

```
1 # === pieChartDemo.ipynb ===
2 #%%
3 %matplotlib notebook
4 #get_ipython().run_line_magic('matplotlib', 'notebook')
5
6
7 #%%
8 # Import our dependencies
9 import matplotlib.pyplot as plt
10 import numpy as np
11
12
13 #%%
14 # Labels for the sections of our pie chart
15 labels = ["Humans", "Smurfs", "Hobbits", "Ninjas"]
16
17 # The values of each section of the pie chart
18 sizes = [220, 95, 80, 100]
19
20 # The colors of each section of the pie chart
21 colors = ["red", "orange", "lightcoral", "lightskyblue"]
22
23 # Tells matplotlib to separate the "Python" section from the others
24 explode = (0.1, 0, 0, 0)
25
26
27 #%%
28 # Creates the pie chart based upon the values above
29 # Automatically finds the percentages of each part of the pie chart
30 # -----
31 # the sizes of each wedge are passed into `plt.pie()` as an arrayList
32 # containing the labels for each wedge and the colors for each wedge are also
33 # passed in.
34 #
35 # we can also choose a wedge to "explode," using the `explode` option.
36 # This will separate one wedge from the rest so that it is easier to examine
37 #
38 # `autopct="%1.1f%%"` is being passed to automatically convert the value passed
39 # in to percentages with one decimal place
40 # -----
41 plt.pie(sizes, explode=explode, labels=labels, colors=colors,
42         autopct="%1.1f%%", shadow=True, startangle=140)
43
44
45 #%%
46 # -----
47 # Matplotlib does not constrain pie charts to be circular – by default,
48 # they will be ovals if the window the plot lives in is not a square.
49 # This is why `plt.axis("equal")` is being passed.
50 # -----
51
52 # Tells matplotlib that we want a pie chart with equal axes
53 plt.axis("equal")
54
55
```

```
56  #%%
57  # there are additional configuration options available for improving the
58  # appearance of Matplotlib's pie charts should we desire to look into them
```