Heron's algorithm

For a nonnegative number $a \geq 0$ compute the square root $\sqrt{a}$ up to an error of 10^{-10}. For this purpose, use the following iteration:

$$x_{n+1} = \frac{1}{2}\left(\frac{x_n^2 + a}{x_n}\right).$$

Choose different initial values x_0 and observe the convergence behavior by printing the error $|x_n - \sqrt{a}|$ in each iteration step.

Hint: Use a while-loop for the iteration (`while`) and the built-in function abs as well as the Python value a `** 0.5` to compute the error $|x_n - \sqrt{a}|$ in each iteration step.

Solution:

```
def heron(a, x0=0.1, tol=10e-10, maxiter=1000):
    """
    applies the iteration rule according to the so-called "Heron method"
    """
    counter = 0
    # while the error is above our tolerance we do the iteration
    while abs(x0 - a**0.5) > tol:
        # print the current error
        print("Error =", abs(x0 - a**0.5))
        # perfom one step of the heron method:
        x0 = 0.5 * ((x0 ** 2 + a) / x0)
        # update the counter
        counter += 1
        if counter > maxiter:
            # stop the while loop if too many iterations
            break
    print("\n Result: sqrt(a) =", x0, "\n")
    return x0


if __name__ == "__main__":
    a = 2.

    # Choose different initial values
    x0 = [0.1, 1., 100000, 1.3]
    for x in x0:
        print("\n x0 =", x, "\n---------------\n")
        heron(a, x0=x)

    help(heron)
```