In [1]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from spicedata import get_spice_dc, get_spice_tran, get_spice_ac
```

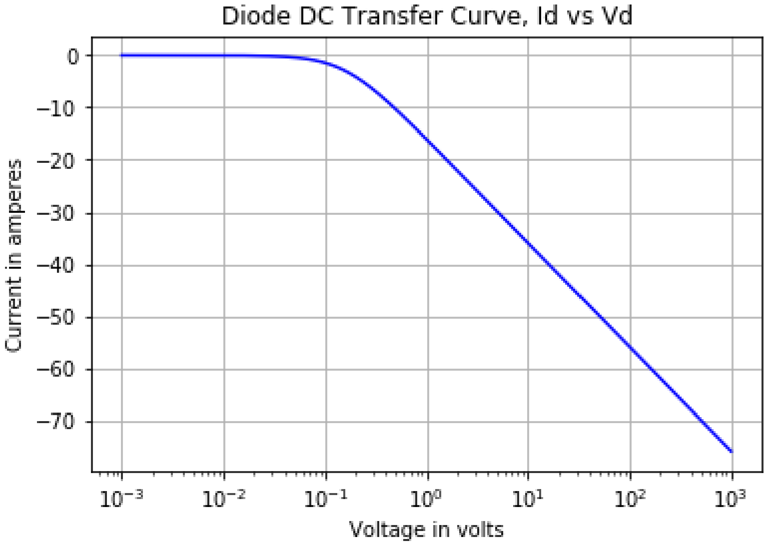
In [2]:
```python
ac = pd.DataFrame()
get_spice_ac(ac, "rc_1.out")
ac
```

Out[2]:

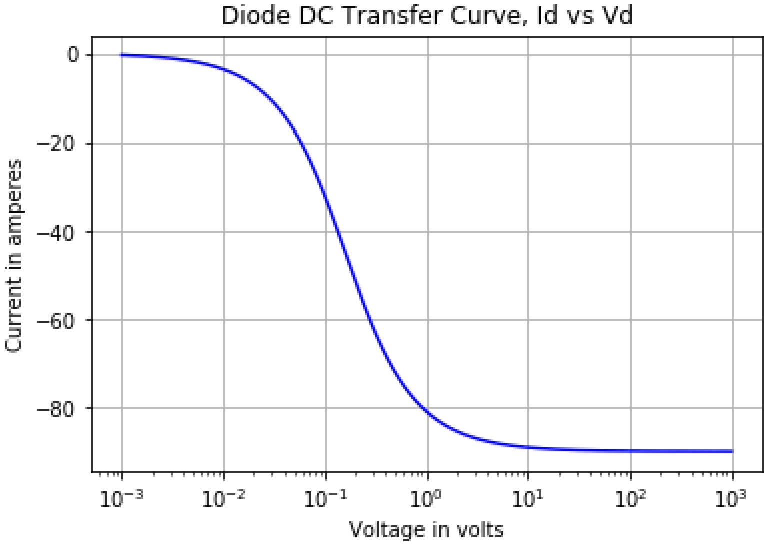
	FREQUENCY	IMAGINARY	VDB(VIN)	PHASE(VIN)	VDB(VOUT)	PHASE(VOUT)	VDB(VSIN_IN)	PHASE(VSIN_IN)	VDB(VSIN_OUT)	PHASE(VSI
0	0.001000	0.0	0.0	0.0	-0.000171	-0.359995	0.0	0.0	-0.000171	-0
1	0.001012	0.0	0.0	0.0	-0.000175	-0.364164	0.0	0.0	-0.000175	-0
2	0.001023	0.0	0.0	0.0	-0.000180	-0.368380	0.0	0.0	-0.000180	-0
3	0.001035	0.0	0.0	0.0	-0.000184	-0.372646	0.0	0.0	-0.000184	-0
4	0.001047	0.0	0.0	0.0	-0.000188	-0.376961	0.0	0.0	-0.000188	-0
...	...	...	...	...	...	...	...	...	...	
1196	954.992600	0.0	0.0	0.0	-75.563600	-89.990450	0.0	0.0	-75.563600	-89
1197	966.050900	0.0	0.0	0.0	-75.663600	-89.990560	0.0	0.0	-75.663600	-89
1198	977.237200	0.0	0.0	0.0	-75.763600	-89.990670	0.0	0.0	-75.763600	-89
1199	988.553100	0.0	0.0	0.0	-75.863600	-89.990780	0.0	0.0	-75.863600	-89
1200	1000.000000	0.0	0.0	0.0	-75.963600	-89.990880	0.0	0.0	-75.963600	-89

1201 rows × 10 columns

In [3]:
```
plt.semilogx(ac['FREQUENCY'], ac['VDB(VOUT)'], 'b-')

plt.ylabel('Current in amperes')
plt.xlabel('Voltage in volts')
plt.title('Diode DC Transfer Curve, Id vs Vd')
plt.grid(True)
plt.show()
```



In [4]:
```
plt.semilogx(ac['FREQUENCY'], ac['PHASE(VOUT)'], 'b-')

plt.ylabel('Current in amperes')
plt.xlabel('Voltage in volts')
plt.title('Diode DC Transfer Curve, Id vs Vd')
plt.grid(True)
plt.show()
```



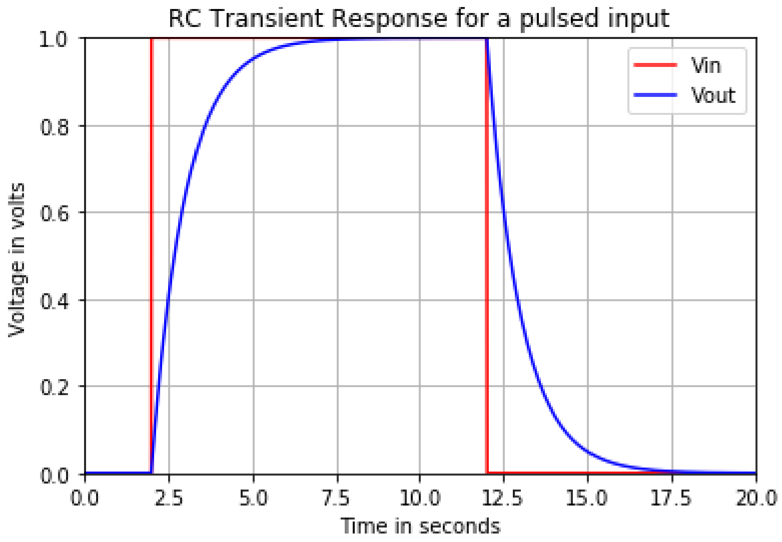
In [5]:
```python
tran = pd.DataFrame()
get_spice_tran(tran, "rc_1.out")
tran
```

Out[5]:

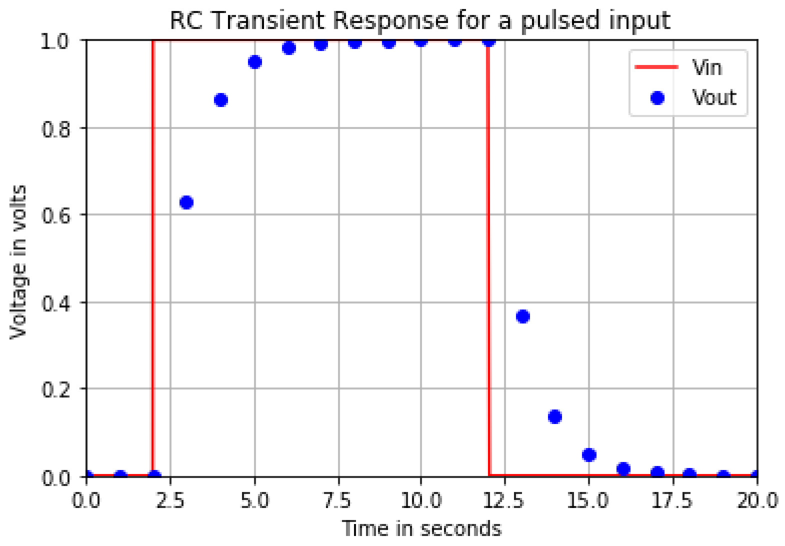
	TIME	VIN	VOUT	VSIN_IN	VSIN_OUT
0	0.000	0.0	0.000000	0.000000e+00	0.000000
1	0.001	0.0	0.000000	6.283130e-03	0.000003
2	0.002	0.0	0.000000	1.256592e-02	0.000014
3	0.003	0.0	0.000000	1.884825e-02	0.000030
4	0.004	0.0	0.000000	2.512915e-02	0.000055
...	...	...	...	...	...
19996	19.996	0.0	0.000335	-2.484664e-02	-0.151906
19997	19.997	0.0	0.000335	-1.863498e-02	-0.151883
19998	19.998	0.0	0.000335	-1.242332e-02	-0.151860
19999	19.999	0.0	0.000334	-6.211660e-03	-0.151837
20000	20.000	0.0	0.000334	9.080736e-12	-0.151814

20001 rows × 5 columns

In [6]:
```
plt.plot(tran['TIME'], tran['VIN'], 'r-')
plt.plot(tran['TIME'], tran['VOUT'], 'b-')

plt.legend(['Vin', 'Vout'], loc='best')
plt.ylabel('Voltage in volts')
plt.xlabel('Time in seconds')
plt.title('RC Transient Response for a pulsed input')
plt.grid(True)
plt.axis([0, 20, 0, 1])

plt.show()
```

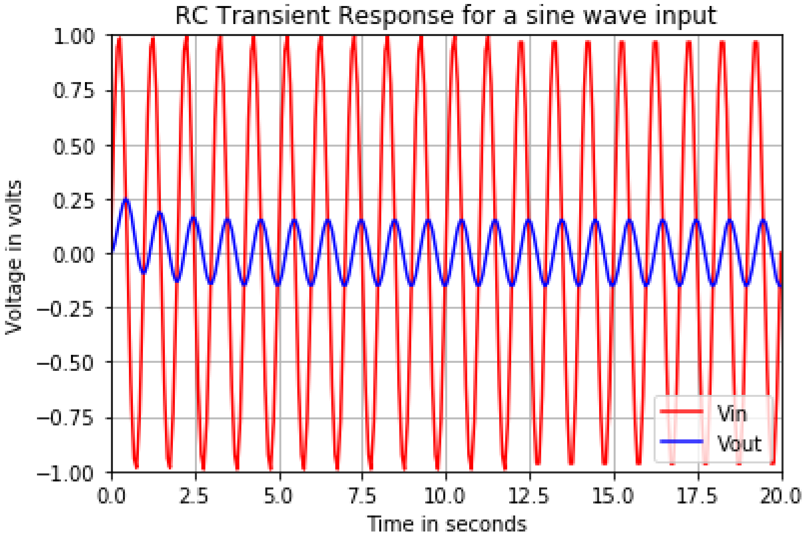


In [23]:
```
t = []
for i in range(21):
    d = tran[tran['TIME'] == i]
    t.append(d['VOUT'].values[0])
plt.plot(tran['TIME'], tran['VIN'], 'r-')
plt.plot(range(21), t, 'bo')

plt.legend(['Vin', 'Vout'], loc='best')
plt.ylabel('Voltage in volts')
plt.xlabel('Time in seconds')
plt.title('RC Transient Response for a pulsed input')
plt.grid(True)
plt.axis([0, 20, 0, 1])

plt.show()
```



In [10]:

```
plt.plot(tran['TIME'], tran['VSIN_IN'], 'r-')
plt.plot(tran['TIME'], tran['VSIN_OUT'], 'b-')

plt.legend(['Vin', 'Vout'], loc='best')
plt.ylabel('Voltage in volts')
plt.xlabel('Time in seconds')
plt.title('RC Transient Response for a sine wave input')
plt.grid(True)
plt.axis([0, 20, -1, 1])

plt.show()
```



In [ ]:

```
```