```
function [p_current, how_many_iterations] = newton2()

% how many iterations it takes to approximate the zero of the function
how_many_iterations = 0;

% f(x) = x^2 - 2e^(-x)x + e^(-2x)
f = @(x) x*x - 2*x*exp(-x) + exp(-2*x);

% the derivative of f(x)
dfdx = @(x) 2*x +2*x*exp(-x) -2*exp(-x) -2*exp(-2*x);

% pn-1
p_last = 0;

% pn
p_current = 1;

% tolerance
TOL = 10^(-10);

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

    % set pn-1 to the last value of pn
    p_last = p_current;

    % calculate pn = pn-1 - f(pn-1)/f'(pn-1)
    p_current = p_last - (f(p_last) / dfdx(p_last));

    % increment how_many_iterations
    how_many_iterations = how_many_iterations + 1;

end

end
```

```
ans =

    0.5671
```

Published with MATLAB® R2018b