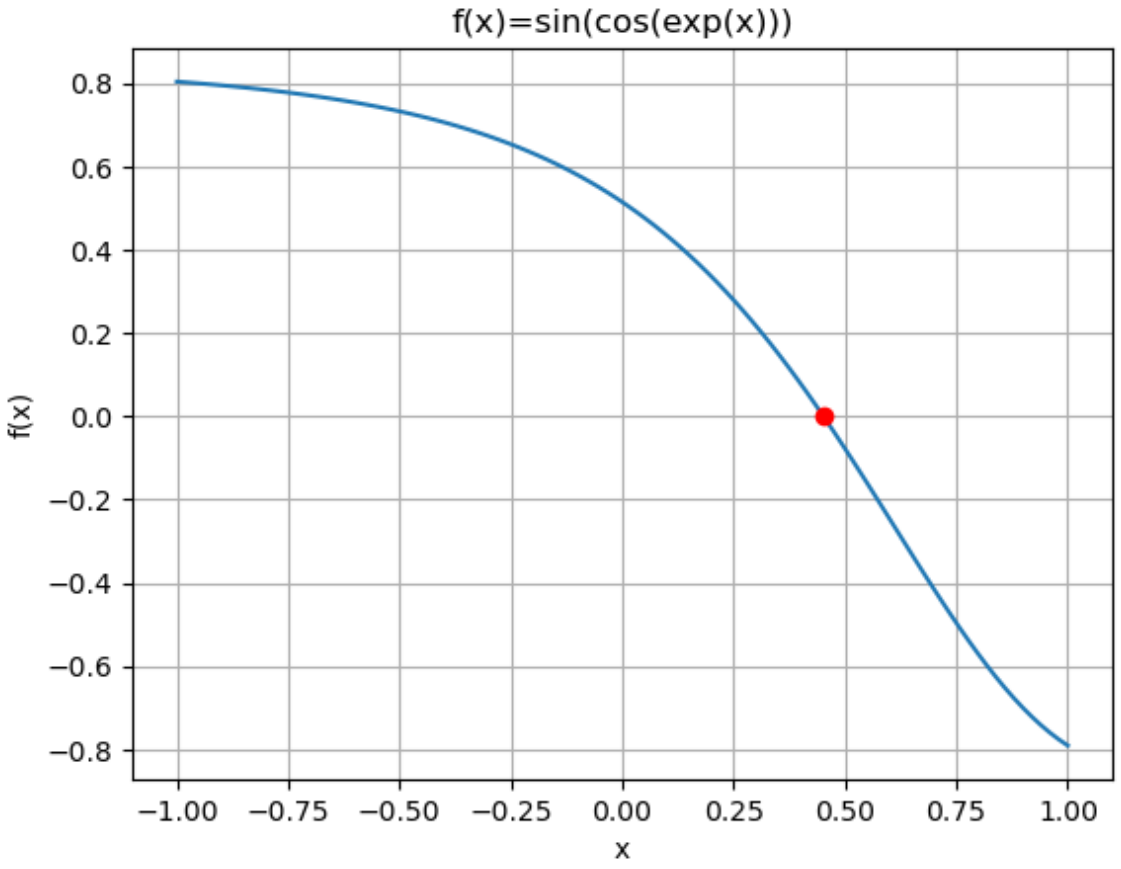
In [2]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy import optimize
```

In [12]:
```
def f(x):
    return np.sin(np.cos(np.exp(x)))
a=-1
b=1
root=optimize.bisect(f,a,b)
print("root=",root)
print("f(root)=",f(root))
plt.plot(np.linspace(-1,1,100),f(np.linspace(-1,1,100)))
plt.grid()
plt.plot(root,f(root),"or")
plt.xlabel("x")
plt.ylabel("f(x)")
plt.title("f(x)=sin(cos(exp(x)))")
```

```
root= 0.45158270529100264
f(root)= -2.4313271915891353e-12
```

Out[12]:
```
Text(0.5, 1.0, 'f(x)=sin(cos(exp(x)))')
```



In [19]:
```
"""The root obtained above by bisection method is correct(by default)
```

Out[19]:
```
'The root obtained above by bisection method is correct(by default)
upto a tolerance of 2e-12. The correct root is an irrational number
(not a machine number)'
```

In [22]:
```
def df(x):
    return -np.cos(np.cos(np.exp(x)))*np.sin(np.exp(x))*np.exp(x)
a=-1
root=optimize.newton(f,a,df)
print("root=",root)
print("f(root)=",f(root))
a=-0.1
root=optimize.newton(f,a,df)
print("root=",root)
print("f(root)=",f(root))
```

```
root= 9.179198883610521
f(root)= -5.921874465645097e-12
root= 0.4515827052894548
f(root)= 6.123233995736766e-17
```

In [25]:
```
"""The answer for x0=-1 and x0=-0.1 are different because the functi
```

Out[25]:
```
'The answer for x0=-1 and x0=-0.1 are different because the functio
n has many roots. Also it is visible that the slope of the function
is at x=-1 is flatter and hence chosing it as the initial point mak
es the series converge to a root that is far from x=-1'
```

In [29]:
```
#Using secant method
a=-0.1
root=optimize.newton(f,a)
print("root=",root)
print("f(root)=",f(root))
```

```
root= 0.4515827052894548
f(root)= 6.123233995736766e-17
```

In [ ]: