```
function t_current = FallingObject()
%   Use Newton's method to approximate how long it would take a
 falling object
%   with mass m to fall from a height s0.

% number of iterations required to approximate time
how_many_iterations = 0;

% height the object is dropped from (ft)
s0 = 300;

% mass of the falling object (lb)
m = 0.25;

% gravitational acceleration (ft/s/s)
g = 32.17;

% coefficient of air resistance (lb/s)
k = 0.1;

% tolerance
TOL = 10^(-5);

% tn-1
t_last = 0;

% tn
t_current = 1;

% the formula for the height of a falling object
s  = @(t) s0 - (m*g/k)*t + ((m^2 * g)/(k^2))*(1-exp(-k*t/m));

% the derivative of the formula above
ds = @(t) (-m*g)/k + ((m*g)/k)*exp((-k*t)/m);

% Until the error range is less than tolerance,
% we continue to apply Newton's method.
while abs((t_current - t_last)/t_current) >= TOL

    % set tn-1 to the last value of tn
    t_last = t_current;

    % calculate tn = tn-1 - s(tn-1)/s'(tn-1)
    t_current = t_last - s(t_last) / ds(t_last);

    % increment how_many_iterations
    how_many_iterations = how_many_iterations + 1;

end

% print the results
fprintf("n: %d\t", how_many_iterations);
```

```
fprintf("t%d:  %.5f\t", how_many_iterations, t_current);
fprintf("|error|: %.5f\n", TOL);


end
```

```
n: 5 t5:  6.00373 |error|: 0.00001

ans =

    6.0037
```

*Published with MATLAB® R2018b*