~***Bring consistency to my daily trades***~

# ToDo

1. Research popular languages for algo trading
2. Should I use OOP ro imprative programming?

# Algorithm

Download a .csv file of my balances in Fidelity. Write a program to read it and generate trading signals that duplicate what I already do, but in a more systematic way. SA <https://www.quantstart.com/articles/Best-Programming-Language-for-Algorithmic-Trading-Systems/> ~

1. Input a ticker symbol => Provides position details.
2. Write a method to generate trading triggers.
   1. Check cash available to trade. If cash > $10000 => Generate bias for accumulate. Else generate bias for decumulate to increase cash position.
   2. E.g. if ModAllocHi / ModAllocLoc > 1.1 => Sell ModAllocHi \* 0.02.
   3. Sort positions by balance. Starting with lowest balance position, if position value increased today => Buy more.
   4. If position is not a stock, it’s an ETF or MF. Buy lesser of 1 unit or $100.
   5. If TQQQ\_bal > 130000 => Generate bias for trim. Else generate bias for accumulate.
3. The major components of an algorithmic trading system are
   1. The research tools. Research is concerned with evaluation of a strategy performance over historical data. The process of evaluating a trading strategy over prior market data is known as backtesting. The data size and algorithmic complexity will have a big impact on the computational intensity of the backtester. CPU speed and concurrency are often the limiting factors in optimising research execution speed.
   2. Portfolio optimiser
   3. Risk manager
   4. Execution engine.
4. Creating a component map of an algorithmic trading system is worth an article in itself. However, an optimal approach is to make sure there are separate components for the historical and real-time market data inputs, data storage, data access API, backtester, strategy parameters, portfolio construction, risk management and automated execution systems.

~