1_SVG_converter_Advanced

January 16, 2021

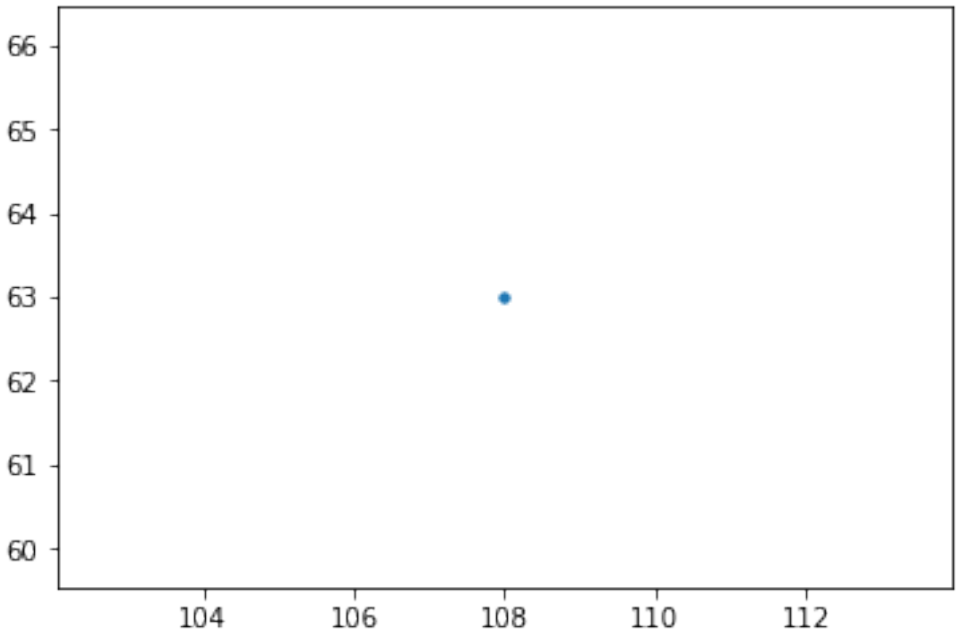
In [27]:
```
import numpy as np
import matplotlib.pyplot as plt

from svg.path import parse_path
from svg.path.path import Line
from xml.dom import minidom

def line_splitter(start, end):
    return (lambda t: (1-t)*start+t*end)

def cubic_bezier_converter(start, control1, control2, end):
    original_data = np.array([start, control1, control2, end])
    cubic_bezier_matrix = np.array([
        [-1,  3, -3,  1],
        [ 3, -6,  3,  0],
        [-3,  3,  0,  0],
        [ 1,  0,  0,  0]
    ])
    return_data = cubic_bezier_matrix.dot(original_data)

    return (lambda t: np.array([t**3, t**2, t, 1]).dot(return_data))

# Learned from
# https://stackoverflow.com/questions/36971363/how-to-interpolate-svg-path-into-a-pix
```

In [4]:
```
doc = minidom.parse('B_sample.svg')
path_strings = [path.getAttribute('d') for path
                in doc.getElementsByTagName('path')]
doc.unlink()

for path_string in path_strings:
    path = parse_path(path_string)
    for e in path:
        if type(e).__name__ == 'Line':
            x0 = e.start.real
            y0 = e.start.imag
            x1 = e.end.real
```

```
                    y1 = e.end.imag
                    print("(%.2f, %.2f) - (%.2f, %.2f)" % (x0, y0, x1, y1))
```

In [59]:
```
block=0
n_dots=100
key=0
points_np=[]

path=parse_path(path_strings[block])

dat=path[key]
if type(path[key]).__name__=='CubicBezier':
    start_np = np.array([dat.start.real, dat.start.imag])
    control1_np = np.array([dat.control1.real, dat.control1.imag])
    control2_np = np.array([dat.control2.real, dat.control2.imag])
    end_np = np.array([dat.end.real, dat.end.imag])
    converted_curve = cubic_bezier_converter(start_np, control1_np, control2_np, end_
    #
    diff_np=start_np-end_np
    n_dots=np.round(np.linalg.norm(diff_np))
    #
    points_np = np.array([converted_curve(t) for t in np.linspace(0, 1, n_dots)])
elif type(path[key]).__name__=='Line':
    start_np = np.array([dat.start.real, dat.start.imag])
    end_np = np.array([dat.end.real, dat.end.imag])
    converted_line = line_splitter(start_np,end_np)
    #
    diff_np=start_np-end_np
    n_dots=np.round(np.linalg.norm(diff_np))
    #
    points_np=np.array([converted_line(t) for t in np.linspace(0, 1, n_dots)])
elif type(path[key]).__name__=='Move':
    #
    n_dots=1
    #
    start_np = np.array([dat.start.real, dat.start.imag])
    end_np = np.array([dat.end.real, dat.end.imag])
    points_np = np.array([start_np,end_np])
else:
    points_np=np.array([])

# == plot the line==
## controls_np = np.array([start_np, control1_np, control2_np, end_np])
#  curve segmentation

plt.plot(points_np[:, 0], points_np[:, 1], '.-')
# showing of control points
## plt.plot(controls_np[:,0], controls_np[:,1], 'o')
```

```
        #  line drawing
        ## plt.plot([start_np[0], control1_np[0]], [start_np[1], control1_np[1]], '-', lw=1)
        ## plt.plot([control2_np[0], end_np[0]], [control2_np[1], end_np[1]], '-', lw=1)

        plt.show()
        print(points_np)
```



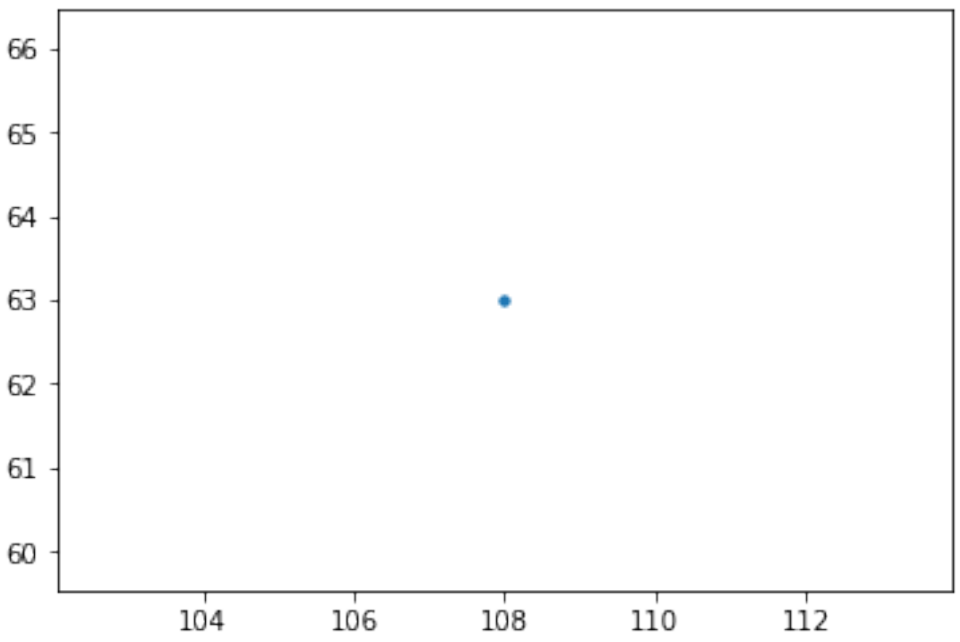

```
[[108.  63.]
 [108.  63.]]
```

```
In [231]: block=0
          n_dots=100
          key=0

          points_np_all=[]
          points_np_all=np.empty((len(path_strings)),dtype=object)
          print(len(points_np_all))
          #points_np_all[k]=np.array([])

          for k in range(len(path_strings)):
          #for path_string in path_strings:
              path = parse_path(path_strings[k])
              points_np_merge=np.empty((0,2), float)
              #points_np_merge=np.empty(points_np_merge)
              for dat in path:
```

```
#path=parse_path(path_strings[block])

#dat=path[key]

        if type(dat).__name__=='CubicBezier':
            start_np = np.array([dat.start.real, dat.start.imag])
            control1_np = np.array([dat.control1.real, dat.control1.imag])
            control2_np = np.array([dat.control2.real, dat.control2.imag])
            end_np = np.array([dat.end.real, dat.end.imag])
            converted_curve = cubic_bezier_converter(start_np, control1_np, control2
            #
            diff_np=start_np-end_np
            n_dots=np.round(np.linalg.norm(diff_np))
            #
            points_np = np.array([converted_curve(t) for t in np.linspace(0, 1, n_dot
        elif type(dat).__name__=='Line':
            start_np = np.array([dat.start.real, dat.start.imag])
            end_np = np.array([dat.end.real, dat.end.imag])
            converted_line = line_splitter(start_np,end_np)
            #
            diff_np=start_np-end_np
            n_dots=np.round(np.linalg.norm(diff_np))
            #
            points_np=np.array([converted_line(t) for t in np.linspace(0, 1, n_dots)]
        elif type(dat).__name__=='Move':
            #
            n_dots=1
            #
            start_np = np.array([dat.start.real, dat.start.imag])
            end_np = np.array([dat.end.real, dat.end.imag])
            points_np = np.array([start_np,end_np])
        else:
            points_np=np.array([])
        #points_np_merge=np.concatenate(points_np_merge,points_np)
        points_np_merge=np.append(points_np_merge, points_np, axis=0)
#         if k==0:
#             points_np_merge=points_np
#         else:
#             points_np_merge=np.append(points_np_merge,points_np,axis=0)
        plt.plot(points_np[:, 0], points_np[:, 1], '.-')
        plt.show()
        print(len(points_np))
        print(len(points_np_merge))
    #points_np_all1=points_np_all1.append(points_np_merge)
    #points_np_all=points_np_merge
    points_np_all[k]= points_np
#     points_np_all=points_np_all.append(points_np_merge)
```

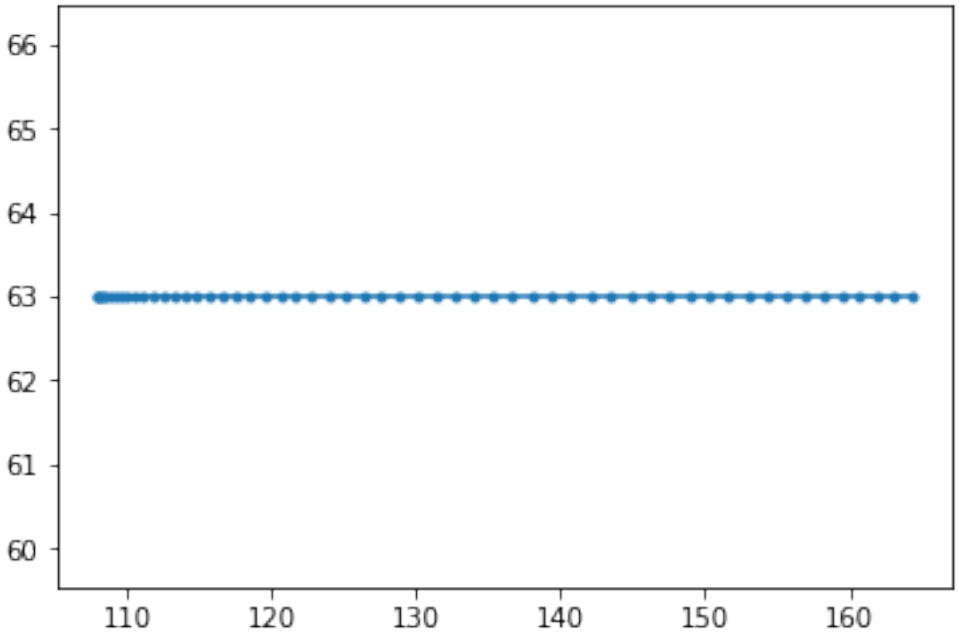
```
        print(len(points_np_all))
        plt.plot(points_np_merge[:, 0], points_np_merge[:, 1], '.-')
        plt.show()

    # == plot the line==
    ## controls_np = np.array([start_np, control1_np, control2_np, end_np])
    #  curve segmentation

    #for points_np in points_np_all:
    ## plt.plot(points_np[:, 0], points_np[:, 1], '.-')
    # showing of control points
    ## plt.plot(controls_np[:,0], controls_np[:,1], 'o')
    #  line drawing
    ## plt.plot([start_np[0], control1_np[0]], [start_np[1], control1_np[1]], '-', lw=1)
    ## plt.plot([control2_np[0], end_np[0]], [control2_np[1], end_np[1]], '-', lw=1)

    ##plt.show()
    #print(points_np)
    points_np_all
```

1

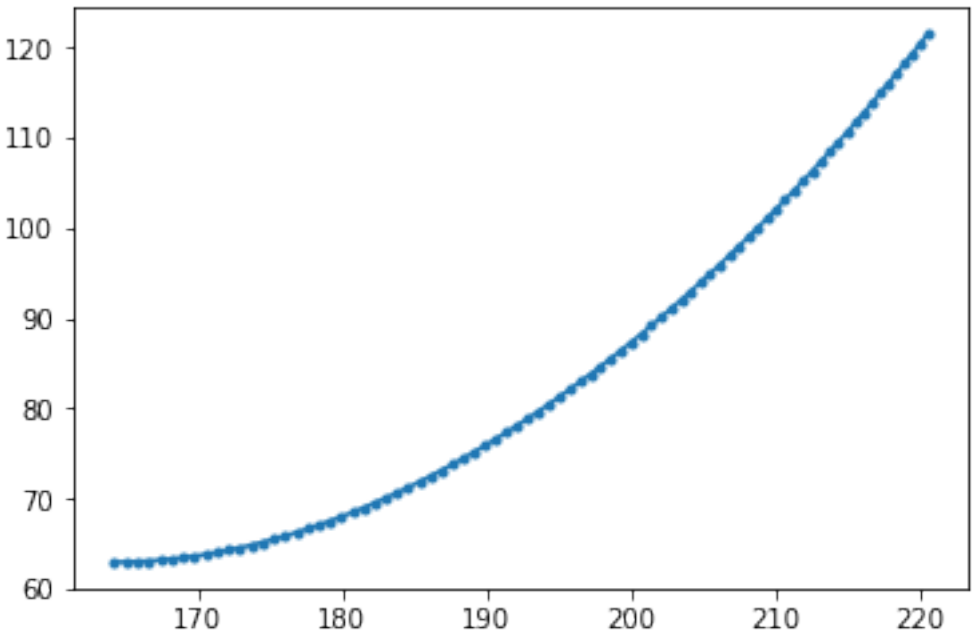


2

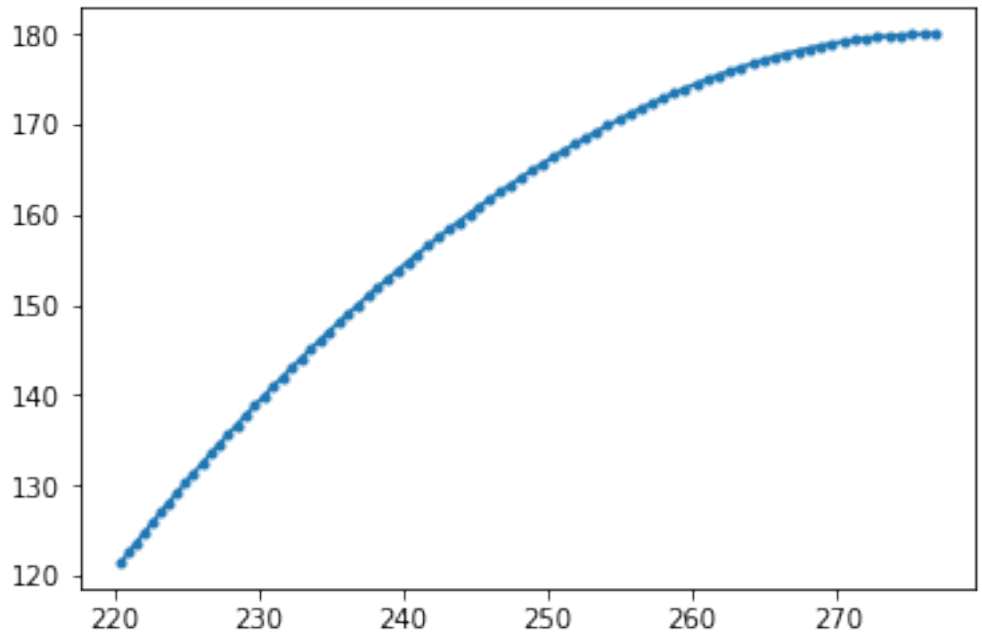
2


66
65
64
63
62
61
60
110
120
130
140
150
160

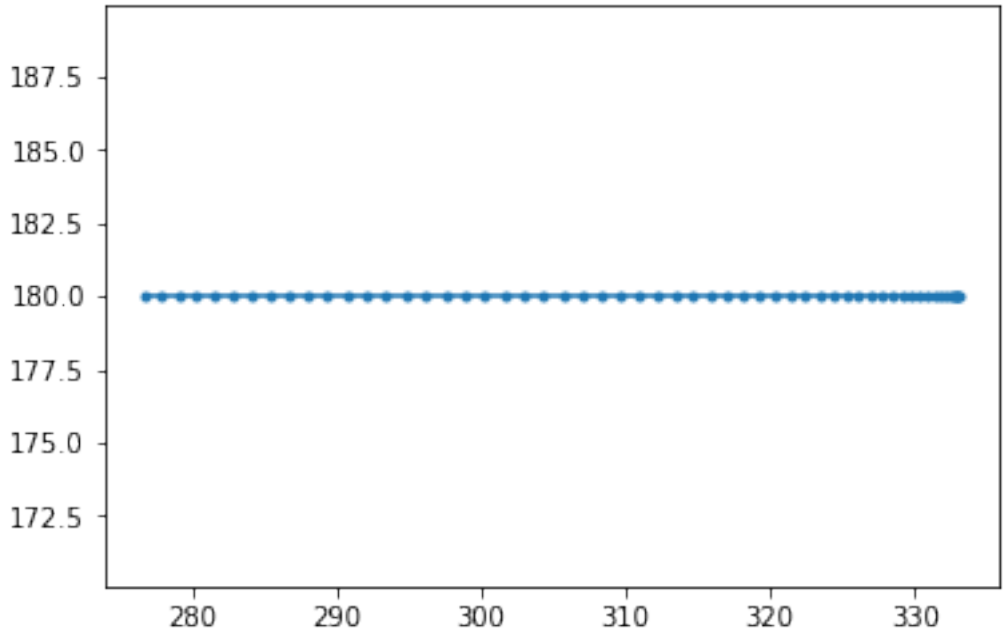
56
58


120
110
100
90
80
70
60
170
180
190
200
210
220

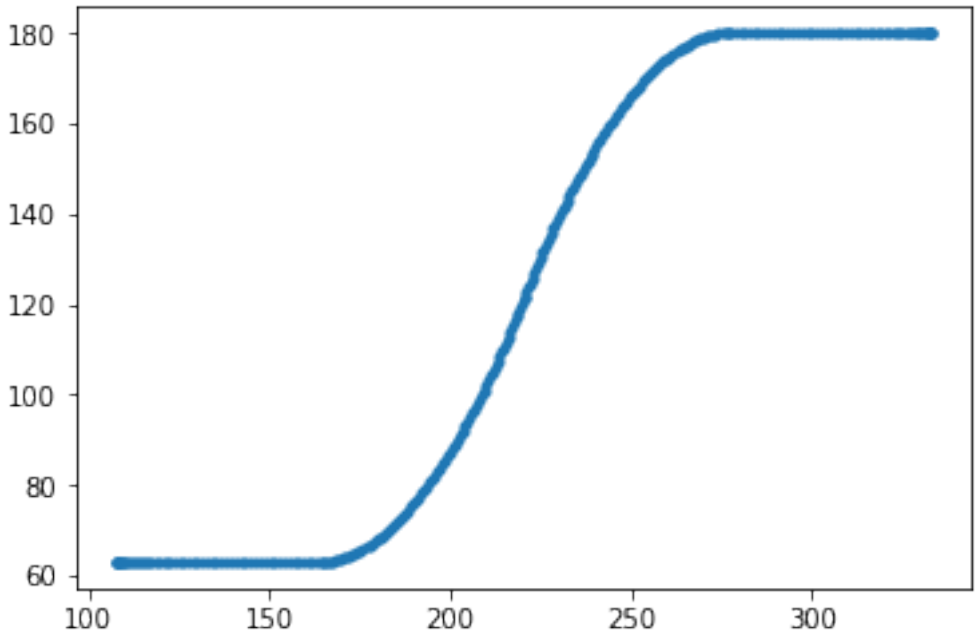
81
139


180
170
160
150
140
130
120
220
230
240
250
260
270

81
220


187.5
185.0
182.5
180.0
177.5
175.0
172.5
280
290
300
310
320
330

56
276
1




```
Out[231]: array([array([[276.75      , 180.        ],
                        [277.91423178, 180.        ],
                        [279.10458017, 180.        ],
                        [280.31927023, 180.        ],
                        [281.55652699, 180.        ],
                        [282.81457551, 180.        ],
                        [284.09164082, 180.        ],
                        [285.38594798, 180.        ],
                        [286.69572204, 180.        ],
                        [288.01918803, 180.        ],
                        [289.354571  , 180.        ],
                        [290.700096  , 180.        ],
                        [292.05398808, 180.        ],
                        [293.41447227, 180.        ],
                        [294.77977363, 180.        ],
                        [296.14811721, 180.        ],
                        [297.51772804, 180.        ],
                        [298.88683117, 180.        ],
```

```
                [300.25365165, 180.        ],
                [301.61641453, 180.        ],
                [302.97334485, 180.        ],
                [304.32266766, 180.        ],
                [305.662608  , 180.        ],
                [306.99139092, 180.        ],
                [308.30724146, 180.        ],
                [309.60838467, 180.        ],
                [310.8930456 , 180.        ],
                [312.15944929, 180.        ],
                [313.40582079, 180.        ],
                [314.63038514, 180.        ],
                [315.83136739, 180.        ],
                [317.00699259, 180.        ],
                [318.15548578, 180.        ],
                [319.275072  , 180.        ],
                [320.36397631, 180.        ],
                [321.42042374, 180.        ],
                [322.44263935, 180.        ],
                [323.42884818, 180.        ],
                [324.37727528, 180.        ],
                [325.28614568, 180.        ],
                [326.15368445, 180.        ],
                [326.97811662, 180.        ],
                [327.75766723, 180.        ],
                [328.49056135, 180.        ],
                [329.175024  , 180.        ],
                [329.80928024, 180.        ],
                [330.39155511, 180.        ],
                [330.92007366, 180.        ],
                [331.39306094, 180.        ],
                [331.80874199, 180.        ],
                [332.16534185, 180.        ],
                [332.46108557, 180.        ],
                [332.6941982 , 180.        ],
                [332.86290479, 180.        ],
                [332.96543037, 180.        ],
                [333.        , 180.        ]])], dtype=object)

In [233]: len(points_np_all)
          points_np_all[0]

Out[233]: array([[276.75      , 180.        ],
                 [277.91423178, 180.        ],
                 [279.10458017, 180.        ],
                 [280.31927023, 180.        ],
                 [281.55652699, 180.        ],
                 [282.81457551, 180.        ],
```

```
[284.09164082, 180.        ],
[285.38594798, 180.        ],
[286.69572204, 180.        ],
[288.01918803, 180.        ],
[289.354571  , 180.        ],
[290.700096  , 180.        ],
[292.05398808, 180.        ],
[293.41447227, 180.        ],
[294.77977363, 180.        ],
[296.14811721, 180.        ],
[297.51772804, 180.        ],
[298.88683117, 180.        ],
[300.25365165, 180.        ],
[301.61641453, 180.        ],
[302.97334485, 180.        ],
[304.32266766, 180.        ],
[305.662608  , 180.        ],
[306.99139092, 180.        ],
[308.30724146, 180.        ],
[309.60838467, 180.        ],
[310.8930456 , 180.        ],
[312.15944929, 180.        ],
[313.40582079, 180.        ],
[314.63038514, 180.        ],
[315.83136739, 180.        ],
[317.00699259, 180.        ],
[318.15548578, 180.        ],
[319.275072  , 180.        ],
[320.36397631, 180.        ],
[321.42042374, 180.        ],
[322.44263935, 180.        ],
[323.42884818, 180.        ],
[324.37727528, 180.        ],
[325.28614568, 180.        ],
[326.15368445, 180.        ],
[326.97811662, 180.        ],
[327.75766723, 180.        ],
[328.49056135, 180.        ],
[329.175024  , 180.        ],
[329.80928024, 180.        ],
[330.39155511, 180.        ],
[330.92007366, 180.        ],
[331.39306094, 180.        ],
[331.80874199, 180.        ],
[332.16534185, 180.        ],
[332.46108557, 180.        ],
[332.6941982 , 180.        ],
[332.86290479, 180.        ],
```

```
                [332.96543037, 180.        ],
                [333.        , 180.        ]])
```

```
In [230]: points_np_all=points_np_merge
          print(len(points_np_all))
          print(len(points_np_merge))
          # points_np_merge=np.empty((1),dtype=object)
          points_np_all=np.empty((len(path_strings)),dtype=object)

          for k in range(len(path_strings)):
              points_np_all[k]= points_np
              print(len(points_np_all))

          #     points_np_merge=points_np_merge.append(points_np)
          # points_np_merge[1]= points_np
          # print(len(points_np_all))
```

```
276
276
1
```

```
In [226]: len(path_strings)
```

```
Out[226]: 1
```

```
In [161]: len(path_strings)
          path_strings
```

```
Out[161]: ['M108,63c0,0,35.157,0,56.25,0s42.188,29.25,56.25,58.5  s35.156,58.5,56.25,58.5s56.25
```

```
In [167]: arr = np.empty((0,2), float)
          arr = np.append(arr, points_np, axis=0)
          arr
```

```
Out[167]: array([[276.75      , 180.        ],
                 [277.91423178, 180.        ],
                 [279.10458017, 180.        ],
                 [280.31927023, 180.        ],
                 [281.55652699, 180.        ],
                 [282.81457551, 180.        ],
                 [284.09164082, 180.        ],
                 [285.38594798, 180.        ],
                 [286.69572204, 180.        ],
                 [288.01918803, 180.        ],
                 [289.354571  , 180.        ],
                 [290.700096  , 180.        ],
                 [292.05398808, 180.        ],
                 [293.41447227, 180.        ],
```

```
                [294.77977363, 180.        ],
                [296.14811721, 180.        ],
                [297.51772804, 180.        ],
                [298.88683117, 180.        ],
                [300.25365165, 180.        ],
                [301.61641453, 180.        ],
                [302.97334485, 180.        ],
                [304.32266766, 180.        ],
                [305.662608  , 180.        ],
                [306.99139092, 180.        ],
                [308.30724146, 180.        ],
                [309.60838467, 180.        ],
                [310.8930456 , 180.        ],
                [312.15944929, 180.        ],
                [313.40582079, 180.        ],
                [314.63038514, 180.        ],
                [315.83136739, 180.        ],
                [317.00699259, 180.        ],
                [318.15548578, 180.        ],
                [319.275072  , 180.        ],
                [320.36397631, 180.        ],
                [321.42042374, 180.        ],
                [322.44263935, 180.        ],
                [323.42884818, 180.        ],
                [324.37727528, 180.        ],
                [325.28614568, 180.        ],
                [326.15368445, 180.        ],
                [326.97811662, 180.        ],
                [327.75766723, 180.        ],
                [328.49056135, 180.        ],
                [329.175024  , 180.        ],
                [329.80928024, 180.        ],
                [330.39155511, 180.        ],
                [330.92007366, 180.        ],
                [331.39306094, 180.        ],
                [331.80874199, 180.        ],
                [332.16534185, 180.        ],
                [332.46108557, 180.        ],
                [332.6941982 , 180.        ],
                [332.86290479, 180.        ],
                [332.96543037, 180.        ],
                [333.        , 180.        ]])
```

```
In [170]: arr = np.append(arr, points_np, axis=0)
          arr
          len(arr)
```

```
Out[170]: 224
```

```
In [109]: len(points_np)
          #points_np
          points_np

Out[109]: array([[108.,  63.],
                 [108.,  63.]])

In [139]: k
          points_np_merge.shape
          #np.vstack(points_np_merge,points_np)
          points_np.shape

Out[139]: (56, 2)

In [147]: #print(points_np_merge)
          #print(points_np)
          np.append(points_np_merge,points_np,axis=0)

Out[147]: array([[108.        ,  63.        ],
                 [108.        ,  63.        ],
                 [108.        ,  63.        ],
                 [108.0345706 ,  63.        ],
                 [108.13709904,  63.        ],
                 [108.30581024,  63.        ],
                 [108.53892914,  63.        ],
                 [108.83468069,  63.        ],
                 [109.19128982,  63.        ],
                 [109.60698147,  63.        ],
                 [110.07998057,  63.        ],
                 [110.60851207,  63.        ],
                 [111.1908009 ,  63.        ],
                 [111.825072  ,  63.        ],
                 [112.50955031,  63.        ],
                 [113.24246075,  63.        ],
                 [114.02202829,  63.        ],
                 [114.84647784,  63.        ],
                 [115.71403434,  63.        ],
                 [116.62292275,  63.        ],
                 [117.57136798,  63.        ],
                 [118.55759499,  63.        ],
                 [119.5798287 ,  63.        ],
                 [120.63629406,  63.        ],
                 [121.725216  ,  63.        ],
                 [122.84481946,  63.        ],
                 [123.99332938,  63.        ],
                 [125.1689707 ,  63.        ],
                 [126.36996835,  63.        ],
                 [127.59454727,  63.        ],
                 [128.8409324 ,  63.        ],
```

```
                [130.10734868,  63.        ],
                [131.39202104,  63.        ],
                [132.69317442,  63.        ],
                [134.00903376,  63.        ],
                [135.337824  ,  63.        ],
                [136.67777007,  63.        ],
                [138.02709692,  63.        ],
                [139.38402948,  63.        ],
                [140.74679268,  63.        ],
                [142.11361147,  63.        ],
                [143.48271078,  63.        ],
                [144.85231555,  63.        ],
                [146.22065072,  63.        ],
                [147.58594123,  63.        ],
                [148.94641201,  63.        ],
                [150.300288  ,  63.        ],
                [151.64579414,  63.        ],
                [152.98115537,  63.        ],
                [154.30459662,  63.        ],
                [155.61434283,  63.        ],
                [156.90861894,  63.        ],
                [158.18564989,  63.        ],
                [159.44366061,  63.        ],
                [160.68087605,  63.        ],
                [161.89552113,  63.        ],
                [163.0858208 ,  63.        ],
                [164.25      ,  63.        ]])
```

In [128]:
```
#points_np_merge=np.zeros((1, 2))
print(points_np_merge)
print(points_np)
points_np_merge=points_np_merge+points_np
#np.vstack((points_np_merge,points_np)
```

```
[[108.  63.]
 [108.  63.]]
[[108.          63.        ]
 [108.0345706   63.        ]
 [108.13709904  63.        ]
 [108.30581024  63.        ]
 [108.53892914  63.        ]
 [108.83468069  63.        ]
 [109.19128982  63.        ]
 [109.60698147  63.        ]
 [110.07998057  63.        ]
 [110.60851207  63.        ]
 [111.1908009   63.        ]
 [111.825072    63.        ]
```

```
[112.50955031  63.        ]
[113.24246075  63.        ]
[114.02202829  63.        ]
[114.84647784  63.        ]
[115.71403434  63.        ]
[116.62292275  63.        ]
[117.57136798  63.        ]
[118.55759499  63.        ]
[119.5798287   63.        ]
[120.63629406  63.        ]
[121.725216    63.        ]
[122.84481946  63.        ]
[123.99332938  63.        ]
[125.1689707   63.        ]
[126.36996835  63.        ]
[127.59454727  63.        ]
[128.8409324   63.        ]
[130.10734868  63.        ]
[131.39202104  63.        ]
[132.69317442  63.        ]
[134.00903376  63.        ]
[135.337824    63.        ]
[136.67777007  63.        ]
[138.02709692  63.        ]
[139.38402948  63.        ]
[140.74679268  63.        ]
[142.11361147  63.        ]
[143.48271078  63.        ]
[144.85231555  63.        ]
[146.22065072  63.        ]
[147.58594123  63.        ]
[148.94641201  63.        ]
[150.300288    63.        ]
[151.64579414  63.        ]
[152.98115537  63.        ]
[154.30459662  63.        ]
[155.61434283  63.        ]
[156.90861894  63.        ]
[158.18564989  63.        ]
[159.44366061  63.        ]
[160.68087605  63.        ]
[161.89552113  63.        ]
[163.0858208   63.        ]
[164.25        63.        ]]
```

```
ValueError                                Traceback (most recent call last)

<ipython-input-128-133a28ad3c3f> in <module>()
  2 print(points_np_merge)
  3 print(points_np)
----> 4 points_np_merge=points_np_merge+points_np
  5 #np.vstack((points_np_merge,points_np)


ValueError: operands could not be broadcast together with shapes (2,2) (56,2)
```

```
In [103]: np.zeros((0, 2))
          #size(points_np_all)

Out[103]: array([], shape=(0, 2), dtype=float64)
```

```
In [61]: print(dat)
         dir(dat.start)
         # dat.start
         # dat.end

Move(to=(108+63j))


Out[61]: ['__abs__',
          '__add__',
          '__bool__',
          '__class__',
          '__delattr__',
          '__dir__',
          '__divmod__',
          '__doc__',
          '__eq__',
          '__float__',
          '__floordiv__',
          '__format__',
          '__ge__',
          '__getattribute__',
          '__getnewargs__',
          '__gt__',
          '__hash__',
          '__init__',
          '__init_subclass__',
          '__int__',
          '__le__',
          '__lt__',
          '__mod__',
```

```
 '__mul__',
 '__ne__',
 '__neg__',
 '__new__',
 '__pos__',
 '__pow__',
 '__radd__',
 '__rdivmod__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__rfloordiv__',
 '__rmod__',
 '__rmul__',
 '__rpow__',
 '__rsub__',
 '__rtruediv__',
 '__setattr__',
 '__sizeof__',
 '__str__',
 '__sub__',
 '__subclasshook__',
 '__truediv__',
 'conjugate',
 'imag',
 'real']
```

```
In [43]: points_np
```

```
Out[43]: array([[108.        ,  63.        ],
                [108.01071052,  63.        ],
                [108.04263917,  63.        ],
                [108.09548159,  63.        ],
                [108.16893341,  63.        ],
                [108.26269026,  63.        ],
                [108.37644778,  63.        ],
                [108.50990161,  63.        ],
                [108.66274736,  63.        ],
                [108.83468069,  63.        ],
                [109.02539722,  63.        ],
                [109.23459259,  63.        ],
                [109.46196243,  63.        ],
                [109.70720238,  63.        ],
                [109.97000807,  63.        ],
                [110.25007513,  63.        ],
                [110.5470992 ,  63.        ],
                [110.86077591,  63.        ],
                [111.1908009 ,  63.        ],
```

```
[111.5368698 ,  63.        ],
[111.89867824,  63.        ],
[112.27592186,  63.        ],
[112.6682963 ,  63.        ],
[113.07549718,  63.        ],
[113.49722014,  63.        ],
[113.93316081,  63.        ],
[114.38301483,  63.        ],
[114.84647784,  63.        ],
[115.32324546,  63.        ],
[115.81301333,  63.        ],
[116.31547708,  63.        ],
[116.83033236,  63.        ],
[117.35727479,  63.        ],
[117.896     ,  63.        ],
[118.44620363,  63.        ],
[119.00758132,  63.        ],
[119.5798287 ,  63.        ],
[120.1626414 ,  63.        ],
[120.75571506,  63.        ],
[121.3587453 ,  63.        ],
[121.97142778,  63.        ],
[122.59345811,  63.        ],
[123.22453193,  63.        ],
[123.86434488,  63.        ],
[124.51259259,  63.        ],
[125.1689707 ,  63.        ],
[125.83317483,  63.        ],
[126.50490063,  63.        ],
[127.18384373,  63.        ],
[127.86969975,  63.        ],
[128.56216434,  63.        ],
[129.26093313,  63.        ],
[129.96570176,  63.        ],
[130.67616585,  63.        ],
[131.39202104,  63.        ],
[132.11296296,  63.        ],
[132.83868726,  63.        ],
[133.56888956,  63.        ],
[134.30326549,  63.        ],
[135.0415107 ,  63.        ],
[135.78332081,  63.        ],
[136.52839146,  63.        ],
[137.27641829,  63.        ],
[138.02709692,  63.        ],
[138.78012299,  63.        ],
[139.53519214,  63.        ],
[140.292     ,  63.        ],
```

```
                [141.0502422 ,  63.        ],
                [141.80961438,  63.        ],
                [142.56981217,  63.        ],
                [143.33053121,  63.        ],
                [144.09146712,  63.        ],
                [144.85231555,  63.        ],
                [145.61277213,  63.        ],
                [146.37253249,  63.        ],
                [147.13129226,  63.        ],
                [147.88874709,  63.        ],
                [148.64459259,  63.        ],
                [149.39852442,  63.        ],
                [150.15023819,  63.        ],
                [150.89942956,  63.        ],
                [151.64579414,  63.        ],
                [152.38902758,  63.        ],
                [153.1288255 ,  63.        ],
                [153.86488355,  63.        ],
                [154.59689735,  63.        ],
                [155.32456254,  63.        ],
                [156.04757476,  63.        ],
                [156.76562963,  63.        ],
                [157.4784228 ,  63.        ],
                [158.18564989,  63.        ],
                [158.88700654,  63.        ],
                [159.58218839,  63.        ],
                [160.27089106,  63.        ],
                [160.9528102 ,  63.        ],
                [161.62764143,  63.        ],
                [162.29508039,  63.        ],
                [162.95482272,  63.        ],
                [163.60656404,  63.        ],
                [164.25      ,  63.        ]])

In [31]: dat=path[key]
         dat

Out[31]: CubicBezier(start=(220.5+121.5j), control1=(234.562+150.75j), control2=(255.656+180j)

In [41]: key=0
         dat=path[key]
         dat

Out[41]: Move(to=(108+63j))

In [9]: block=0
        n_dots=100
        key=3
```

```
        path=parse_path(path_strings[block])
        dat=path[key]

        start_np = np.array([dat.start.real, dat.start.imag])
        end_np = np.array([dat.end.real, dat.end.imag])

        print(start_np)
        print(end_np)

        diff_np=start_np-end_np
        n_dots=np.round(np.linalg.norm(diff_np))

        np.array([converted_curve(t) for t in np.linspace(0, 1, n_dots)])
```

```
[220.5 121.5]
[276.75 180.  ]
```

```
In [10]: n_dots
```

```
Out[10]: 81.0
```

```
In [13]: t=0.5
         start_np=np.array([0,0])
         end_np=np.array([100,100])
         (1-t)*start_np+t*end_np
```

```
Out[13]: array([50., 50.])
```

```
In [16]: def line_splitter(start, end):

             return (lambda t: (1-t)*start+t*end)
```

```
In [34]: diff_np=start_np-end_np
         n_dots=np.round(np.linalg.norm(diff_np))

         converted_line = line_splitter(start_np,end_np)
         np.array([converted_line(t) for t in np.linspace(0, 1, n_dots)])
```

```
Out[34]: array([[220.5     , 121.5     ],
                [221.203125, 122.23125 ],
                [221.90625 , 122.9625  ],
                [222.609375, 123.69375 ],
                [223.3125  , 124.425   ],
                [224.015625, 125.15625 ],
                [224.71875 , 125.8875  ],
                [225.421875, 126.61875 ],
                [226.125   , 127.35    ],
                [226.828125, 128.08125 ],
```

```
[227.53125 , 128.8125  ],
[228.234375, 129.54375 ],
[228.9375  , 130.275   ],
[229.640625, 131.00625 ],
[230.34375 , 131.7375  ],
[231.046875, 132.46875 ],
[231.75    , 133.2     ],
[232.453125, 133.93125 ],
[233.15625 , 134.6625  ],
[233.859375, 135.39375 ],
[234.5625  , 136.125   ],
[235.265625, 136.85625 ],
[235.96875 , 137.5875  ],
[236.671875, 138.31875 ],
[237.375   , 139.05    ],
[238.078125, 139.78125 ],
[238.78125 , 140.5125  ],
[239.484375, 141.24375 ],
[240.1875  , 141.975   ],
[240.890625, 142.70625 ],
[241.59375 , 143.4375  ],
[242.296875, 144.16875 ],
[243.      , 144.9     ],
[243.703125, 145.63125 ],
[244.40625 , 146.3625  ],
[245.109375, 147.09375 ],
[245.8125  , 147.825   ],
[246.515625, 148.55625 ],
[247.21875 , 149.2875  ],
[247.921875, 150.01875 ],
[248.625   , 150.75    ],
[249.328125, 151.48125 ],
[250.03125 , 152.2125  ],
[250.734375, 152.94375 ],
[251.4375  , 153.675   ],
[252.140625, 154.40625 ],
[252.84375 , 155.1375  ],
[253.546875, 155.86875 ],
[254.25    , 156.6     ],
[254.953125, 157.33125 ],
[255.65625 , 158.0625  ],
[256.359375, 158.79375 ],
[257.0625  , 159.525   ],
[257.765625, 160.25625 ],
[258.46875 , 160.9875  ],
[259.171875, 161.71875 ],
[259.875   , 162.45    ],
[260.578125, 163.18125 ],
```

```
                [261.28125 , 163.9125  ],
                [261.984375, 164.64375 ],
                [262.6875  , 165.375   ],
                [263.390625, 166.10625 ],
                [264.09375 , 166.8375  ],
                [264.796875, 167.56875 ],
                [265.5     , 168.3     ],
                [266.203125, 169.03125 ],
                [266.90625 , 169.7625  ],
                [267.609375, 170.49375 ],
                [268.3125  , 171.225   ],
                [269.015625, 171.95625 ],
                [269.71875 , 172.6875  ],
                [270.421875, 173.41875 ],
                [271.125   , 174.15    ],
                [271.828125, 174.88125 ],
                [272.53125 , 175.6125  ],
                [273.234375, 176.34375 ],
                [273.9375  , 177.075   ],
                [274.640625, 177.80625 ],
                [275.34375 , 178.5375  ],
                [276.046875, 179.26875 ],
                [276.75    , 180.      ]])

In [22]: n_dots

Out[22]: 141.0

In [26]: diff_np=start_np-end_np
         n_dots=np.round(np.linalg.norm(diff_np))
         n_dots

Out[26]: 141.0

In [ ]:
```