In [1]:
```
from sympy import *
```

Example 1: Solve 3y^2e^x dy/dx - x^2 = 0.

Continuing to use **dsolve** to solve in Python

In [7]:
```
x=symbols('x')
y=Function('y')
deq=3*y(x)**2*exp(x)*diff(y(x),x)-x**2
ysoln=dsolve(deq,y(x))
#print(ysoln) # Ran this line first.  Saw that only the third solution is a
real function.
print('Solution is',ysoln[2])  # NOTE that Python starts counting list elem
ents at ZERO (hence the "2" index)
```

```
Solution is Eq(y(x), (C1 - x**2*exp(-x) - 2*x*exp(-x) - 2*exp(-x))**(1/3))
```

Example 2: Solve dy/dt = (6t^5+1)/(cos(y)+e^y)

In [8]:
```
t=symbols('t')
y=Function('y')
deq=diff(y(t),t)-(6*t**5+1)/(cos(y(t))+exp(y(t)))
ysoln=dsolve(deq,y(t))
print('Solution is',ysoln)  # NOTE solution is implicit since you CANNOT so
lve for y(t)
```

```
Solution is Eq(-t**6 - t + exp(y(t)) + sin(y(t)), C1)
```

Example 3: Solve dydt = t^2 + t^2y^2, y(0)=1

In [9]:
```
t=symbols('t')
y=Function('y')
deq=diff(y(t),t)-t**2-t**2*y(t)**2
ysoln=dsolve(deq,y(t),ics={y(0):1})
print('Solution is',ysoln)
```

```
Solution is Eq(y(t), tan(t**3/3 + pi/4))
```

Example 4: Solve dx/dt = (1-2t)/x, x(1)=-2. Plot the direction field and the solution.

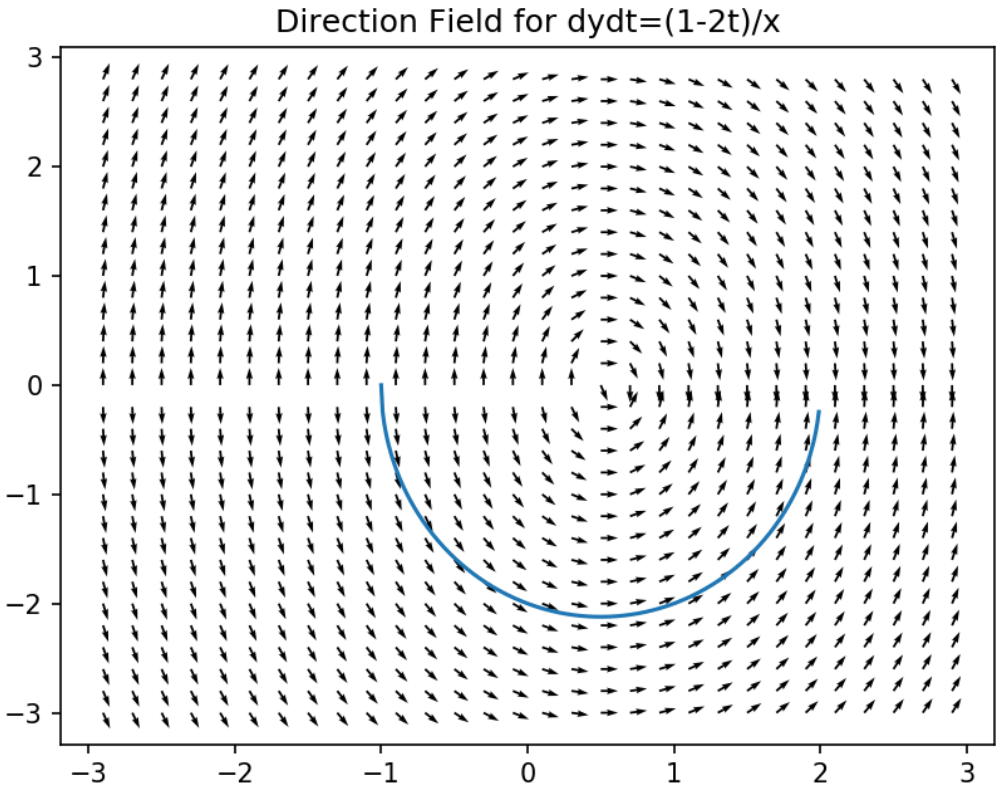
In [11]:
```
t=symbols('t')
x=Function('x')
deq=diff(x(t),t)-(1-2*t)/x(t)
xsoln=dsolve(deq,x(t),ics={x(1):-2})
print('Solution is',xsoln)
print('Since x is a negative square root, the solution is defined for x<0')
```

```
Solution is Eq(x(t), -sqrt(2)*sqrt(-t**2 + t + 2))
Since x is a negative square root, the solution is defined for x<0
```

In [14]:
```
import numpy as np
import matplotlib.pyplot as plt
```

In [15]:
```
matplotlib notebook
```

In [18]:
```
# Plot direction field with solution to IVP
T, X = np.meshgrid(np.arange(-2.9, 3.1, .2), np.arange(-3, 3, .2))
dXdT = (1-2*T)/X
U = 1/(1+dXdT**2)**0.5*np.ones(T.shape)
V = 1/(1+dXdT**2)**0.5*dXdT
plt.figure()
plt.title('Direction Field for dydt=(1-2t)/x')
Q = plt.quiver(T, X, U, V)
# It can be shown that the domain of the solution is [-1,2]
tplot=np.arange(-1,2,0.01)
yplot=-np.sqrt(-2*tplot**2+2*tplot+4)
plt.plot(tplot,yplot)
```



Out[18]:
```
[<matplotlib.lines.Line2D at 0xa32d8b0>]
```

In [ ]:
```
```