地図描画

from ipyleaflet import Map

m=Map(center=(35.68,139.75),zoom=12)

m

マーカー描画

from ipyleaflet import Map, Marker

(lat,lon)=(35.682,139.765)

m=Map(center=(lat,lon),zoom=17)

for n in range(-3,4,1):

 mark= Marker(location=(lat,lon +n/2000))

 m +=mark

m

マーカークラスター

from ipyleaflet import Map,Marker

from ipyleaflet import Map,MarkerCluster

from ipyleaflet import Map

(lat,lon)=(35.682,139.765)

mrks=[ ]

m=Map(center=(lat,lon),zoom=17)

for n in range(-3,4,1):

 mrks.append(Marker(location=(lat,lon +n/1000)))

marker\_cluster=MarkerCluster(markers=mrks)

m +=marker\_cluster

m

直線描画

from ipyleaflet import Map, Polyline

(lat,lon)=(35.682,139.765)

m=Map(center=(lat,lon),zoom=15)

for n in range(0,10):

 pl=Polyline(locations=[(lat+0.005,lon-0.005),

 (lat+0.005-0.001\*n,lon+0.005)],

 weight=10-n,color='#f00')

 m +=pl

m

画像挿入

from ipyleaflet import Map, ImageOverlay

(lat,lon)=(35.682,139.765)

m=Map(center=(lat,lon),zoom=15)

io=ImageOverlay(url='富士山.jpg',

 bounds=[(lat-0.005,lon-0.005),(lat+0.005,lon+0.005)])

m +=io

m

描画コントロール

from ipyleaflet import Map,DrawControl

(lat,lon)=(35.682,139.765)

m=Map(center=(lat,lon),zoom=17)

dc=DrawControl()

m.add\_control(dc)

m

マーカー表示

from ipyleaflet import Map,Marker

def event\_handle(\*\*kwargs):

 if kwargs['type']=='click':

 global m

 mk=Marker(location=kwargs['coordinates'])

 m +=mk

(lat,lon)=(35.682,139.765)

m=Map(center=(lat,lon),zoom=17)

m.on\_interaction(event\_handle)

m

リアルタイム表示

from ipyleaflet import Map,Marker

from ipywidgets import Label

label1=Label()

label2=Label()

display(label1,label2)

def event\_handle(\*\*kwargs):

 if kwargs['type']=='mousemove':

 here=kwargs['coordinates']

 label1.value='Latitude:'+str(here[0])

 label2.value='Longitude:'+str(here[1])

(lat,lon)=(35.682,139.765)

m=Map(center=(lat,lon),zoom=17)

m.on\_interaction(event\_handle)

m

ボタンとイベント

from sympy import \*

from ipywidgets import Label,IntText,Button

label=Label()

input=IntText()

label.value='please slide'

x=Symbol('x')

re=x\*\*2-x\*3

def handle\_event(target):

 global re,label,input

 label.value='result:'+str(re.subs(x,input.value))

button=Button(description='click')

button.on\_click(handle\_event)

display(label,input,button)button.on\_click(handle\_event)

display(label,input,button)

真偽チェック

from ipywidgets import \*

label=Label(value='slide me!')

check=Checkbox(description='check now')

toggle=ToggleButton(description='Toggle')

display(label,check,toggle)

def on\_change(change):

 global check,toggle

 label.value='check:' +str(check.value)+' ,toggle:'+str(toggle.value)

check.observe(on\_change)

toggle.observe(on\_change)

アコーディオン

from ipywidgets import \*

btnsH=[]

for n in range(0,5):

 btnsH.append(Button(description='First ' + str(n)))

box1=VBox(btnsH)

btnsV=[]

for n in range(0,5):

 btnsV.append(Button(description='Second ' + str(n)))

box2=VBox(btnsV)

accordion=Accordion(children=[box1,box2])

accordion.set\_title(0,'First ')

accordion.set\_title(1,'Second ')

display(accordion)

タブ

from ipywidgets import Tab

btnsH=[]

for n in range(0,5):

 btnsH.append(Button(description='First ' + str(n)))

box1=VBox(btnsH)

btnsV=[]

for n in range(0,5):

 btnsV.append(Button(description='Second ' + str(n)))

box2=VBox(btnsV)

tab=Tab(children=[box1,box2])

tab.set\_title(0,'First ')

tab.set\_title(1,'Second ')

display(tab)