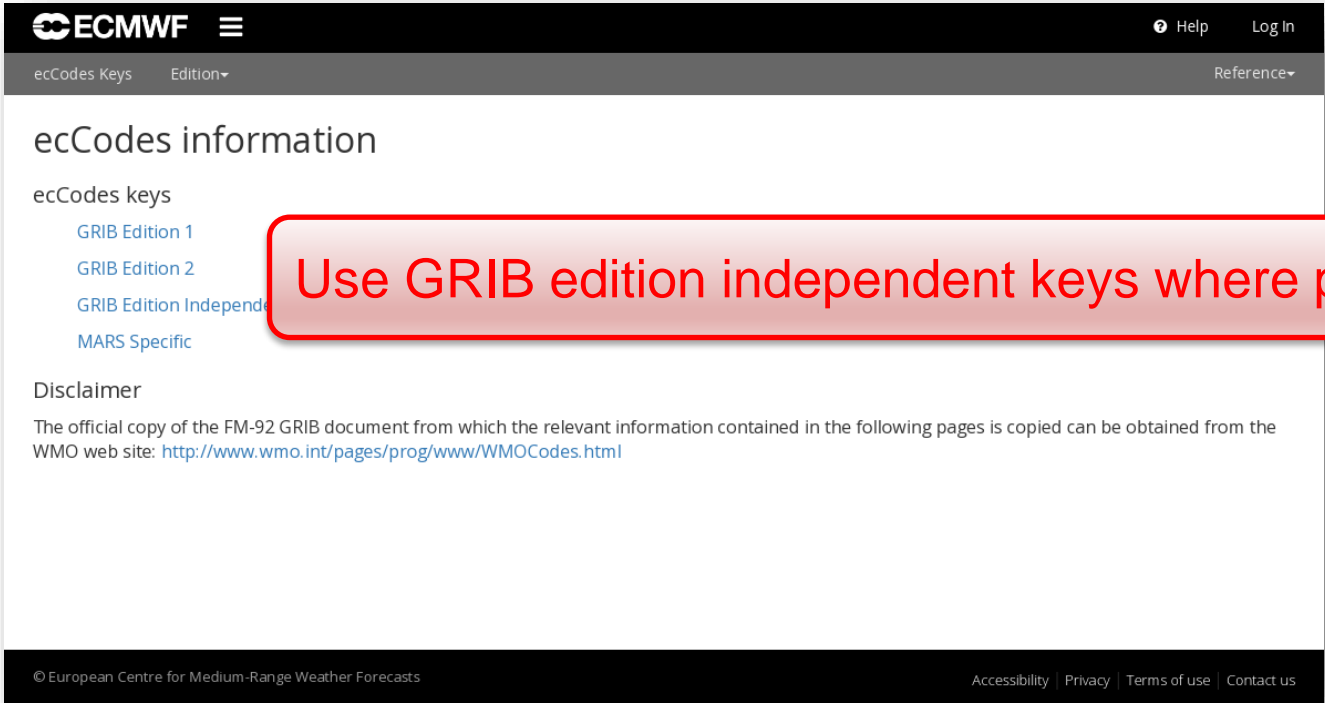
- **ecCodes** is a package developed by ECMWF for decoding and encoding messages in the following formats:
  - WMO FM-92 GRIB edition 1 and edition 2 (and edition 3)
  - WMO FM-94 BUFR edition 3 and edition 4
  - WMO GTS abbreviated header
- The package includes:
  - command line tools (the GRIB and BUFR Tools) to provide a quick and easy way to manipulate data
  - C, Python and Fortran 90 APIs
- Provides an easy and reliable way of encoding and decoding
  - GRIB 1 and GRIB 2 (and GRIB 3) messages using the **SAME** function calls
- Released under the Apache 2.0 license
- Available on GitHub



## ecCodes key / value based approach

- ecCodes uses a key/value based approach to access information in a GRIB message
  - numberOfPointsAlongAParallel => Number of points along a parallel
  - numberOfPointsAlongAMeridian => Number of points along a meridian
  - ...

<https://codes.ecmwf.int/grib/>



The screenshot shows the ECMWF website's 'ecCodes Keys' page. The header includes the ECMWF logo, a hamburger menu, and links for 'Help' and 'Log In'. Below the header, there are navigation links for 'ecCodes Keys', 'Edition', and 'Reference'. The main content area is titled 'ecCodes information' and lists 'ecCodes keys' with sub-links for 'GRIB Edition 1', 'GRIB Edition 2', 'GRIB Edition Independent', and 'MARS Specific'. A 'Disclaimer' section follows, stating that the official copy of the FM-92 GRIB document can be obtained from the WMO website. The footer contains copyright information for the European Centre for Medium-Range Weather Forecasts and links for 'Accessibility', 'Privacy', 'Terms of use', and 'Contact us'.

Use GRIB edition independent keys where possible !



## ecCodes keys – parameter

- The definition of the parameter is very different in the two GRIB editions

GRIB 1 keys	GRIB 2 keys
centre	discipline
table2Version	parameterCategory
indicatorOfParameter	parameterNumber
levelType	typeOfFirstFixedSurface
level	scaleFactorOfFirstFixedSurface
...	scaledValueOfFirstFixedSurface
	typeOfSecondFixedSurface
	scaleFactorOfSecondFixedSurface
	scaledValueOfSecondFixedSurface
	...

## ecCodes keys – parameter namespace

- ecCodes provides some **edition-independent** keys to identify a parameter

Key name	Example value
paramId	151
shortName	msl
centre	ecmf (or 98)
name	Mean sea level pressure
unit	Pa

- This set of keys is the parameter **namespace**
- ecCodes provides a number of additional namespaces:
  - ‘parameter’, ‘time’, ‘geography’, ‘vertical’, ‘statistics’, ‘mars’

## ecCodes – using GRIB edition-independent keys

```
...  
edition = codes_get(gid,"edition")  
  
if edition == 1:  
    paramNum = codes_get(gid,"indicatorOfparameter")  
    table = codes_get(gid,"table2Version")  
  
elif edition == 2:  
    paramNum = codes_get(gid,"parameterNumber")  
    category = codes_get(gid,"parameterCategory")  
  
...
```



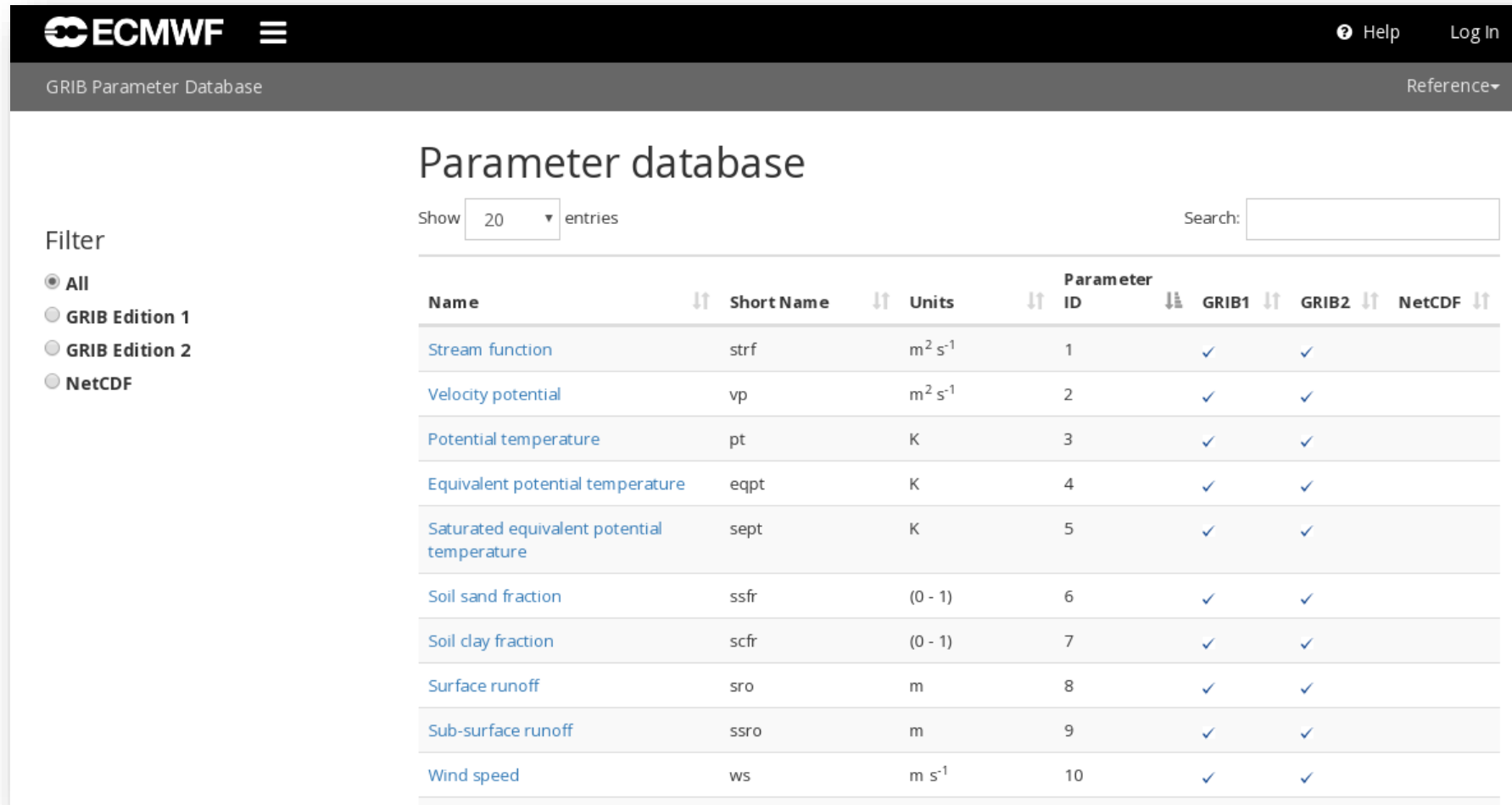
```
...  
shortName = codes_get(gid,"shortName")  
  
paramId = codes_get(gid,"paramId")  
  
units = codes_get(gid,"units")  
  
...
```





# GRIB parameters – the Parameter Database

<https://codes.ecmwf.int/grib/param-db/>



The screenshot shows the ECMWF GRIB Parameter Database interface. At the top, there is the ECMWF logo and navigation links for Help and Log In. Below the header, the page title is "GRIB Parameter Database" with a "Reference" dropdown. The main content area is titled "Parameter database" and includes a "Show 20 entries" dropdown and a search box. On the left, there is a "Filter" section with radio buttons for "All", "GRIB Edition 1", "GRIB Edition 2", and "NetCDF". The main part of the page is a table listing parameters with columns for Name, Short Name, Units, Parameter ID, GRIB1, GRIB2, and NetCDF.

Name	Short Name	Units	Parameter ID	GRIB1	GRIB2	NetCDF
Stream function	strf	$\text{m}^2 \text{s}^{-1}$	1	✓	✓	
Velocity potential	vp	$\text{m}^2 \text{s}^{-1}$	2	✓	✓	
Potential temperature	pt	K	3	✓	✓	
Equivalent potential temperature	eqpt	K	4	✓	✓	
Saturated equivalent potential temperature	sept	K	5	✓	✓	
Soil sand fraction	ssfr	(0 - 1)	6	✓	✓	
Soil clay fraction	scfr	(0 - 1)	7	✓	✓	
Surface runoff	sro	m	8	✓	✓	
Sub-surface runoff	ssro	m	9	✓	✓	
Wind speed	ws	$\text{m s}^{-1}$	10	✓	✓	

# GRIB Tools

- All of the tools use a common syntax

```
grib_<tool> [options] grib_file grib_file ... [output_grib]
```

- There is tools for getting information about the ecCodes installation
  - `codes_info`
- There are tools to inspect the content of GRIB messages
  - `grib_ls` `grib_dump` `grib_get` `grib_get_data`
- There are tools for counting and copying some messages
  - `grib_count`, `grib_copy` `codes_split_file`
- There is a tool for comparing GRIB messages
  - `grib_compare`
- There are tools for making changes to the content of a GRIB message and converting from GRIB to NetCDF
  - `grib_set`, `grib_filter`, `grib_to_netcdf`

# General framework for using the Python API

- Open one or more GRIB files (for read or write)
- Calls to load one or more GRIB messages into memory
  - Two main routines: `codes_grib_new_from_file` / `codes_new_from_index`
  - These return a unique `identifier` used to manipulate the loaded GRIB messages
- Calls to decode / encode the loaded GRIB messages – only `loaded` GRIB messages can be decoded or encoded
  - `codes_get` / `codes_get_array`
  - `codes_set` / `codes_set_array`
  - Decode and encode only what you need and not the full message !
- Release the loaded GRIB messages:
  - `codes_release`
- Close the opened GRIB files

# ecCodes – Python API decoding example

```
import sys
from eccodes import *
```

Decode as little as possible !  
You will never decode all the loaded GRIB message

```
# Load all the GRIB messages contained in file.grib1
ifile = open('file.grib1','rb')
while 1:
    igrib = codes_grib_new_from_file(ifile)
    if igrib is None: break
```

Loop on all the messages in a file.  
A new grib message is loaded from file. igrib is the grib id to be used in subsequent calls

```
# Decode data from the loaded message
date = codes_get( igrib , "dataDate")
levtype = codes_get(igrib, "typeOfLevel")
level = codes_get(igrib, "level")
values = codes_get_array(igrib, "values")
print (f'{date} {levtype} {level} {values[0]:.5f} {values[-1]:.5f}')
```

Values returned as an array

```
# Release
codes_release(igrib)
ifile.close()
```

Release the memory !

## ecCodes and cfgrib

- cfgrib provides a Python interface to map GRIB files to the netCDF Common Data Model following the CF Convention using ecCodes.
- The high level enables the engine='cfgrib' option to read GRIB files with xarray
- Inspired by netCDF4-python and h5netcdf
- ecCodes used for low-level decoding



- Install from conda

```
$ conda install -c conda-forge cfgrib
```

- Or PyPI

```
$ pip install cfgrib
```

<https://github.com/ecmwf/cfgrib>

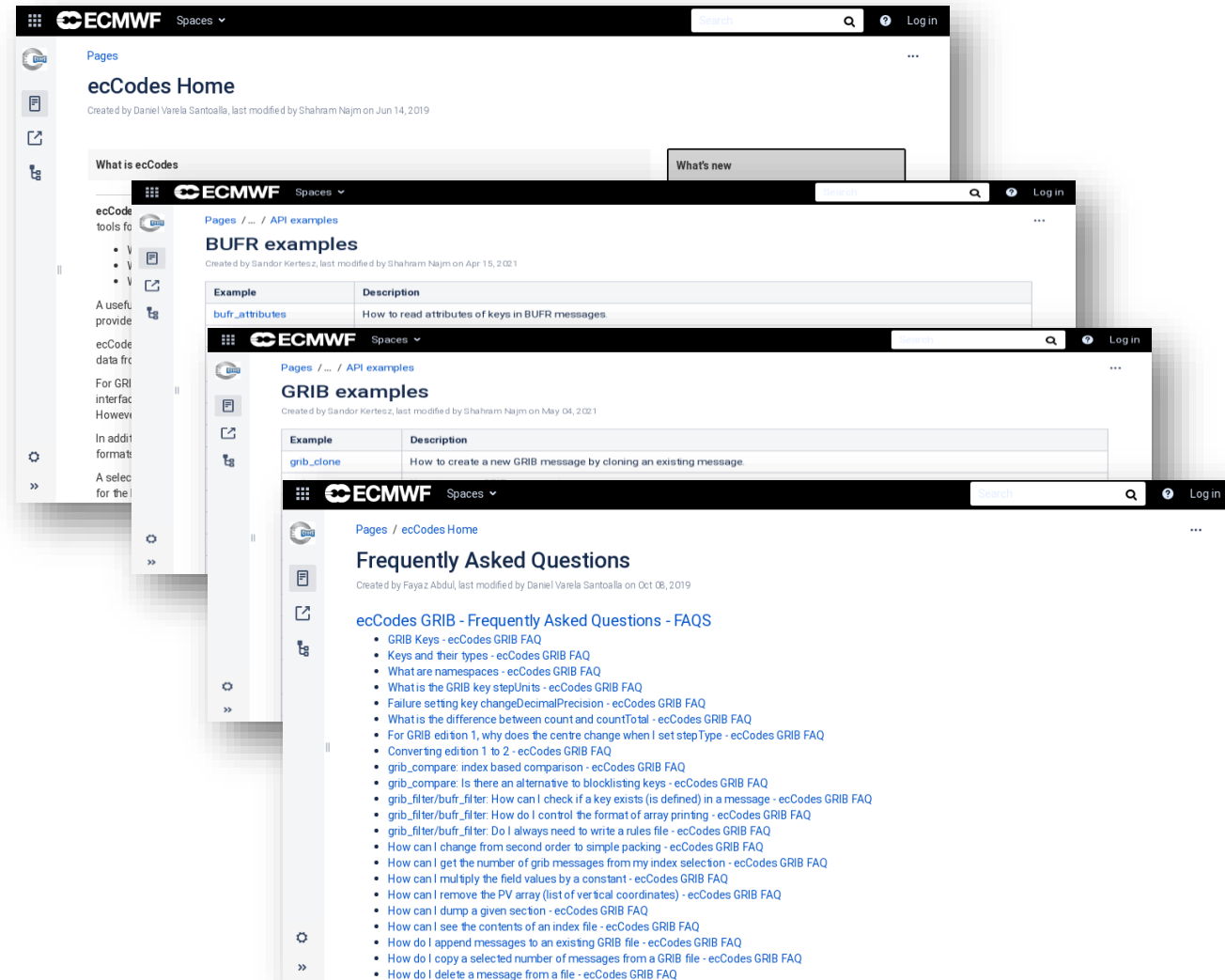
```
>>> import xarray as xr
>>> ds = xr.open_dataset('era5-levels-members.grib',
engine='cfgrib')
>>> ds
<xarray.Dataset>
Dimensions:          (isobaricInhPa: 2, latitude: 61,
longitude: 120, number: 10, time: 4)
Coordinates:
  * number            (number) int64 0 1 2 3 4 5 6 7 8 9
  * time              (time) datetime64[ns] 2017-01-01
... 2017-01-02T12:00:00
  step              timedelta64[ns] ...
  * isobaricInhPa    (isobaricInhPa) float64 850.0 500.0
  * latitude          (latitude) float64 90.0 87.0 84.0
81.0 ... -84.0 -87.0 -90.0
  * longitude         (longitude) float64 0.0 3.0 6.0 9.0
... 351.0 354.0 357.0
  valid_time        (time) datetime64[ns] ...
```

# ecCodes documentation

<https://confluence.ecmwf.int/display/ECC/ecCodes+Home>

- GRIB and BUFR API examples
  - C
  - Python
  - Fortran 90
- Command-line tools
- FAQs

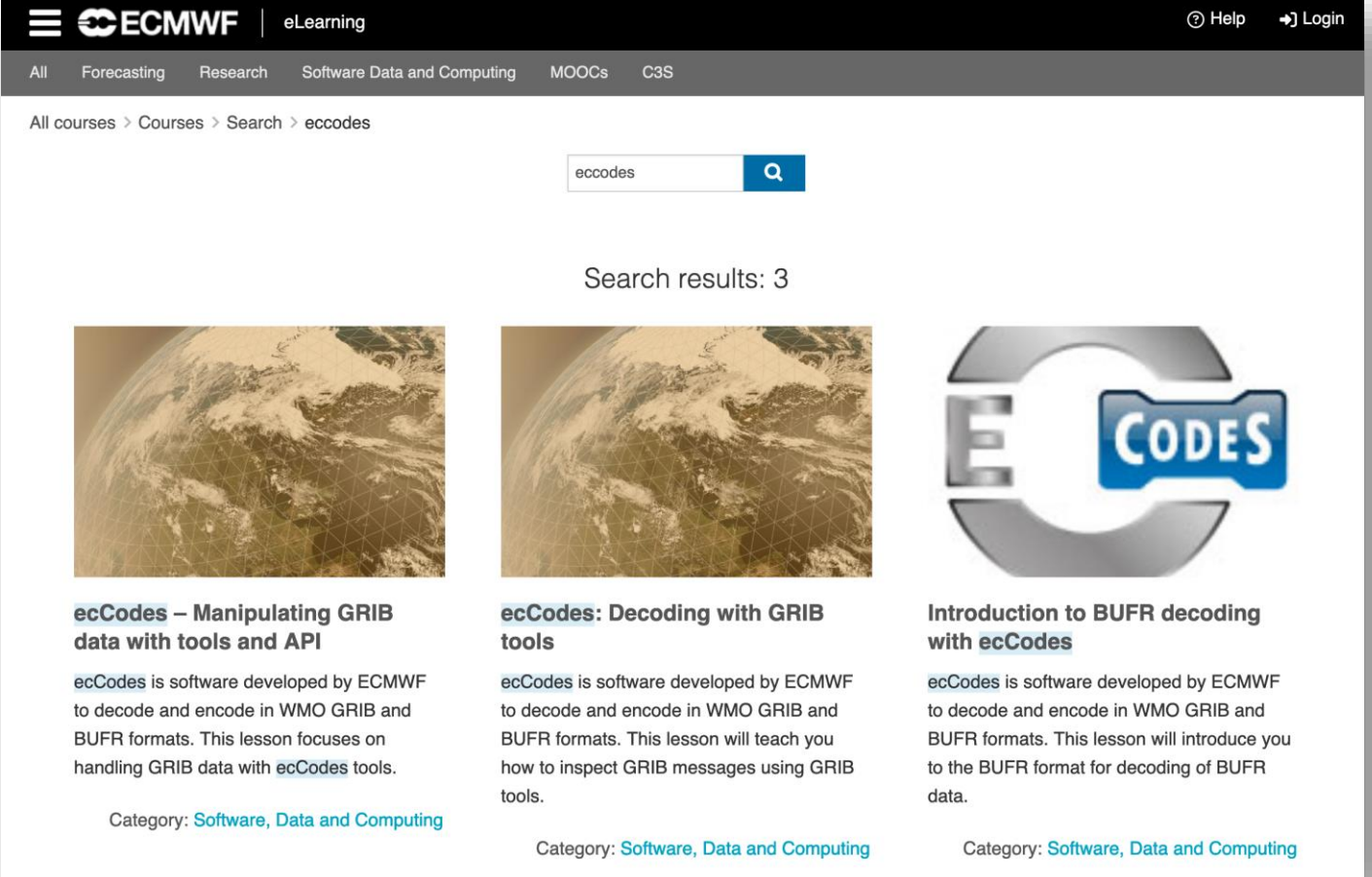
Python3 only



# ecCodes eLearning resources

<https://www.ecmwf.int/en/learning/education-material/elearning-online-resources>

- ecCodes: Decoding with GRIB tools
- ecCodes: Manipulating GRIB data with tools and API



The screenshot shows the ECMWF eLearning website interface. At the top, there is a navigation bar with the ECMWF logo and 'eLearning' text. Below this is a secondary navigation bar with categories: All, Forecasting, Research, Software Data and Computing, MOOCs, and C3S. The main content area shows search results for 'ecCodes'. A search bar at the top right contains the text 'ecCodes' and a magnifying glass icon. Below the search bar, it says 'Search results: 3'. There are three search results displayed as cards. Each card features a satellite image of Earth with a grid overlay. The first card is titled 'ecCodes – Manipulating GRIB data with tools and API' and includes a description: 'ecCodes is software developed by ECMWF to decode and encode in WMO GRIB and BUFR formats. This lesson focuses on handling GRIB data with ecCodes tools.' and a category link: 'Category: Software, Data and Computing'. The second card is titled 'ecCodes: Decoding with GRIB tools' and includes a description: 'ecCodes is software developed by ECMWF to decode and encode in WMO GRIB and BUFR formats. This lesson will teach you how to inspect GRIB messages using GRIB tools.' and a category link: 'Category: Software, Data and Computing'. The third card is titled 'Introduction to BUFR decoding with ecCodes' and includes a description: 'ecCodes is software developed by ECMWF to decode and encode in WMO GRIB and BUFR formats. This lesson will introduce you to the BUFR format for decoding of BUFR data.' and a category link: 'Category: Software, Data and Computing'. To the right of the search results is a large graphic of the 'ECODES' logo, which consists of a stylized 'E' and 'C' forming a circle, with the word 'CODES' in a blue box.



## ecCodes – jupyter notebooks

- Four notebooks showing how to use the ecCodes Python API are provided
  - [nwp\\_eccodes\\_api\\_1.ipynb](#) – First steps with the ecCodes Python API
  - [nwp\\_eccodes\\_api\\_2.ipynb](#) – Going further with the ecCodes Python API
  - [nwp\\_eccodes\\_api\\_2.ipynb](#) – Next steps with the ecCodes Python API
  - [nwp\\_eccodes\\_api\\_4.ipynb](#) – Handling files containing multiple GRIBs with the ecCodes Python API
  
- Five notebooks showing how to use the GRIB tools in the grib\_tools folder
  - [nwp\\_eccodes\\_grib\\_tools\\_intro.ipynb](#) – Introduction to using the GRIB tools
  - [nwp\\_eccodes\\_grib\\_tools\\_inspect.ipynb](#) – Using the GRIB tools to inspect GRIB messages
  - [nwp\\_eccodes\\_grib\\_tools\\_modify.ipynb](#) – Using the GRIB tools to modify GRIB messages
  - [nwp\\_eccodes\\_grib\\_tools\\_compare.ipynb](#) – Using the GRIB tools to compare GRIB messages
  - [nwp\\_eccodes\\_grib\\_tools\\_to\\_netcdf.ipynb](#) – Using the GRIB tools to convert GRIB messages to NetCDF





# Questions ?

