### Assignment 1

### Part 1: Using JupyterLab

## 1.1. Running Cells and Displaying Output

In [36]:

```
"Will this line be displayed?"
```

Out[36]:

```
'Will this line be displayed?'
```

In [37]:

```
print("Hello" + ",", "world!")
```

```
Hello, world!
```

In [38]:

```
5 + 3
```

Out[38]:

```
8
```

## 1.2 Viewing Documentation

In [39]:

```
help(print)
```

```
Help on built-in function print in module builtins:

print(...)
    print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)

    Prints the values to a stream, or to sys.stdout by default.
    Optional keyword arguments:
    file:  a file-like object (stream); defaults to the current sys.stdout.
    sep:   string inserted between values, default a space.
    end:   string appended after the last value, default a newline.
    flush: whether to forcibly flush the stream.
```

In [40]:

```
print('Welcome. ')
```

```
Welcome.
```

## 1.3 Importing Libraries

In [41]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

### Part 2: Python

## 2.1

In [42]:

```
def summation(n):
    total = 0
    for i in range(1,n+1):
        sum = pow(i,3) + 3 * pow(i,2)
        total = total + sum
    print("Total summation: ", total)

n=int(input("Enter n value: "))
summation(n)
```

```
Total summation:  78
```

### 2.2

In [43]:

```
%%time
def list_sum(list_a, list_b):
    square_list = [number ** 2 for number in list_a]

    cube_list = [number * number * number for number in list_b]
    sum = [a + b for a, b in zip(square_list, cube_list)]
    print(sum)

def main():
    list_1 = [int(x) for x in input().split()]
    list_2 = [int(y) for y in input().split()]
    list_sum(list_1, list_2)

main()
```

```
[65, 129, 225]
Wall time: 12.2 s
```

## Part 3: Numpy

### 3.1

In [44]:

```
arr1 = [1,2,3]
arr2 = [4,5,6]
```

### 3.2

In [45]:

```
sum = [a + b for a, b in zip(arr1, arr2)]
print(sum)
```

```
[5, 7, 9]
```

### 3.3

In [46]:

```
%%time
def array_sum(list_a, list_b):
    square_list = [number ** 2 for number in list_a]
    cube_list = [number * number * number for number in list_b]
    array_1 = np.array(square_list)
    array_2 = np.array(cube_list)

    sum = array_1 + array_2
    print(sum)

def main():
    list_1 = [int(x) for x in input().split()]
    list_2 = [int(y) for y in input().split()]
    array_sum(list_1, list_2)

main()
```

```
[ 65 129 225]
Wall time: 4.19 s
```

### 3.4

**list_sum's execution time: 3.79s array_sum's execution time: 2.64s array_sum is 1.15s faster than list_sum.**

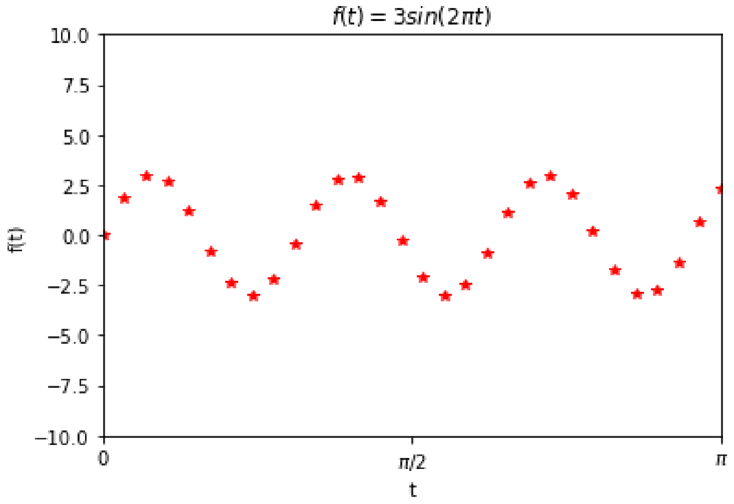
### Part 4: Matplotlib

In [47]:

```
x=np.linspace(0,np.pi,30)
y = 3*np.sin(2*np.pi*x)
plt.plot(x,y, '*r')
plt.axis([0,np.pi, -10, 10])
plt.xticks([0,np.pi/2,np.pi],(0,r'$\pi /2$',r'$\pi$' ))
plt.xlabel('t')
plt.ylabel('f(t)')
plt.title(r'$f(t) = 3sin(2\pi t)$')
```

Out[47]:

```
Text(0.5, 1.0, '$f(t) = 3sin(2\\pi t)$')
```


$f(t) = 3sin(2\pi t)$
f(t)
π/2
π
t