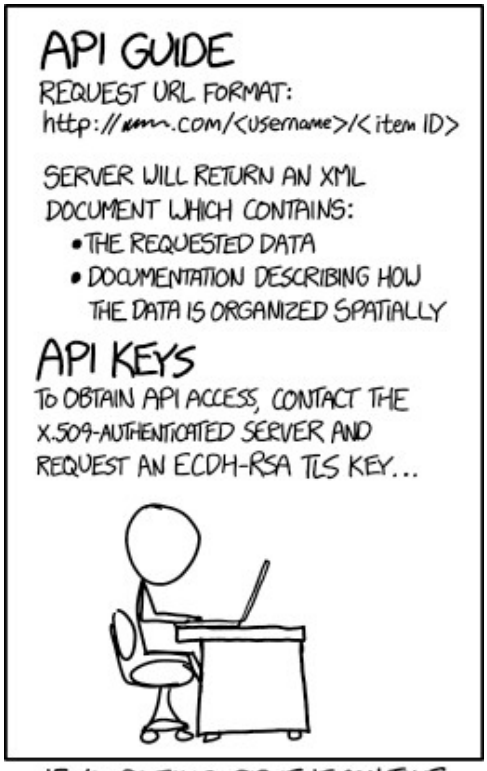
CS 326 – Project #9

Purpose: Become familiar with Google Go syntax, data types, and control structures.
Points: 100

Assignment:
Create a simple Go Language web scraper to, given a series of web sites;

- Create a list of external links (**`foundUrls.txt`**)
- Create a list of image files at the URL (**`foundImages.txt`**)
- Download image files at the URL



For the URLs, you do not need to address relative links (i.e., incomplete URL's referencing a sub-page). This will reduce the number of URLs found. The images should only be those with the "<img" tag as referenced by the "src=". This will reduce the number of images to download.

The program should display the time required to execute (including all downloads and file creations).

The scrapping for each web site should be performed in parallel (via go routines). Since the files must be performed sequentially, channels should be used to send the information back to the main where the files are created and written.

Refer to the class web page for some links regarding Go Language resources. ***Note***, you may not use an additional Go packages or imports (beyond the provided list).

Web Scrapping:
It is important to understand ethics of such technical capabilities. Miss-use of such functionality can violate a web-sites terms can conditions (and get you in trouble). Additionally, this may incur bandwidth costs, cause denial of service issues, and overflow log files. You are responsible for the actions you take including any costs or repercussions that comes along with it. When doing any scraping or crawling, you should be considerate of the server owners and prevent overloading a single site, and use reasonable settings and limits. Worst case, you might face legal problems or they may ban your account (if you have one), block your IP address, or otherwise revoke your access to the website or service.

Submission:

1) Submit a copy of the Go Language source file before class on the assigned due date.

Imports:
Below are the required imports:

```
import (
      "fmt"
      "io"
      "strings"
      "os"
      "time"
      "path"
      "net/http"
      "golang.org/x/net/html"
)
```

Given URLs:
The list of URLs is as follows:

```
urlList := []string {
   "https://www.unlv.edu/cs",
   "https://www.unlv.edu/engineering",
   "https://www.unlv.edu/engineering/advising-center",
   "https://www.unlv.edu/engineering/about",
   "https://www.unlv.edu/engineering/academic-programs",
   "https://www.unlv.edu/ceec",
   "https://ece.unlv.edu/",
   "https://www.unlv.edu/me",
   "https://www.unlv.edu/rotc",
   "https://www.unlv.edu/afrotc",
   "https://www.unlv.edu/eed",
   "https://www.unlv.edu/engineering/mendenhall",
   "https://www.unlv.edu/engineering/uas",
   "https://www.unlv.edu/engineering/solar",
   "https://www.unlv.edu/engineering/techcommercialization",
   "https://www.unlv.edu/engineering/railroad",
   "https://www.unlv.edu/engineering/future-students",
   "https://www.physics.unlv.edu/",
}
```

Yes, the comma on the last line is correct. :-)

Timing:
The timing can be accomplished as follows:

```
start := time.Now()

// program body...

elapsed := time.Since(start)
fmt.Printf("Downloads completed in %s \n", elapsed)
```