Para definir sucesiones simbólicas

1. cargar paquete sympy, con opciones lim_seq, symbols,...
2. definir símbolos
3. definir sucesión con símbolos

In [21]:
```
from sympy import limit_seq, symbols
```

In [22]:
```
n = symbols('n')
```

In [23]:
```
a=(-1)**n/n**2
```

In [24]:
```
a
```

Out[24]:
$$\frac{(-1)^{n}}{n^{2}}$$

In [25]:
```
a.subs(n, 3)
```

Out[25]:
$$-\frac{1}{9}$$

In [26]:
```
limit_seq(a, n)
```

Out[26]:
$$0$$

Otro ejemplo...

In [27]:
```
b=(n+1)*(2-3*n)/((2*n-1)*(4*n+3))
```

In [28]:
```
b
```

Out[28]:
$$\frac{(2-3n)\,(n+1)}{(2n-1)\,(4n+3)}$$

Veamos algunos valores, simbólicos y numéricos...

In [29]:
```
for i in range(0,10):
    print(b.subs(n,i))
```

```
-2/3
-2/7
-4/11
-28/75
-50/133
-26/69
-112/297
-152/403
-66/175
-250/663
```

In [30]:
```
for i in range(0,10):
    print(float(b.subs(n,i)))
```

```
-0.6666666666666666
-0.2857142857142857
-0.36363636363636365
-0.37333333333333335
-0.37593984962406013
-0.37681159420289856
-0.3771043771043771
-0.3771712158808933
-0.37714285714285717
-0.3770739064856712
```

Para calcular límites, usar comando limit_seq

In [31]:
```
print(limit_seq(a))
print(limit_seq(b))
float(limit_seq(b))
```

Out[31]:
```
0
-3/8
-0.375
```

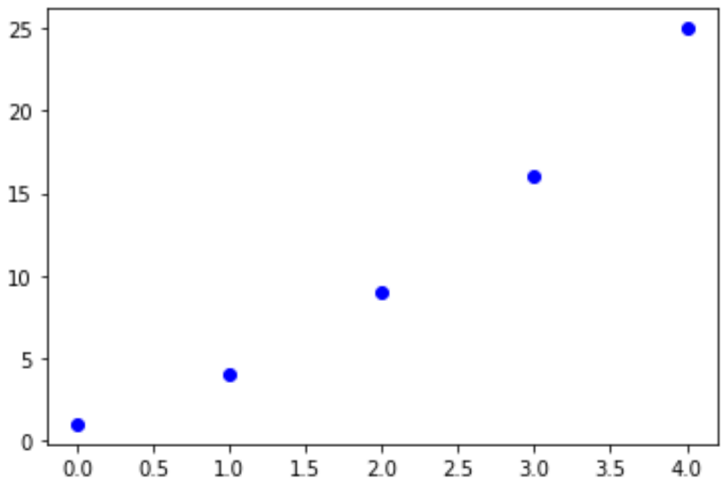
Para trabajar con sucesiones numéricas

1. cargar paquetes numpy, matplotlib.pyplot
2. definir listas
3. dibujar

In [32]:
```
import numpy as np
import matplotlib.pyplot as plt
```

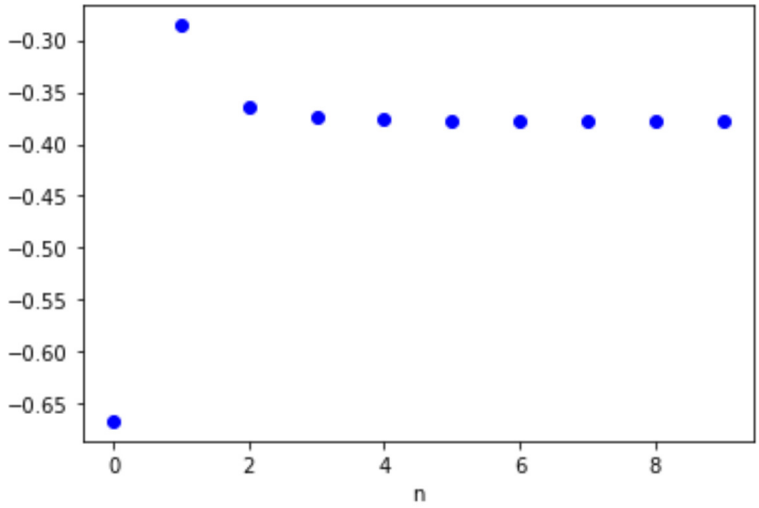
In [33]:
```
# ejemplo de sucesión introducida a mano
a=(1,4,9,16,25)
plt.plot(a,'bo')
```

Out[33]:
```
[<matplotlib.lines.Line2D at 0x27eca825910>]
```



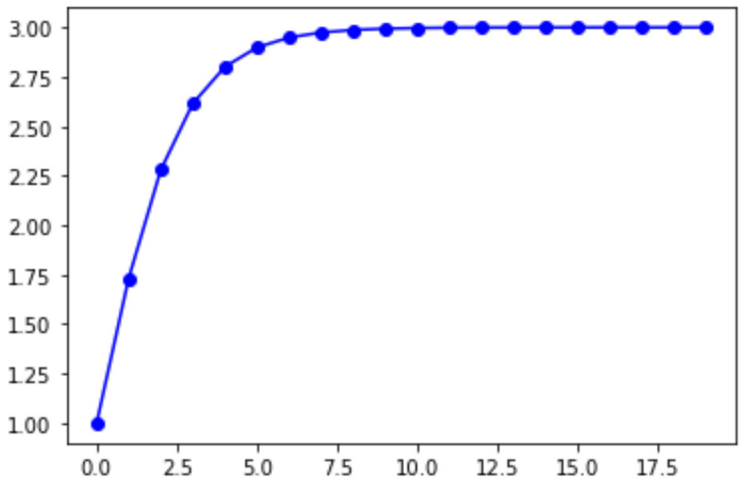

In [34]:
```
# ejemplo de sucesión, como valores de una función
def f(n):
    return (n+1)*(2-3*n)/((2*n-1)*(4*n+3))

n=np.arange(0,10, 1)

plt.plot(n,f(n),'bo')
plt.xlabel('n')
```

Out[34]: Text(0.5, 0, 'n')




Ejemplo de recurrencia

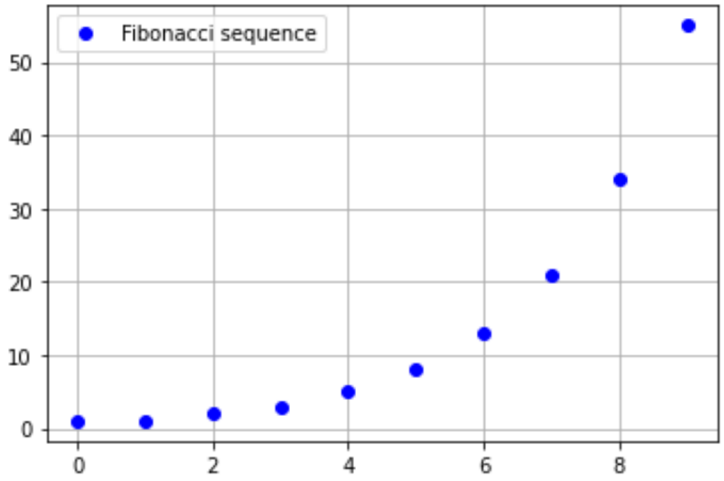
In [35]:
```
a=np.zeros(20)
a[0]=1
for i in range(0,19):
        a[i+1] = np.sqrt(3*a[i])

n=np.arange(0,20,1)
#plt.subplot(311)

plt.plot(n,a, "bo-")
plt.show()
```



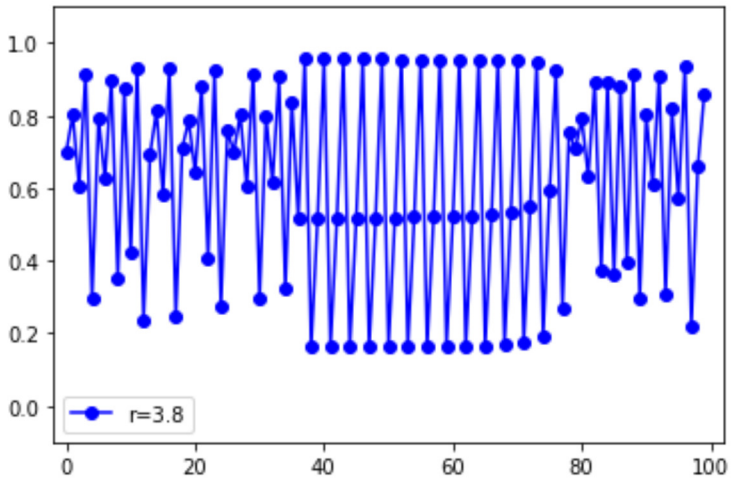

In [36]:
```
# Otro ejemplo, sucesión de Fibonacci

def fib(N):
    fib=[1,1]
    for i in range(2,N):
        next=fib[-1]+fib[-2]
        fib.append(next)
    return fib
print(fib(10))
plt.plot(fib(10),'bo', label='Fibonacci sequence')
plt.grid(True)
plt.legend(loc="upper left")
```

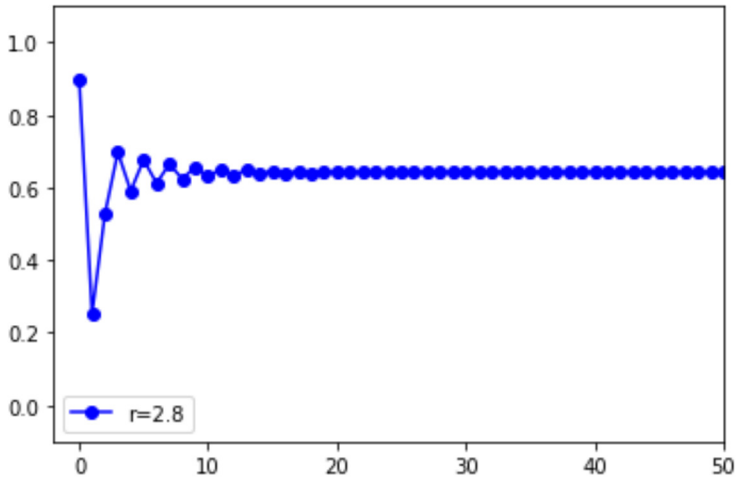
```
[1, 1, 2, 3, 5, 8, 13, 21, 34, 55]
```

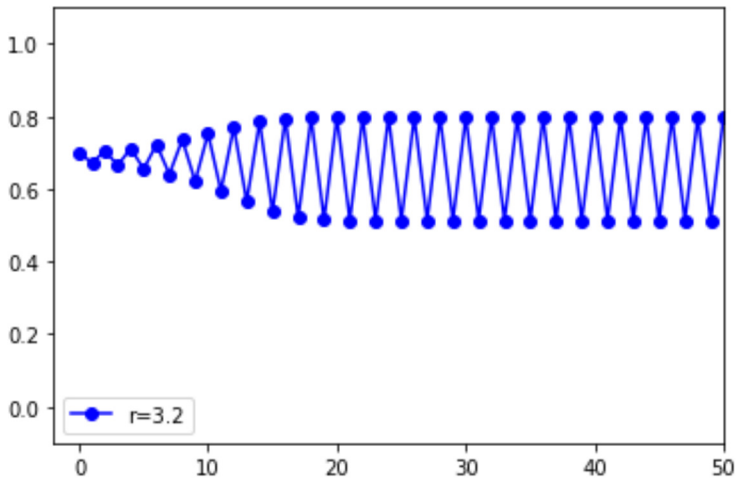
Out[36]: <matplotlib.legend.Legend at 0x27ecb0a4fd0>




In [37]:
```
# Otros ejemplo, aplicación logística

import numpy as np
import matplotlib.pyplot as plt
a=np.zeros(100)
a[0]=0.7
t=np.arange(0,100,1)
for i in range(0,99):
        a[i+1] = 3.82843*a[i]*(1-a[i])


#plt.subplot(311)
plt.axis([-2,102,-0.1,1.1])
plt.plot(t,a, "bo-", label='r=3.8')
plt.legend(loc="lower left")
plt.show()

a[0]=0.9
for i in range(0,99):
        a[i+1] = 2.8*a[i]*(1-a[i])
#plt.subplot(312)
plt.axis([-2,50,-0.1,1.1])
plt.plot(t,a, "bo-", label='r=2.8')
plt.legend(loc="lower left")
plt.show()

a[0]=0.7
for i in range(0,99):
        a[i+1] = 3.2*a[i]*(1-a[i])
#plt.subplot(313)
plt.axis([-2,50,-0.1,1.1])
plt.plot(t,a, "bo-", label='r=3.2')
plt.legend(loc="lower left")
plt.show()
```




In [ ]: