Markdown and GIT

November 2020

My first day in Markdown and Git

Getting to understand and know more about the github program.

Sean says

It seems quite useful as its collaborative positive. I see benefits which allow me to work offline or remotely without having to deal with the distraction of Internet platforms. It also allows me to write in plain text which is tres cute. Further I am very much impressed by the fact that there is capacity for accessing several saved versions of one document.

It also consumes Markdown which means .docx, .pdf, .pptx, .odt, .html and a host of other file types are grist for the old Pandoc conversion mill. (*Pandoc is a file conversion system which yeh, converts .md files to most anything under the programming sun*).

Below is an example of Markdown text. Yes since Markdown is a simple text file it can be used on the simplest of text editors on any Operating system to make those documents potnential fits for any body anywhere.

Bullets

- Demo point one
- Demo point two

- DEMO POINT THREE
- **DEMO POINT FOUR**
- *DEMO POINT FIVE*

Basically Markdown allows a writer to tap out anything that a normal document uses. And does that in a format that is still human readable. No worries either that Markdown is going anywhere. Every major and minor platform allows text.

Bertrand says

Back to GIT and branching. There is much more to this versioning exercise actually. For instance I am able to see your branch (so can you see mine if you wanted to.) Though you'd have to type

```
$ git checkout BBs-branch
```

Don't!!! Not yet. You want to master the concept of the Master and the Branch first.

You like the idea that you can create any amount of branches to keep any work - it does not have to reflect the Master. (Files on 'master' repository ought to arrive (be pushed) there only when 100% ready for whomever.

GIT and the Desktop. I'd say you are happy with the desktop graphic user interface (GUI) and yo can in fact so exist there for eternity, achieving full productivity and all but the command line interface (CLI) allows the manipulation of files i.e the

- add (Staging the file. Means you've positioned the file filename1.docx *if so named*) to the GIT selector engine.
- commit (this command technically means 'save changes' to the file you added to the lineup
- push (put the file filename.docx [if] into the online branch repository you were working on.

If you do decide to learn the CLI just use it daily until it becomes second nature. Keep creating or modifying files and adding or removing them from your branch. Once in a while yo can push a final version to the origin:17asap folder where the present structure will suffice for now.

I'm thinking you may want to download Notepad++ for text editing. Notepad is fine but it doesn't give you line numbers and a few other goodies.

Incidentally I did type this in one sitting without moving to the mouse. Took 10 mins. My CPU was at 3% average, RAM was at 12% or 4mb out of my machne's 32GB and the disk drive was chillaxing at 2-3%.

I'll keep making changes to this file as we go on. I could (but have not) easily convert to the most recent iteration of the HTML, .docx or pdf of this too but let's hold off so oyu can learn that first hand when you are ready.

About resource material. The definitive reading is here