

```
In [11]: #Invertir una cadena de texto (pista:[::-1])
def invertir(cadena):
    return cadena[::-1]

invertir("ivan eht nioJ")
```

Out[11]: 'Join the navi'

```
In [21]: #Comprobar si un numero es capicua
def capicua(numero:int):
    num=str(numero)
    return num==num[::-1]
capicua(101)
```

Out[21]: True

```
In [47]: #Determinar si un numero es primo
def primo(numero):
    if numero>1:
        for i in range(2,numero+1):
            if(numero%i==0 and i!=numero):
                return False
        else:
            return True
    else:
        return False
primo(3)
```

Out[47]: True

```
In [66]: #Determinar cuál es el enésimo primo
def primoene(numero):
    primos=[]
    i=0
    while(numero>len(primos)):
        if(primo(i)):
            primos.append(i)
            i=i+1
        else:
            i=i+1
    return primos[numero-1]

primoene(10)
```

[2, 3, 5, 7, 11, 13, 17, 19, 23, 29]

Out[66]: 29

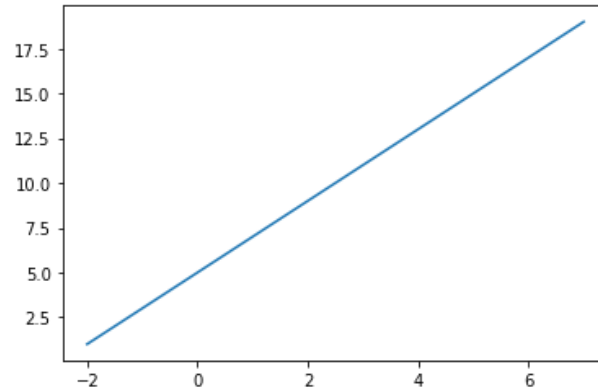
```
In [72]: #Imprimir por pantalla los divisores de un numero dado.
def divisores(numero):
    divisores=[]
    for i in range(1,numero+1):
        if(numero%i==0):
            divisores.append(i)
    print("Los divisores de",numero,"son:",divisores)

divisores(11)
```

Los divisores de 11 son: [1, 11]

In [77]: *#Graficar la funcion $y = 2x + 5$ en el intervalo $(-2, 7)$*

```
import numpy
from matplotlib import pyplot
x = numpy.linspace(-2, 7, 100)
y = 2*x+5
pyplot.plot(x, y)
pyplot.show()
```



In [124]: *'''
Resolver el siguiente sistema de ecuaciones:*

$$\begin{aligned}x + y + 3z &= 3 \\ 2x + 4y + 5z &= 5 \\ 2x + 2y + 4z &= 8\end{aligned}$$

```
'''
import numpy as np
A = np.array([[1, 1, 3],[2, 4, 5],[2, 2, 4]])
B = np.array([3,5,8])
x = np.linalg.solve(A,b)
print(x)
#Resuelto con array
[ 7. -1. -1.]
```

In [126]: **import numpy as matrix**

```
A = np.matrix([[1, 1, 3],[2, 4, 5],[2, 2, 4]])
B = np.matrix([3,5,8])
x = np.linalg.solve(A,b)
print(x)
[ 7. -1. -1.]
```

In []: