Examples of Newton's method in MATLAB

Example 1. Suppose you want to solve

$$\cos x = x$$

And suppose you observe there is a solution on $(0, 1)$. You can do:

```
>> f = @(x) cos(x) - x;
>> df = @(x) -sin(x) - 1;
>> format long g
>> x = 0.5;
>> x = x - f(x)/df(x)
x =    0.755222417105636
>> x = x - f(x)/df(x)
x =    0.739141666149879
>> x = x - f(x)/df(x)
x =    0.739085133920807
>> x = x - f(x)/df(x)
x =    0.739085133215161
>> x = x - f(x)/df(x)
x =    0.739085133215161
```

Example 2. Suppose you want to find the two places where the graph of $y = e^x$ crosses the circle of radius 2 centered at the origin. You can do:

```
>> f = @(x) [x(2) - exp(x(1)); x(1)^2 + x(2)^2 - 4];
>> Jf = @(x) [-exp(x(1)), 1; 2*x(1), 2*x(2)];
>> x = [1 1.9]';
>> p = - Jf(x) \ f(x);  x = x + p;  x'
ans =
     0.698326800574994      1.89824905232895
>> p = - Jf(x) \ f(x);  x = x + p;  x'
     0.641096397056425      1.89533090629169
>> p = - Jf(x) \ f(x);  x = x + p;  x'
     0.639264903941369      1.89508411366421
>> p = - Jf(x) \ f(x);  x = x + p;  x'
     0.639263074810244      1.89508382959371
>> p = - Jf(x) \ f(x);  x = x + p;  x'
     0.639263074808419      1.89508382959343
>> p = - Jf(x) \ f(x);  x = x + p;  x'
     0.639263074808419      1.89508382959343
```

In real life. If you are not merely playing around at the command line then you need to create a `for`-loop. What is your criterion for stopping?