```
In [10]: float(30)
```

```
Out[10]: 30.0
```

```
In [11]: float(True)
```

```
Out[11]: 1.0
```

```
In [14]: float(1+3j)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[14], line 1
----> 1 float(1+3j)

TypeError: float() argument must be a string or a real number, not 'complex'
```

```
In [13]: float('ten')
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[13], line 1
----> 1 float('ten')

ValueError: could not convert string to float: 'ten'
```

```
In [15]: float('10')
```

```
Out[15]: 10.0
```

```
In [16]: complex(10)
```

```
Out[16]: (10+0j)
```

```
In [17]: complex(20,30)
```

```
Out[17]: (20+30j)
```

```
In [18]: complex(10,20,30)#complex take make 2 arg
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[18], line 1
----> 1 complex(10,20,30)

TypeError: complex() takes at most 2 arguments (3 given)
```

```
In [19]: complex(2.6)

Out[19]: (2.6+0j)

In [20]: complex(2.6,20)

Out[20]: (2.6+20j)

In [22]: complex(True)

Out[22]: (1+0j)

In [23]: complex(False)

Out[23]: 0j

In [ ]: complex('20')

In [ ]: bool(1)

In [ ]: bool(0)

In [ ]: bool(0.23)

In [ ]: bool(25)

In [ ]: bool(10+20j)

In [ ]: bool('ten')

In [4]: index='HELLOPYTHON'
        index
```

```
Out[4]:  'HELLOPYTHON'
```

```
In [5]:  index[2:-1]
```

```
Out[5]:  'LLOPYTHO'
```

```
In [6]:  index[2:-2]
```

```
Out[6]:  'LLOPYTH'
```

```
In [7]:  index[::-2]
```

```
Out[7]:  'NHYOLH'
```

```
In [8]:  index[::-4]
```

```
Out[8]:  'NYL'
```

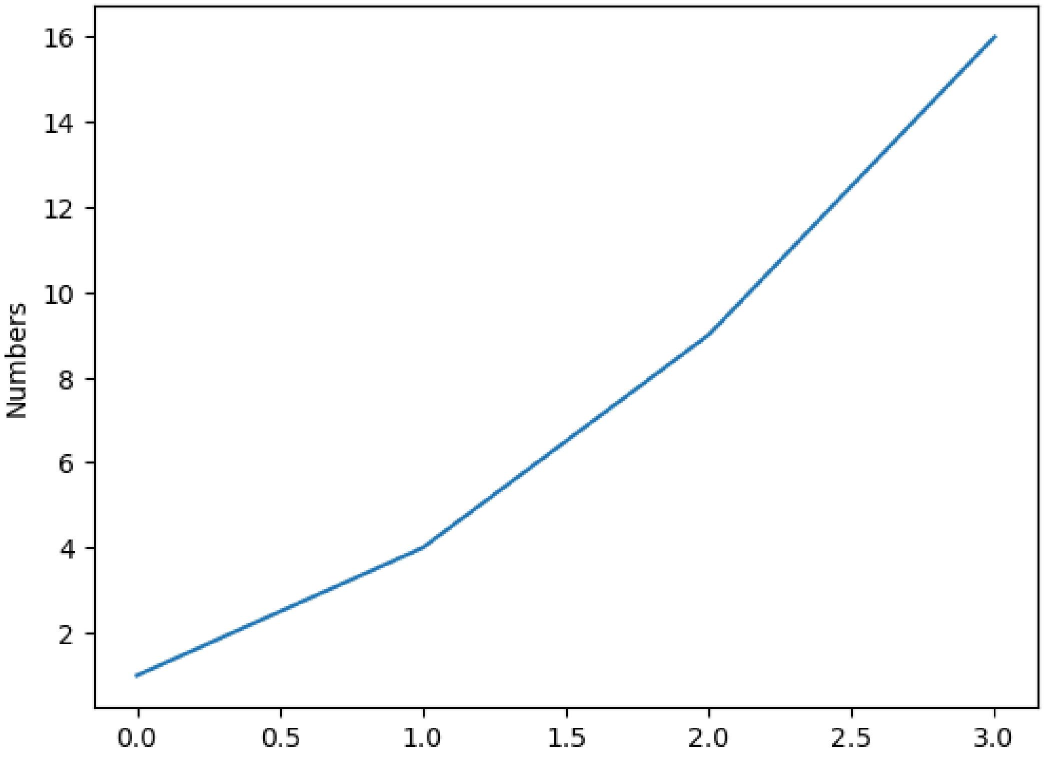
```
In [9]:  index[:-4]
```

```
Out[9]:  'HELLOPY'
```

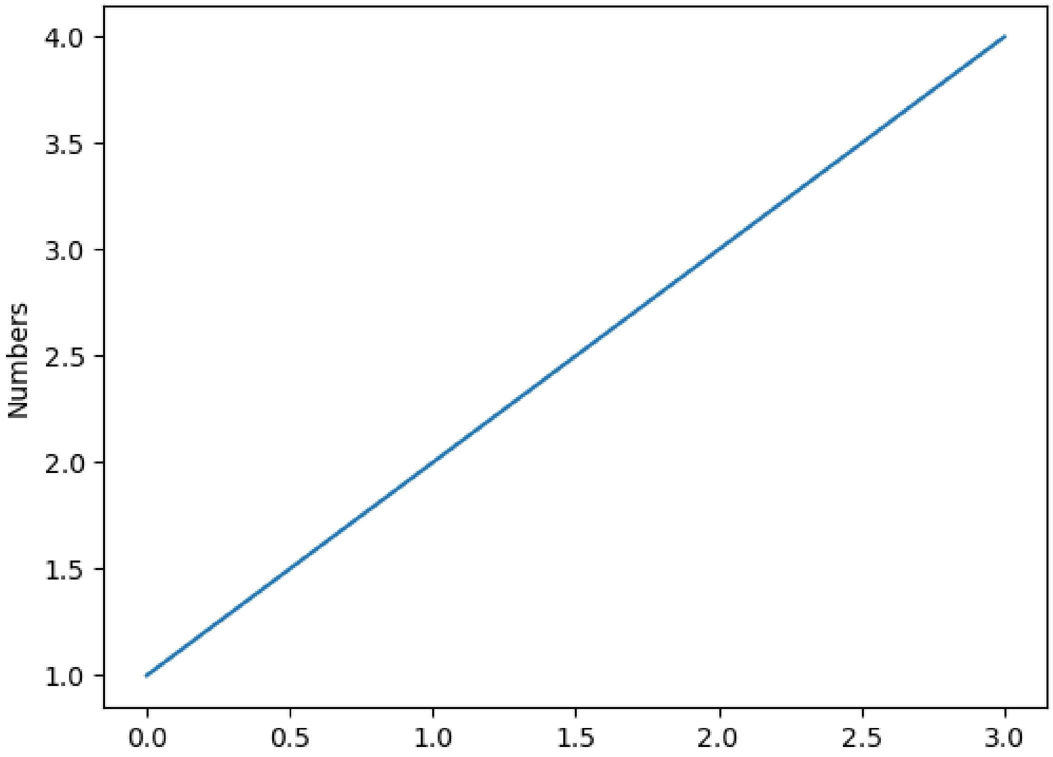
```
In [10]:  index[1:10:3]
```

```
Out[10]:  'EOT'
```

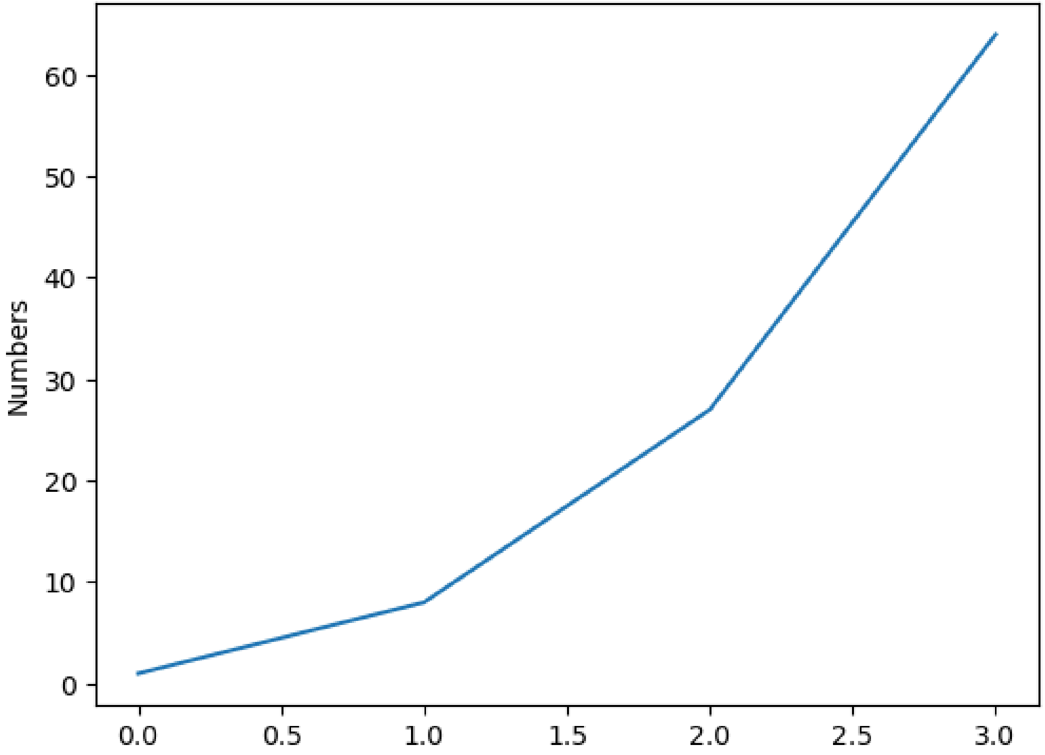
```
In [11]:  import matplotlib.pyplot as plt
          plt.plot([1,4,9,16])
          plt.ylabel('Numbers')
          plt.show()
```



```
In [12]: import matplotlib.pyplot as plt
         plt.plot([1,2,3,4])
         plt.ylabel('Numbers')
         plt.show()
```

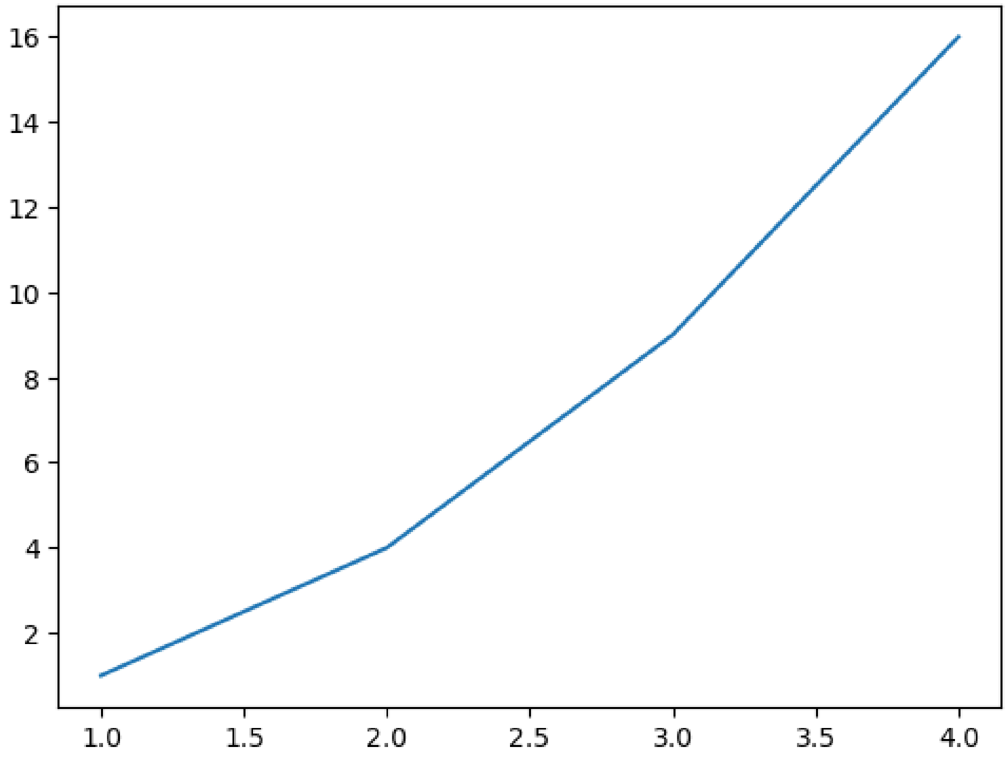


In [13]:
```python
import matplotlib.pyplot as plt
plt.plot([1,8,27,64])
plt.ylabel('Numbers')
plt.show()
```



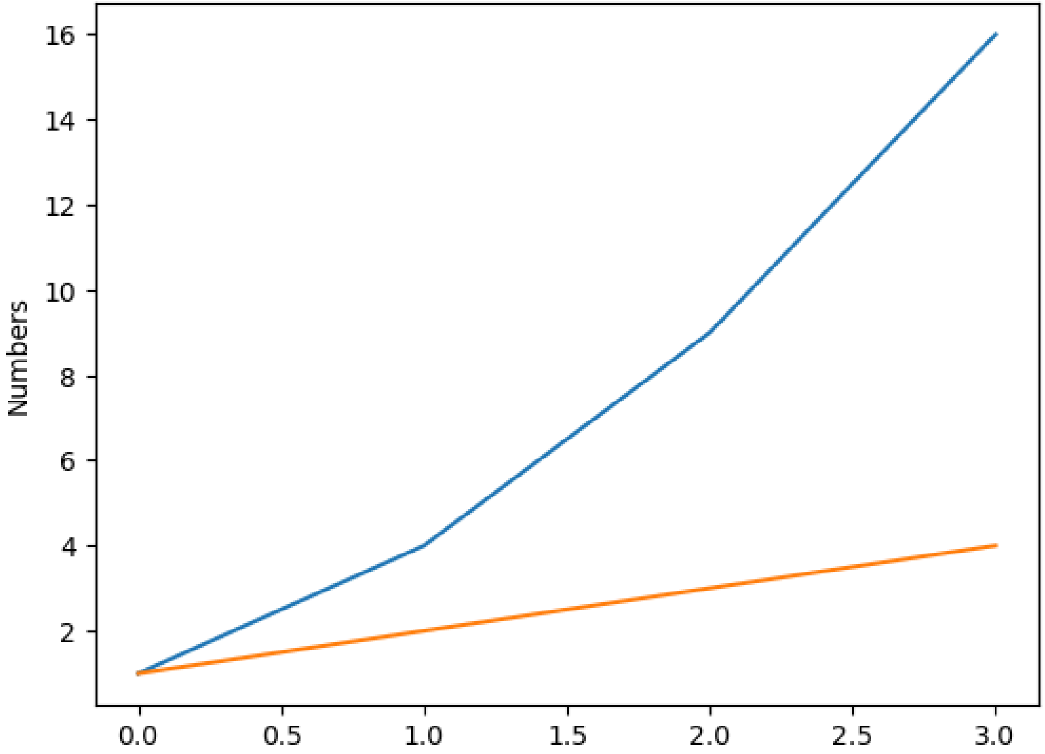
In [14]:

```
import matplotlib.pyplot as plt
plt.plot([1,2,3,4],[1,4,9,16])
plt.axis()
plt.show()
```

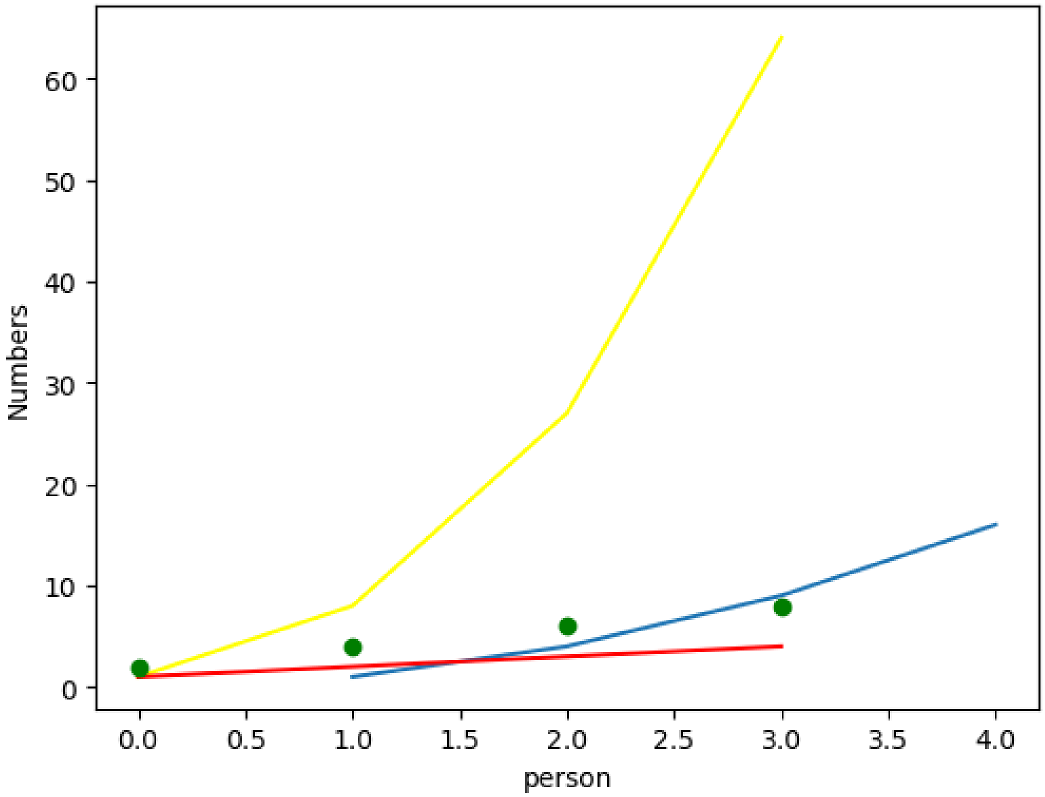


In [15]:

```
import matplotlib.pyplot as plt
plt.plot([1,4,9,16])
plt.plot([1,2,3,4])
plt.axis()
plt.ylabel('Numbers')
plt.show()
```



In [16]:
```
import matplotlib.pyplot as plt
plt.plot([1,2,3,4],[1,4,9,16],[1,8,27,64],'yellow')
plt.plot([1,2,3,4],'red')
plt.plot([2,4,6,8],'go')
plt.ylabel('Numbers')
plt.xlabel('person')
plt.show()
```



```
In [17]: thislist=["apple","banana","cherry"]
         print (thislist)

         ['apple', 'banana', 'cherry']
```

```
In [18]: l=[]
         l

Out[18]: []
```

```
In [19]: type(l)

Out[19]: list
```

```
In [20]: l
```

```
Out[20]: []
```

```
In [21]: l.append(10)
         l
```

```
Out[21]: [10]
```

```
In [23]: l.append(20)
         l
```

```
Out[23]: [10, 20]
```

```
In [24]: len(l)
```

```
Out[24]: 2
```

```
In [25]: l
```

```
Out[25]: [10, 20]
```

```
In [26]: l.remove(20)
         l
```

```
Out[26]: [10]
```

```
In [27]: l1=l.copy()
         l1
```

```
Out[27]: [10]
```

```
In [28]: l==l1
```

```
Out[28]: True
```

```
In [29]: l
```

```
Out[29]: [10]
```

```
In [30]: l.count(0)
```

Out[30]:
```
0
```

In [33]:
```
l.append(10)
l
```

Out[33]:
```
[10, 10, 10]
```

In [34]:
```
l.count(10)
```

Out[34]:
```
3
```

In [35]:
```
l==l1
```

Out[35]:
```
False
```

In [36]:
```
l[:]
```

Out[36]:
```
[10, 10, 10]
```

In [37]:
```
l2=[]
l2
```

Out[37]:
```
[]
```

In [38]:
```
l2.append(1)
l2.append(0.56)
l2.append('True')
l2.append(0+23j)
l2.append('nit')
l2
```

Out[38]:
```
[1, 0.56, 'True', 23j, 'nit']
```

In [39]:
```
l3=l2.copy()
l3
```

Out[39]:
```
[1, 0.56, 'True', 23j, 'nit']
```

In [40]:
```
len(l3)
```

```
Out[40]: 5

In [41]: l3.clear()

In [42]: l3

Out[42]: []

In [43]: del l3

In [ ]: l3

In [44]: l2

Out[44]: [1, 0.56, 'True', 23j, 'nit']

In [45]: l2

Out[45]: [1, 0.56, 'True', 23j, 'nit']

In [46]: l2.index(0.56)

Out[46]: 1

In [47]: l2.insert(5,'mit college')
         l2

Out[47]: [1, 0.56, 'True', 23j, 'nit', 'mit college']

In [48]: l2.insert(3,'False')
         l2

Out[48]: [1, 0.56, 'True', 'False', 23j, 'nit', 'mit college']

In [49]: l2.pop()
         l2

Out[49]: [1, 0.56, 'True', 'False', 23j, 'nit']
```

In [50]:
```
l2.pop(3)
l2
```

Out[50]:
```
[1, 0.56, 'True', 23j, 'nit']
```

In [51]:
```
l3=[10,100,3,45,75,23]
l3
```

Out[51]:
```
[10, 100, 3, 45, 75, 23]
```

In [ ]:
```
l3.sort()
l3
```

In [52]:
```
l3.sort(reverse=False)
l3
```

Out[52]:
```
[3, 10, 23, 45, 75, 100]
```

In [ ]:
```
l3.sort(reverse=True)
l3
```

In [54]:
```
l4=['c','i','u','a','e']
l4
```

Out[54]:
```
['c', 'i', 'u', 'a', 'e']
```

In [55]:
```
l4.sort()
l4
```

Out[55]:
```
['a', 'c', 'e', 'i', 'u']
```

In [56]:
```
l2
```

Out[56]:
```
[1, 0.56, 'True', 23j, 'nit']
```

In [59]:
```
print(l2[2][0])
print(l2[2][1])
print(l2[2][2])
print(l2[2][3])
```

```
T
r
u
e
```

In [60]:
```
l2
```

Out[60]:
```
[1, 0.56, 'True', 23j, 'nit']
```

In [68]:
```
for i in (l2):
print(i)
```

```
  Cell In[68], line 2
    print(i)
    ^
IndentationError: expected an indented block after 'for' statement on line 1
```

In [65]:
```
l
```

Out[65]:
```
[10, 10, 10]
```

In [66]:
```
l[::-1]
```

Out[66]:
```
[10, 10, 10]
```

In [67]:
```
l[::3]
```

Out[67]:
```
[10]
```

In [69]:
```
l[2:]
```

Out[69]:
```
[10]
```

In [70]:
```
l[:3]
```

Out[70]:
```
[10, 10, 10]
```

In [72]:
```
l[0:3:2]
```

Out[72]:
```
[10, 10]
```

In [73]:
```
t=()
t
```

Out[73]:
```
()
```

In [74]:
```
type(t)
```

Out[74]:
```
tuple
```

In [75]:
```
t=(10,10,20,30,40)
t
```

Out[75]:
```
(10, 10, 20, 30, 40)
```

In [77]:
```
icici=(1234,'cizp','3rdsep')
icici
```

Out[77]:
```
(1234, 'cizp', '3rdsep')
```

In [78]:
```
t
```

Out[78]:
```
(10, 10, 20, 30, 40)
```

In [79]:
```
t.count(10)
```

Out[79]:
```
2
```

In [80]:
```
t.index(40)
```

Out[80]:
```
4
```

In [82]:
```
for i in t:
    print(i)
```

```
10
10
20
30
40
```

```
In [87]: t
```

```
Out[87]: (10, 10, 20, 30, 40)
```

```
In [88]: t1=t*4
         t1
```

```
Out[88]: (10,
          10,
          20,
          30,
          40,
          10,
          10,
          20,
          30,
          40,
          10,
          10,
          20,
          30,
          40,
          10,
          10,
          20,
          30,
          40)
```

tuple completed

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```