

Part 1: Basic Python Syntax

Create a variable, var, and set the value to the string “Hello, world!”.

```
In [1]: var = "Hello, world!"  
var
```

```
Out[1]: 'Hello, world!'
```

Now print the length of var.

```
In [2]: len(var)
```

```
Out[2]: 13
```

Extract “world” from var using string index slicing.

```
In [3]: v = var[7:12]  
v
```

```
Out[3]: 'world'
```

Print the types for 1, 1.0, and “1” and note how they differ.

```
In [4]: i = 1  
f = 1.0  
s = "1"  
  
print(type(i))  
print(type(f))  
print(type(s))  
  
<class 'int'>  
<class 'float'>  
<class 'str'>
```

The input for them is different one has 1 its a integer whereas 1.0 is a float and "1" is for a defining a string.

so the type of variable also is changed accordingly.

Calculate and print the area of a circle of radius 5 to two decimal places.

```
In [5]: import math

r = 5
area = 2*math.pi*r*r
print("{:.2f}".format(area))

157.08
```

Write a for loop that prints the numbers 0 to 9.

```
In [6]: for i in range (0,10):
        print(i)

0
1
2
3
4
5
6
7
8
9
```

Write a for loop that prints only even numbers from 10 to 20.

```
In [7]: for i in range (10,20):
        if(i%2 == 0):
            print(i)

10
12
14
16
18
```

Write an if/else statement that takes a variable var and prints

“Greater than 10” if var is greater than 10,

“Less than 10” if var is less than 10,

“Equal to 10” if var is equal to 10.

Make sure to appropriately test for each case.

```
In [8]: j = [10,5,20,25]

for var in j:
    if(var >10):
        print( str(var) + "- Greater than 10")
    elif(var<10):
        print( str(var) + "- Less than 10")
    elif(var ==10):
        print( str(var) + "- Equal to 10")
```

```
10- Equal to 10
5- Less than 10
20- Greater than 10
25- Greater than 10
```

Write a function which takes a string as its parameter and returns the length of the string.

```
In [9]: def strlen(s):
        print(len(s))

st = input("enter a string:")
strlen(st)
```

```
enter a string:Hello, world!!!
15
```

Write a function which takes a positive integer n as its parameter, and prints out all odd numbers from 1 to n.

```
In [10]: var = int(input("Enter the max number:"))  
  
for i in range(1,var):  
    if(i%2 != 0):  
        print(i)
```

Enter the max number:20

1
3
5
7
9
11
13
15
17
19

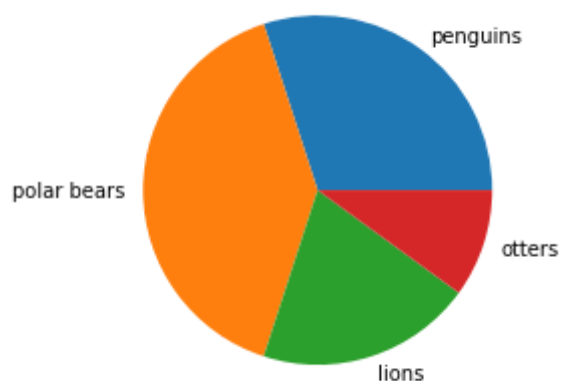
Part 2: Simple Charts

Pie Chart:

Make a pie chart that shows the distribution of animals in a zoo that contains

30 penguins, 40 polar bears, 20 lions, and 10 otters.

```
In [11]: import matplotlib.pyplot as plt  
import numpy as np  
  
y = np.array([30, 40, 20, 10])  
mylabels = ["penguins", "polar bears", "lions", "otters"]  
  
plt.pie(y, labels= mylabels)  
plt.show()
```



Bar Chart:

Make a bar chart that shows the number of people, out of a total of 40,

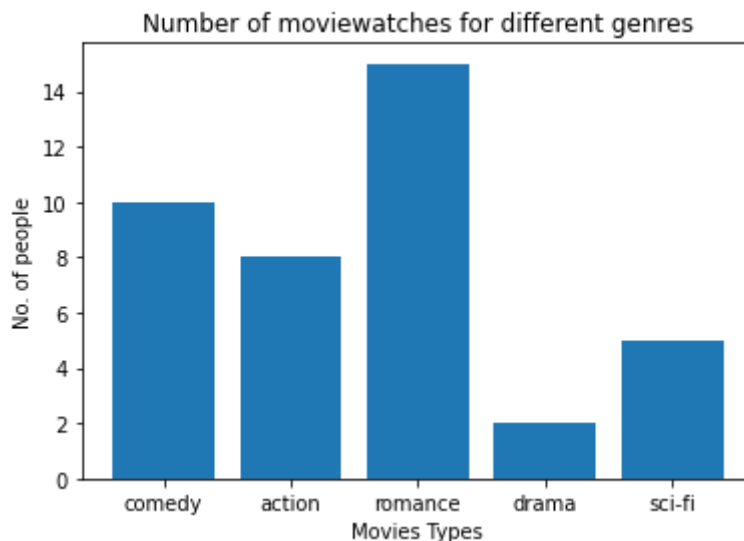
who like various movie types if 10 people like comedy; 8 people like action;

15 people like romance; 2 people like drama, and 5 people like sci-fi.

```
In [12]: data = {'comedy':10, 'action':8, 'romance':15, 'drama':2, 'sci-fi':5}
type = list(data.keys())
no = list(data.values())

plt.bar(type,no)

plt.xlabel("Movies Types")
plt.ylabel("No. of people")
plt.title("Number of moviewatches for different genres")
plt.show()
```



Grouped Bar Plots:

Make both a side-by-side bar plot and a stacked bar plot that displays the number of

child visitors and the number of adult visitors at a waterpark in the months of April, May, June and July.

Be sure to include titles, legends and appropriate labels sufficiently sized for readability.

April**Children: 680****Adults: 215****May****Children: 950****Adults: 400****June****Children: 2056****Adults: 1100****July****Children: 4502****Adults: 1800**

In [13]:

```
X = ['April', 'May', 'June', 'July']
children = [680, 950, 2056, 450]
adults = [215, 400, 1100, 1800]

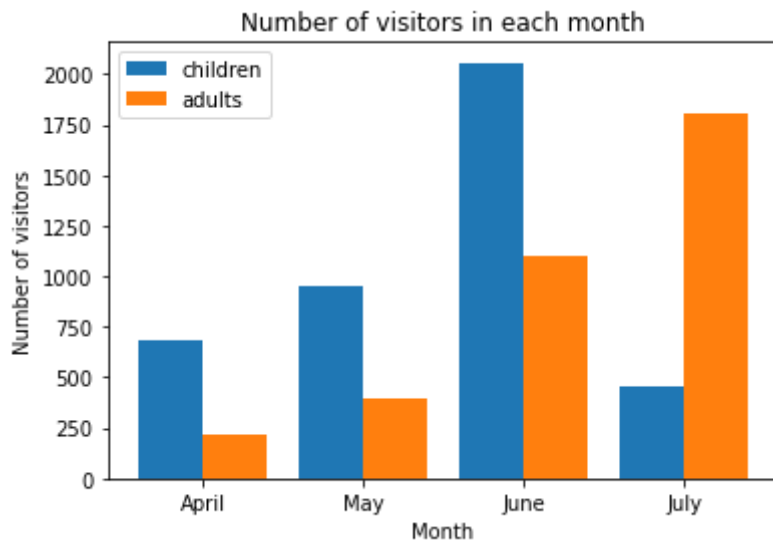
X_axis = np.arange(len(X))

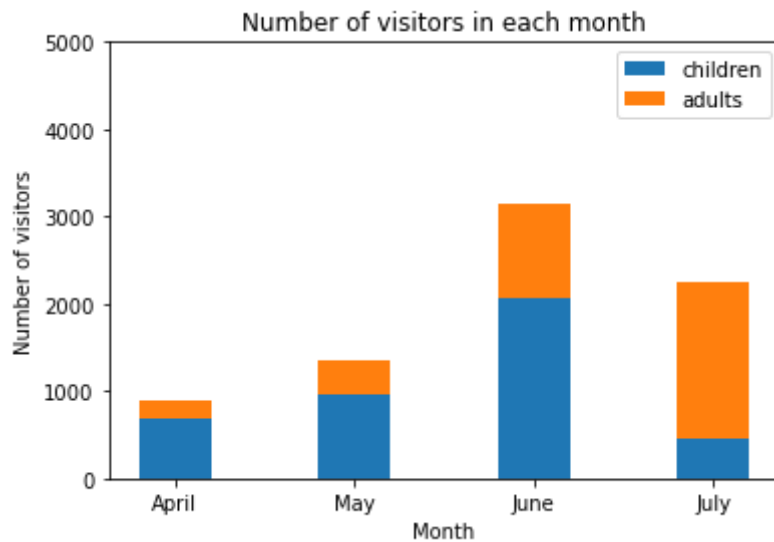
plt.bar(X_axis - 0.2, children, 0.4, label = 'children')
plt.bar(X_axis + 0.2, adults, 0.4, label = 'adults')

plt.xticks(X_axis, X)
plt.xlabel("Month")
plt.ylabel("Number of visitors")
plt.title("Number of visitors in each month")
plt.legend()
plt.show()

plt.bar(X_axis, children, 0.4, label = 'children')
plt.bar(X_axis, adults, 0.4, bottom=children, label = 'adults')

plt.xticks(X_axis, X)
plt.xlabel("Month")
plt.ylabel("Number of visitors")
plt.title("Number of visitors in each month")
plt.ylim([0, 5000])
plt.legend()
plt.show()
```





Histogram:

Make a histogram of the following scores. Feel free to experiment on the best number of histogram bins for visualization.

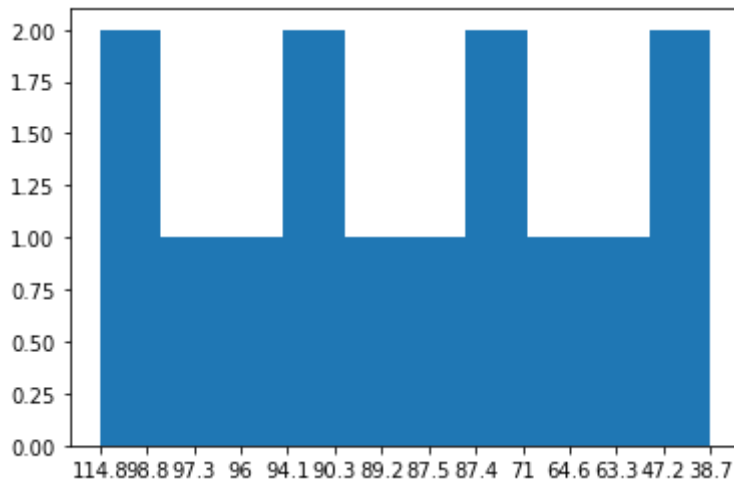
114.8, 98.8, 97.3, 96, 94.1, 93.1, 93.1, 91.6, 91.5, 91.3, 90.3, 89.2, 87.5,

87.4, 85.2, 81.7, 81.6, 81.5, 80, 79.3, 78.2, 77.6, 77.1, 76.7, 75.1, 73.9,

72, 71, 64.6, 63.3, 47.2, 38.7


```
In [14]: h = ['114.8', '98.8', '97.3', '96', '94.1', '90.3', '89.2', '87.5', '87.4',  
             '71', '64.6', '63.3', '47.2', '38.7']  
plt.hist(h)
```

```
Out[14]: (array([2., 1., 1., 2., 1., 1., 2., 1., 1., 2.]),  
         array([ 0. ,  1.3,  2.6,  3.9,  5.2,  6.5,  7.8,  9.1, 10.4, 11.7, 13.  
         ]),  
         <BarContainer object of 10 artists>)
```



```
In [ ]:
```