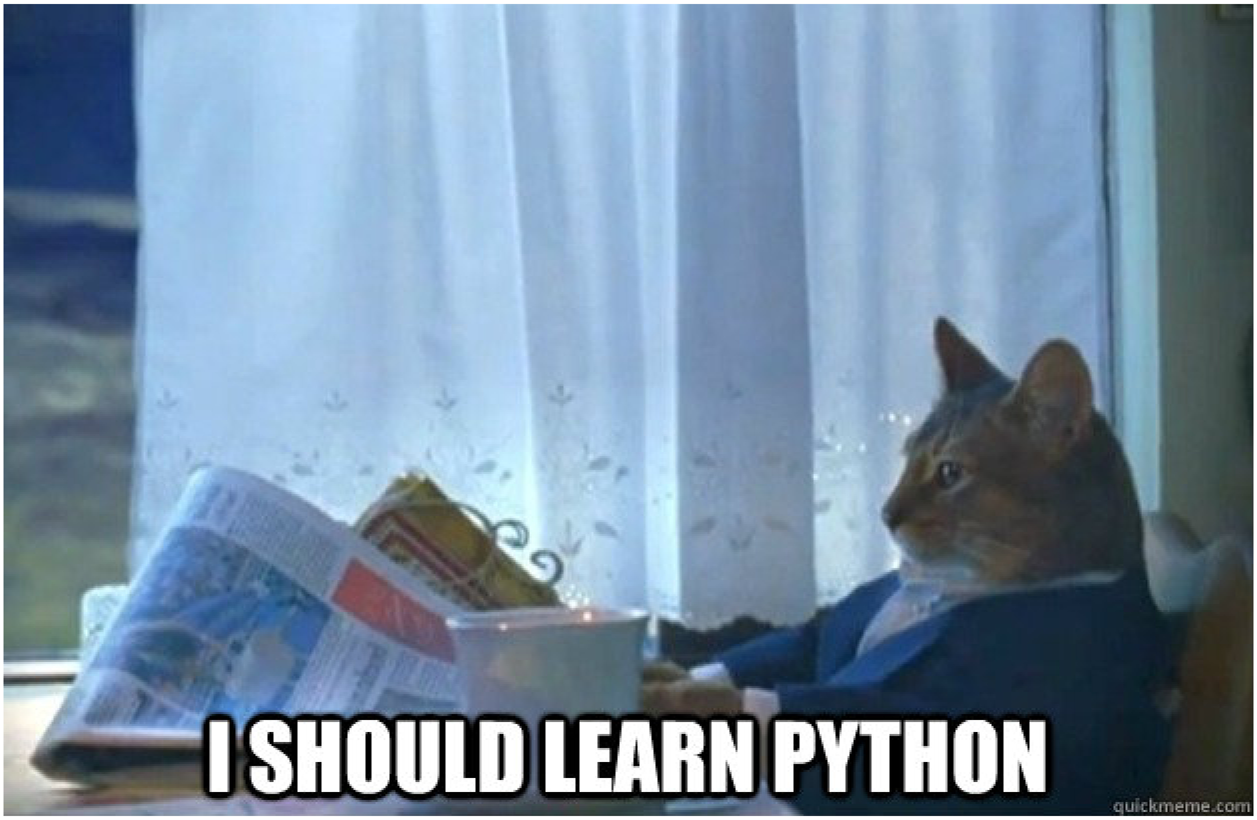
Python: A Simple Introduction



DOAN Quang-Van

計算科学研究センター

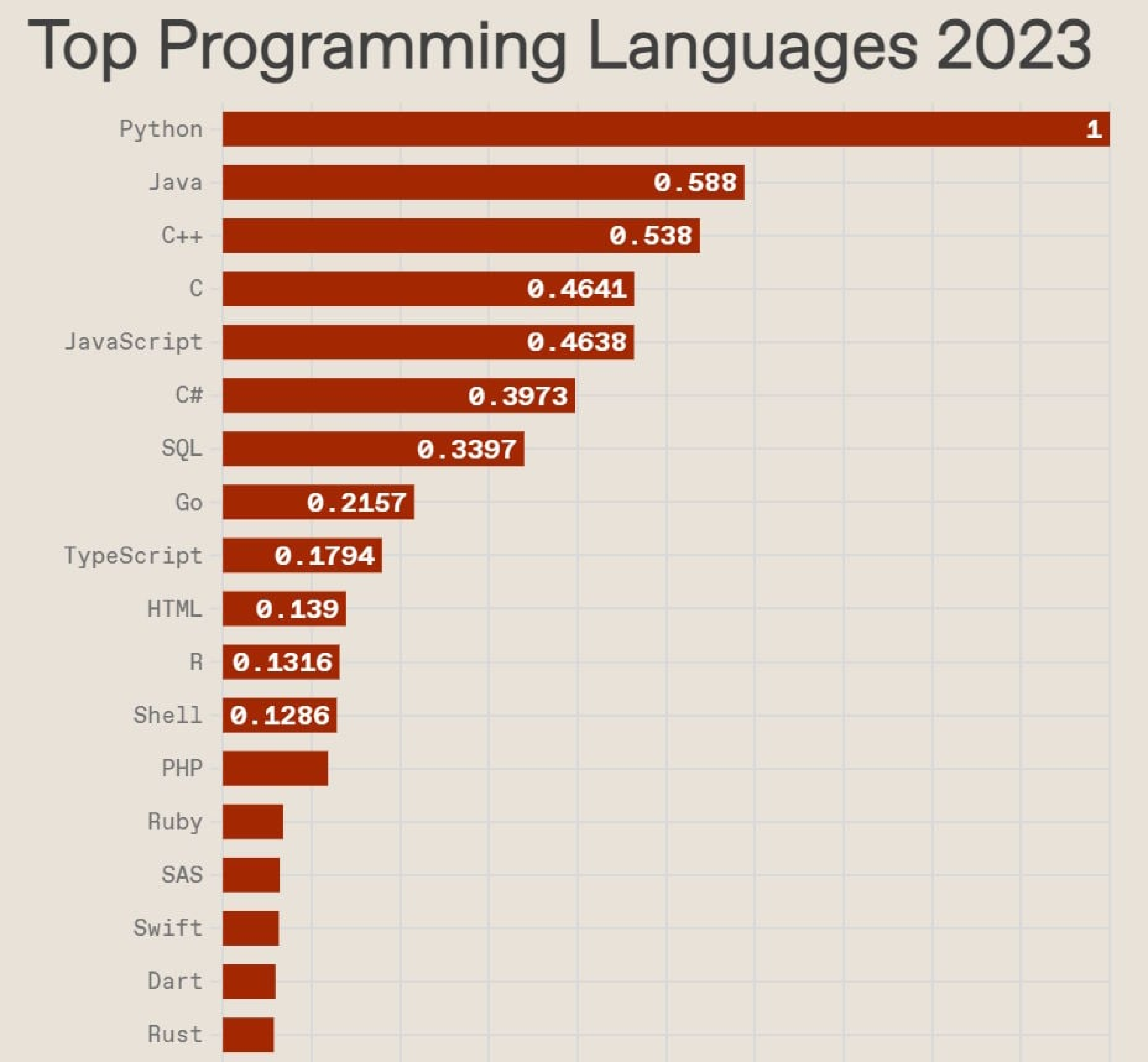
Email: Doan.van.gb@u.tsukuba.ac.jp

参考リンク 1) https://python.keicode.com/lang/control-basic-rule.php

2) https://camp.trainocate.co.jp/magazine/python-basic/

In [ ]:

What is Python



(Source: IEEE Spectrum)

オンラインPythonを試す場合： https://jupyter.org/try

Learn to install, write and run python code

What is python

Pythonの基本概念 Pythonは、オブジェクト指向（object-oriented）プログラミング言語です。 オブジェクト指向とは、値やそれを扱うためのソースコードをまとめて「オブジェクト（物）」として扱う考え方です。（Fortran言語なら構造体にfunctionやsubroutineがくっついたもの。）

How to install python

Good recommendation https://www.anaconda.com/download Using python on Anaconda platform

Basic libraries: numpy, pandas, xarray, rioxarray, cartopy, geopandas, ... Install libraries using conda

> conda install xarray -c conda-forge

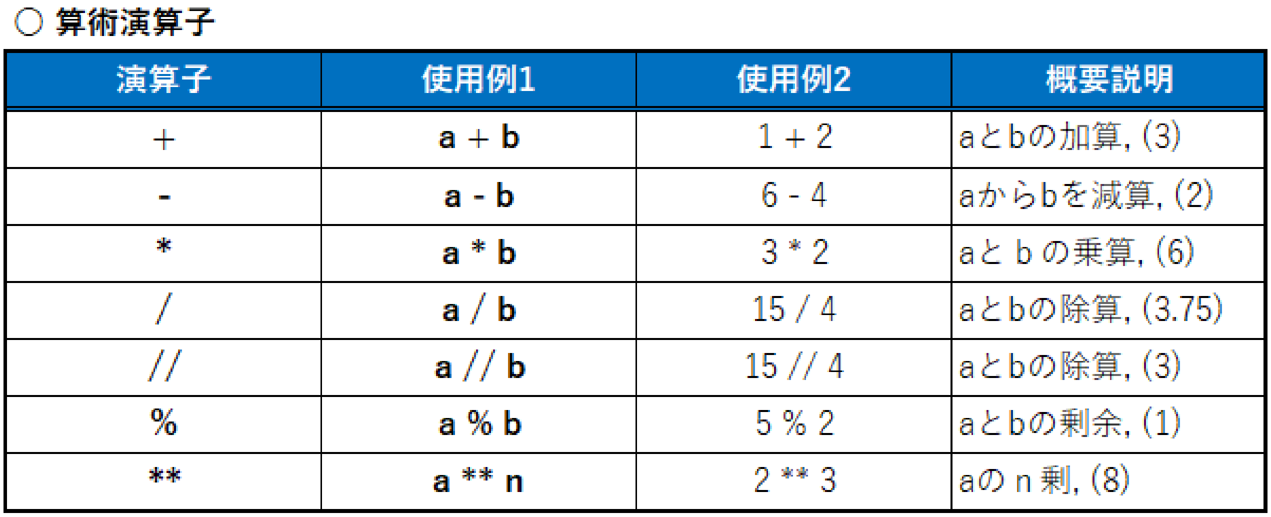
TRY to LEARN using GOOGLE

In [4]:

```
# スタート
print('Hello world!')

string='Hello world'
print(string)

print(string.lower())
print(string.upper())
#.lower(), upper()は string 'Hello world' の functionと言います。
```

```
Hello world!
Hello world
hello world
HELLO WORLD
```

演算



○ 算術演算子

演算子	使用例1	使用例2	概要説明
+	a + b	1 + 2	aとbの加算, (3)
-	a - b	6 - 4	aからbを減算, (2)
*	a * b	3 * 2	aと b の乗算, (6)
/	a / b	15 / 4	aとbの除算, (3.75)
//	a // b	15 // 4	aとbの除算, (3)
%	a % b	5 % 2	aとbの剰余, (1)
**	a ** n	2 ** 3	aの n 剰, (8)

source: https://www.shibutan-bloomers.com/python-basic-2/400/

In [56]:

```
# 基本演算

x = 1.0
y = 2.0

z = x**2 + 2.5*y**2

print(z)

z = x / y - 1.

print(z)
```

```
11.0
-0.5
```

python 配列

numpyはプログラミング言語であるPythonの拡張モジュールです。 numpyを利用することでPythonでの数値計算をより高速に、効率的に行うことができるようになります。

In [57]:

```
# python 配列
import numpy as np
```

```
# 定義し方
x = np.array([1,2,3])
print(x)

x = np.arange(-10,10,.5)
print(x)

y = x**2 + 2.*x  + 10

print(x)
print(y)
```

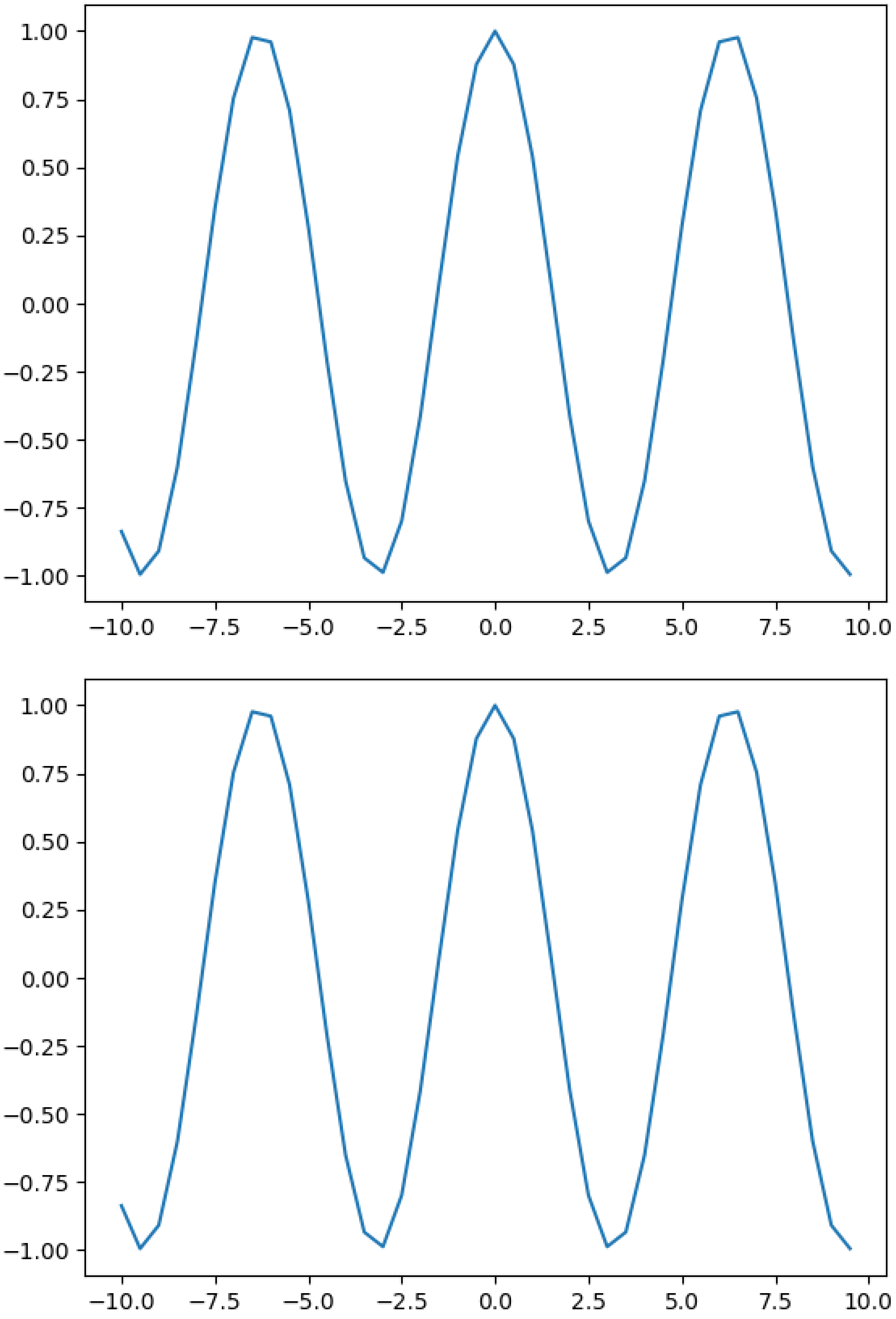
```
[1 2 3]
[-10.   -9.5  -9.   -8.5  -8.   -7.5  -7.   -6.5  -6.   -5.5  -5.   -4.5
  -4.   -3.5  -3.   -2.5  -2.   -1.5  -1.   -0.5   0.    0.5   1.    1.5
   2.    2.5   3.    3.5   4.    4.5   5.    5.5   6.    6.5   7.    7.5
   8.    8.5   9.    9.5]
[-10.   -9.5  -9.   -8.5  -8.   -7.5  -7.   -6.5  -6.   -5.5  -5.   -4.5
  -4.   -3.5  -3.   -2.5  -2.   -1.5  -1.   -0.5   0.    0.5   1.    1.5
   2.    2.5   3.    3.5   4.    4.5   5.    5.5   6.    6.5   7.    7.5
   8.    8.5   9.    9.5]
[ 90.    81.25  73.    65.25  58.    51.25  45.    39.25  34.    29.25
  25.    21.25  18.    15.25  13.    11.25  10.     9.25   9.     9.25
  10.    11.25  13.    15.25  18.    21.25  25.    29.25  34.    39.25
  45.    51.25  58.    65.25  73.    81.25  90.    99.25 109.   119.25]
```

簡単に作図してみる

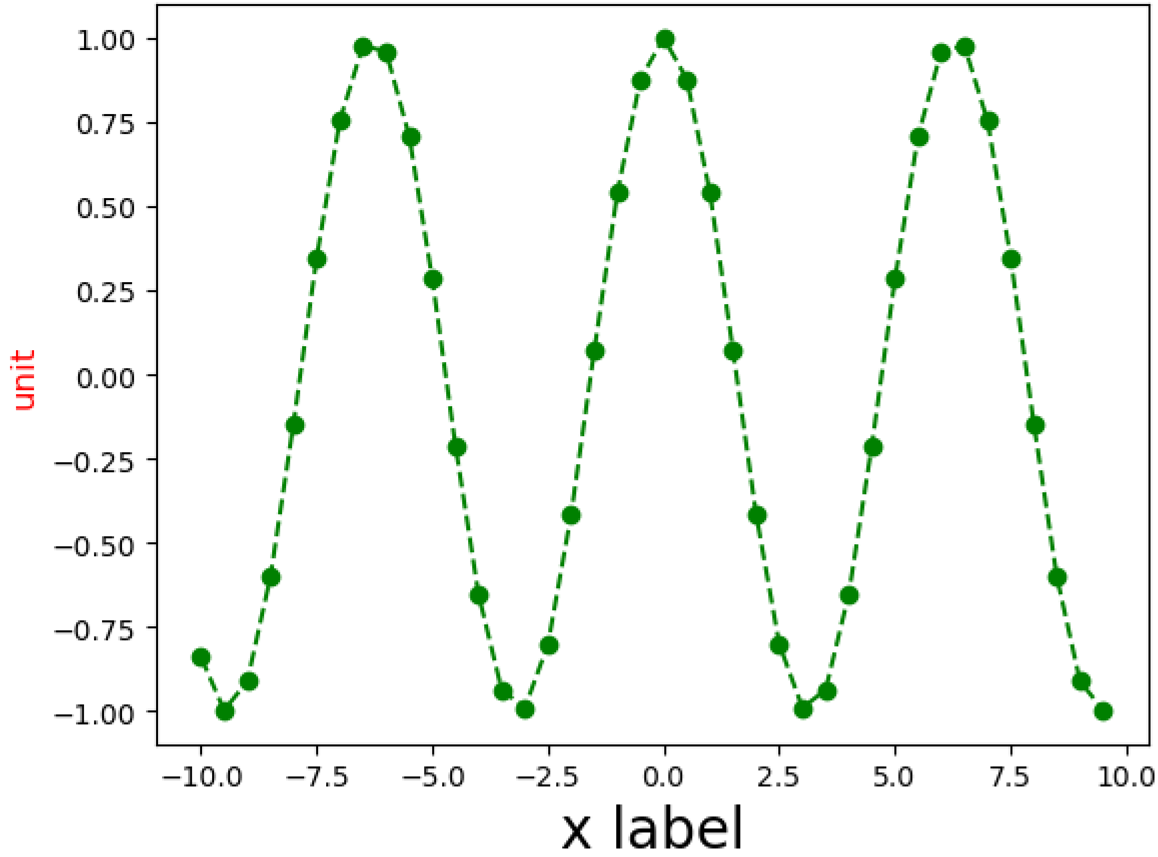
matplotlibとはPythonにおけるグラフ描画の標準的なライブラリです。 https://matplotlib.org

In [60]:

```
import matplotlib.pyplot as plt

plt.plot(x,y)
plt.show()

y = np.cos(x)

plt.plot(x,y)
plt.show()

plt.plot(x,y, ls='--',marker='o', color='g')
plt.xlabel('x label', fontsize=20)
plt.ylabel('unit', fontsize=12, color='r')
```



```
Out[60]: Text(0, 0.5, 'unit')
```

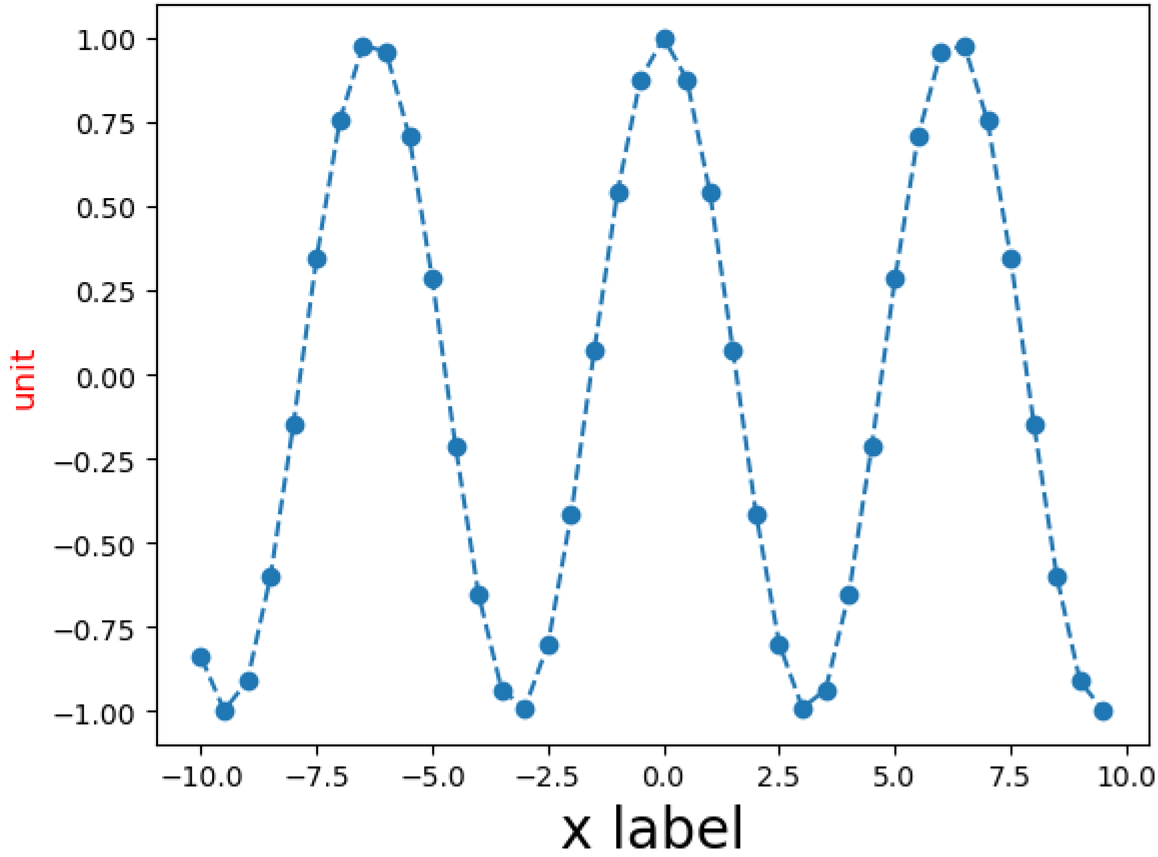


```
In [37]:
```

```
[-10.    -9.5   -9.    -8.5   -8.    -7.5   -7.    -6.5   -6.    -5.5   -5.    -4.5
  -4.    -3.5   -3.    -2.5   -2.    -1.5   -1.    -0.5    0.     0.5    1.     1.5
   2.     2.5    3.     3.5    4.     4.5    5.     5.5    6.     6.5    7.     7.5
   8.     8.5    9.     9.5] [-0.83907153 -0.99717216 -0.91113026 -0.6020119  -0.14550003
0.34663532
  0.75390225  0.97658763  0.96017029  0.70866977  0.28366219 -0.2107958
 -0.65364362 -0.93645669 -0.9899925  -0.80114362 -0.41614684  0.0707372
  0.54030231  0.87758256  1.          0.87758256  0.54030231  0.0707372
 -0.41614684 -0.80114362 -0.9899925  -0.93645669 -0.65364362 -0.2107958
  0.28366219  0.70866977  0.96017029  0.97658763  0.75390225  0.34663532
 -0.14550003 -0.6020119  -0.91113026 -0.99717216]
```

Out[37]:

```
Text(0, 0.5, 'unit')
```



分岐・繰り返し処理

In [61]:

```
#分岐
a = 1
if a > 1:
    print('a > 1')
else:
    print('a <= 1')


#繰り返し処理
for b in [1,2,3]:
    print(b)
```

```
a <= 1
1
2
3
```

CSVデータの読み書き方法・プロット方法

In [4]:

```
# read csv file (AMeDAS data Tokyo station)
import pandas as pd
#
df = pd.read_csv('data/2023_10_23.csv')

print(df.head())
```

```
            Unnamed: 0  glp_hPa  slp_hPa  precip_mm  temp_C  dewtemp_C  \
0  2023-10-23 01:00:00   1018.6   1022.0        NaN     9.4        8.8   
1  2023-10-23 02:00:00   1018.5   1021.9        NaN     8.6        8.3   
2  2023-10-23 03:00:00   1018.5   1021.9        NaN     7.8        7.7   
3  2023-10-23 04:00:00   1018.8   1022.2        NaN     7.7        7.6   
```

```
4  2023-10-23 05:00:00   1019.1   1022.5        NaN     7.8        7.2

   vapor-pres_hPa  rh_percent  wspd_ms  wdir_deg  sunlit_h  rad-global_MJm-2  \
0            11.3          96      0.6     292.5       NaN               NaN
1            11.0          98      1.3     270.0       NaN               NaN
2            10.5          99      1.3     292.5       NaN               NaN
3            10.4          99      1.7     292.5       NaN               NaN
4            10.2          96      1.6     315.0       NaN               NaN

   snowfall_cm  snowdepth_cm  weather_typ  cloudcover_x  visibility_km
0         -999          -999          NaN           NaN           20.0
1         -999          -999          NaN           NaN           20.0
2         -999          -999          NaN           NaN           16.3
3         -999          -999          NaN           NaN           17.9
4         -999          -999          NaN           NaN           20.0
```

In [5]:

```
df = pd.read_csv('data/2023_10_23.csv', index_col=0, parse_dates=True)
print(df.head())
```

```
                     glp_hPa  slp_hPa  precip_mm  temp_C  dewtemp_C  \
2023-10-23 01:00:00   1018.6   1022.0        NaN     9.4        8.8
2023-10-23 02:00:00   1018.5   1021.9        NaN     8.6        8.3
2023-10-23 03:00:00   1018.5   1021.9        NaN     7.8        7.7
2023-10-23 04:00:00   1018.8   1022.2        NaN     7.7        7.6
2023-10-23 05:00:00   1019.1   1022.5        NaN     7.8        7.2

                     vapor-pres_hPa  rh_percent  wspd_ms  wdir_deg  sunlit_h  \
2023-10-23 01:00:00            11.3          96      0.6     292.5       NaN
2023-10-23 02:00:00            11.0          98      1.3     270.0       NaN
2023-10-23 03:00:00            10.5          99      1.3     292.5       NaN
2023-10-23 04:00:00            10.4          99      1.7     292.5       NaN
2023-10-23 05:00:00            10.2          96      1.6     315.0       NaN

                     rad-global_MJm-2  snowfall_cm  snowdepth_cm  weather_typ  \
2023-10-23 01:00:00               NaN         -999          -999          NaN
2023-10-23 02:00:00               NaN         -999          -999          NaN
2023-10-23 03:00:00               NaN         -999          -999          NaN
2023-10-23 04:00:00               NaN         -999          -999          NaN
2023-10-23 05:00:00               NaN         -999          -999          NaN

                     cloudcover_x  visibility_km
2023-10-23 01:00:00           NaN           20.0
2023-10-23 02:00:00           NaN           20.0
2023-10-23 03:00:00           NaN           16.3
2023-10-23 04:00:00           NaN           17.9
2023-10-23 05:00:00           NaN           20.0
```

In [6]:

```
print(df.columns)
```

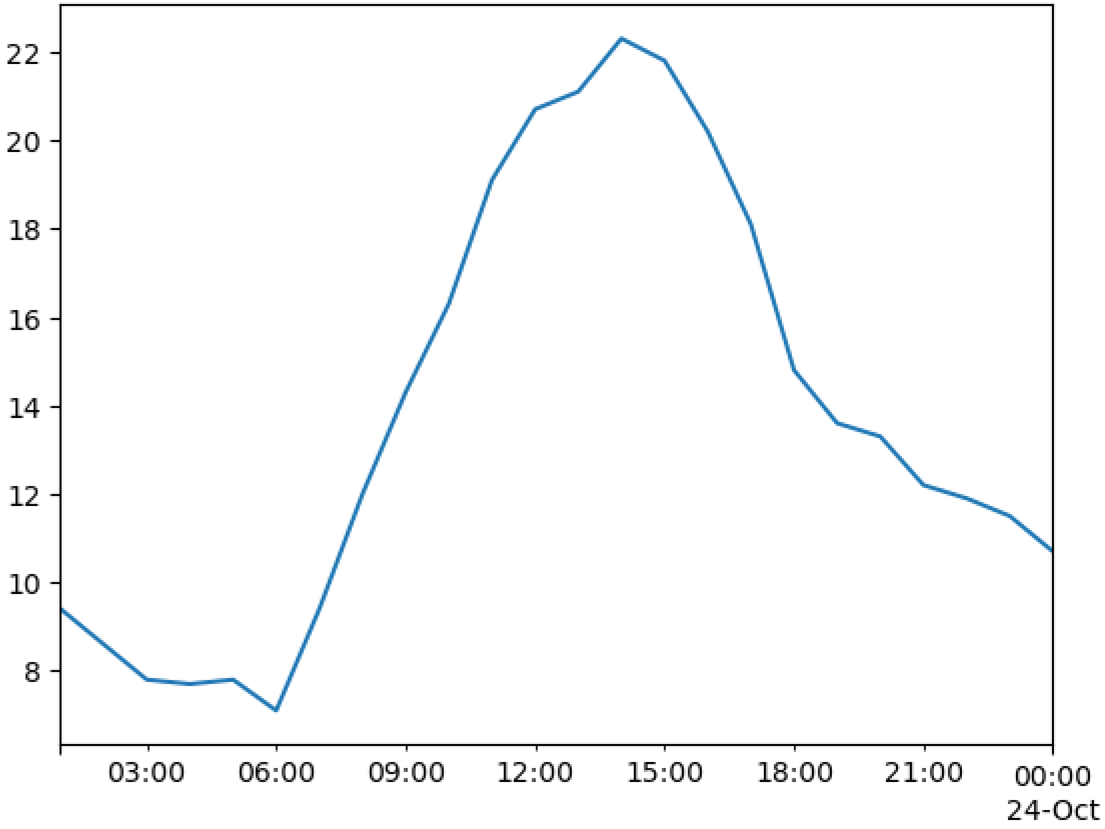
```
Index(['glp_hPa', 'slp_hPa', 'precip_mm', 'temp_C', 'dewtemp_C',
       'vapor-pres_hPa', 'rh_percent', 'wspd_ms', 'wdir_deg', 'sunlit_h',
       'rad-global_MJm-2', 'snowfall_cm', 'snowdepth_cm', 'weather_typ',
       'cloudcover_x', 'visibility_km'],
      dtype='object')
```

In [7]:

```
df['temp_C'].plot()
plt.ylabel('Temp')
plt.legend(['Temp'])
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[7], line 2
      1 df['temp_C'].plot()
----> 2 plt.ylabel('Temp')
      3 plt.legend(['Temp'])
```

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



NetCDFデータの読み込みと作図方法 (Optional)

In [8]:

```
import xarray as xr
```

In [11]:

```
ds = xr.open_dataset('data/geo_em.d03.nc')
```

In [10]:

```
ds
```

Out[10]:

xarray.Dataset

▶ Dimensions: (Time: 1, south_north: 99, west_east: 99, south_north_stag: 100, west_east_stag: 100, land_cat: 24, soil_cat: 16, month: 12)

▶ Coordinates: (0)

▶ Data variables: (50)

▶ Indexes: (0)

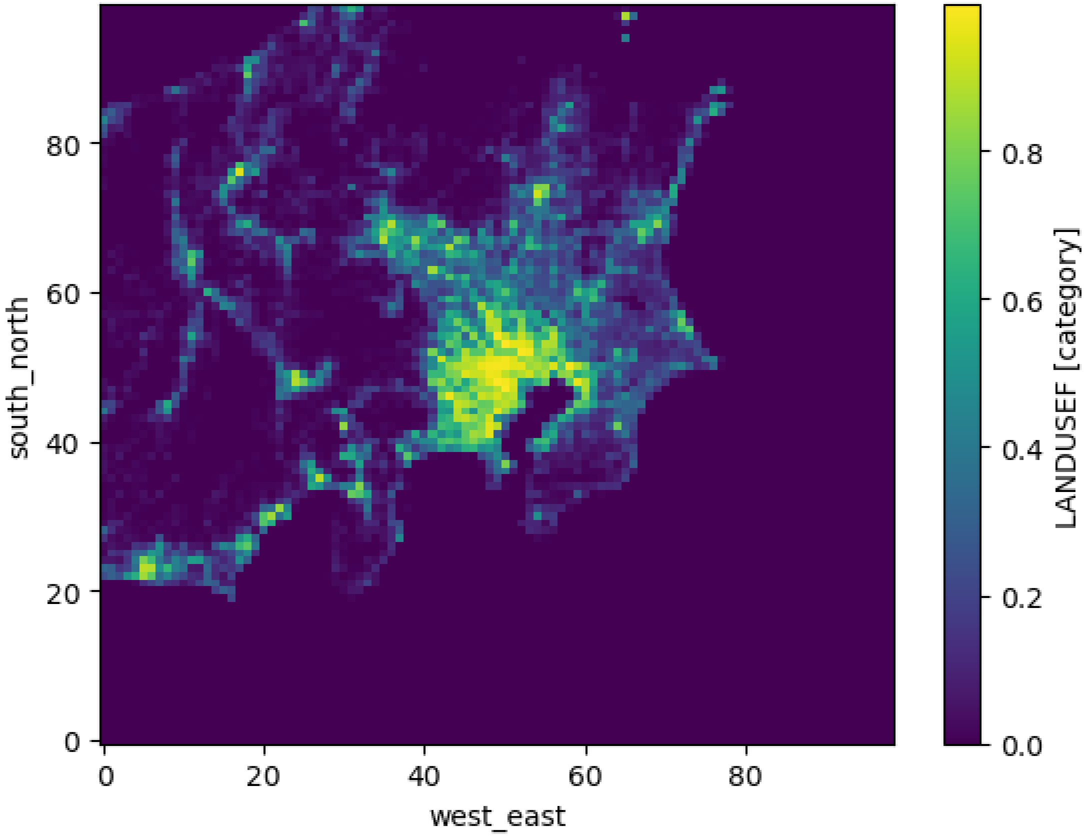
▶ Attributes: (45)

In [12]:

```
ds.LANDUSEF[0,0].plot()
```

Out[12]:

```
<matplotlib.collections.QuadMesh at 0x1598dec90>
```



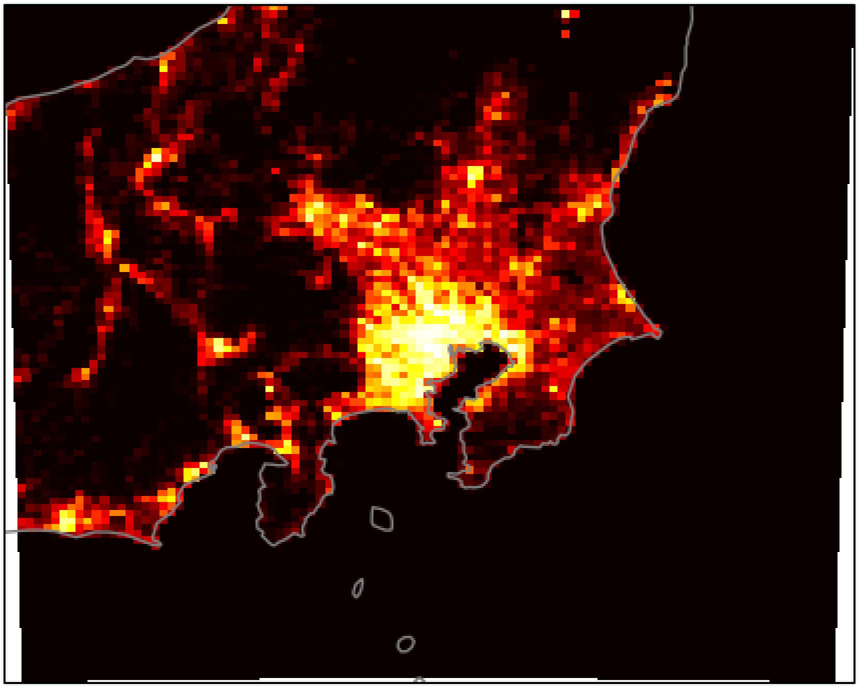
In [14]:

```
import matplotlib.pyplot as plt
import cartopy.crs as ccrs
fig   =  plt.figure(figsize= (5,5))
ax  = plt.axes([0.1,0.1,0.85,0.85], projection=ccrs.PlateCarree())
ax.coastlines('10m', linewidth=1.,color='gray')

lon, lat = ds.XLONG_M[0].values, ds.XLAT_M[0].values
z = ds.LANDUSEF[0,0].values

plt.pcolormesh(lon, lat, z,transform=ccrs.PlateCarree(), cmap=plt.get_cmap('hot'))
```

```
/Users/doan/anaconda3/lib/python3.11/site-packages/cartopy/mpl/geoaxes.py:1781: UserWarn
ing: The input coordinates to pcolormesh are interpreted as cell centers, but are not mo
notonically increasing or decreasing. This may lead to incorrectly calculated cell edge
s, in which case, please supply explicit cell edges to pcolormesh.
  result = super().pcolormesh(*args, **kwargs)
```

Out[14]:

```
<cartopy.mpl.geocollection.GeoQuadMesh at 0x15a022a90>
```



```
In [ ]:
```

```
In [ ]:
```