More On Go / Fiance

1st. the news

1. Google is starting a blockchain/cryptocurrency system.
2. Atleast 30 major elections around the world have been run on the Blockchain. This includes a number of provinces in Spain.

2nd. Purpose of a business

1. To make a profit for the owners of the business.

What is "Fiduciary Responsibility". It means that you have been placed / are in a position of legal responsibility for managing somebody else's money.
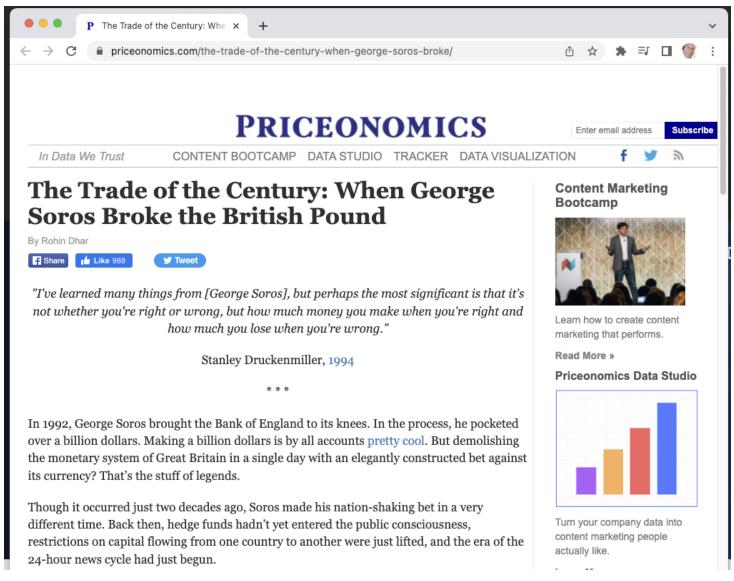
1. Definitions / Economics

"arbitrage" ::= the simultaneous buying and selling of securities, currency, or commodities in different markets or in derivative forms in order to take advantage of differing prices for the same asset.

1. How Soros broke the Bank of England.

Price Stable Cryptocurrences (or pegged cryptocurrencies)

Reference

[Soros Broke the Bank of England](#)

Personal Finance

Two Economies in US today. The "high-tech" and the "traditional" economy.

```
      winner        | per_capita_income_2020
--------------------+------------------------
 Joseph R Biden Jr  |               65531.65
 Donald J Trump     |               47960.04
(2 rows)
```

An income model.

Blockchain and Mining

What is Mining and How is it implemented.

What is proof-of-work? What is proof-of-stake? What is a public blockchian / private blockchain?

Diagram of blocks

Diagram of mining

What is the mining process

1. More on Go

Maps do not synchronize automatically. So... Synchronization Primitives:

```
 1: package main
 2:
 3: import (
 4:     "fmt"
 5:     "sync"
 6:     "time"
 7: )
 8:
 9: // SafeCounter is safe to use concurrently.
10: type SafeCounter struct {
11:     v   map[string]int
12:     mux sync.Mutex
13: }
14:
15: // Inc increments the counter for the given key.
16: func (c *SafeCounter) Inc(key string) {
17:     c.mux.Lock()
18:     // Lock so only one goroutine at a time can access the map c.v.
19:     c.v[key]++
20:     c.mux.Unlock()
21: }
22:
23: // Value returns the current value of the counter for the given key.
24: func (c *SafeCounter) Value(key string) int {
25:     c.mux.Lock()
26:     // Lock so only one goroutine at a time can access the map c.v.
27:     defer c.mux.Unlock()
28:     return c.v[key]
29: }
30:
31: func main() {
32:     c := SafeCounter{v: make(map[string]int)}
33:     for i := 0; i < 1000; i++ {
34:         go c.Inc("somekey")
35:     }
36:
37:     time.Sleep(time.Second)
38:     fmt.Println(c.Value("somekey"))
39: }
```

A Go Core/Panic

First the Code

```
 1: package main
 2:
```

```
 3: import "fmt"
 4:
 5: var mm map[string]int
 6:
 7: func main() {
 8:     fmt.Println("vim-go")
 9:     mm["bob"] = 3
10: }
```

Then the bad output.

```
vim-go
panic: assignment to entry in nil map

goroutine 1 [running]:
main.main()
        /Users/philip/go/src/github.com/Univ-Wyo-Education/S22-4010/class/lect/04/go_panic.go:9 +0x71
exit status 2
```