

Mac over Windows (El Capitan vs Win7)

Here windows is at a disadvantage as we are not comparing latest windows (win 8 or 10) to latest Mac. If we are to compare windows 10 to latest Mac the overall differences maybe fewer or even down to one or two things such as UNIX shell and overall app integration on Mac. But since the windows OS at work is win7, there are more differences. Well it's unfair to compare almost 6 year old win7 (using since 2010) to latest El Capitan Mac OS which is hardly a year old.

A. Modern open source software is built and supported for Mac / Linux over PC.

The following are a few programs that I use daily at work.

- a. **Jekyll** - A static website built from Github pages doesn't officially support windows.. here is what Jekyll websites states:

A screenshot of a Jekyll website. It features a dark blue header with a white information icon (i) on the left. To the right of the icon, the text "Running Jekyll on Windows" is written in white. Below this, in a lighter blue box, is the text: "While Windows is not officially supported, it is possible to get it running on Windows. Special instructions can be found on our [Windows-specific docs page](#)." The link is highlighted in yellow.

Running Jekyll on Windows

While Windows is not officially supported, it is possible to get it running on Windows. Special instructions can be found on our [Windows-specific docs page](#).

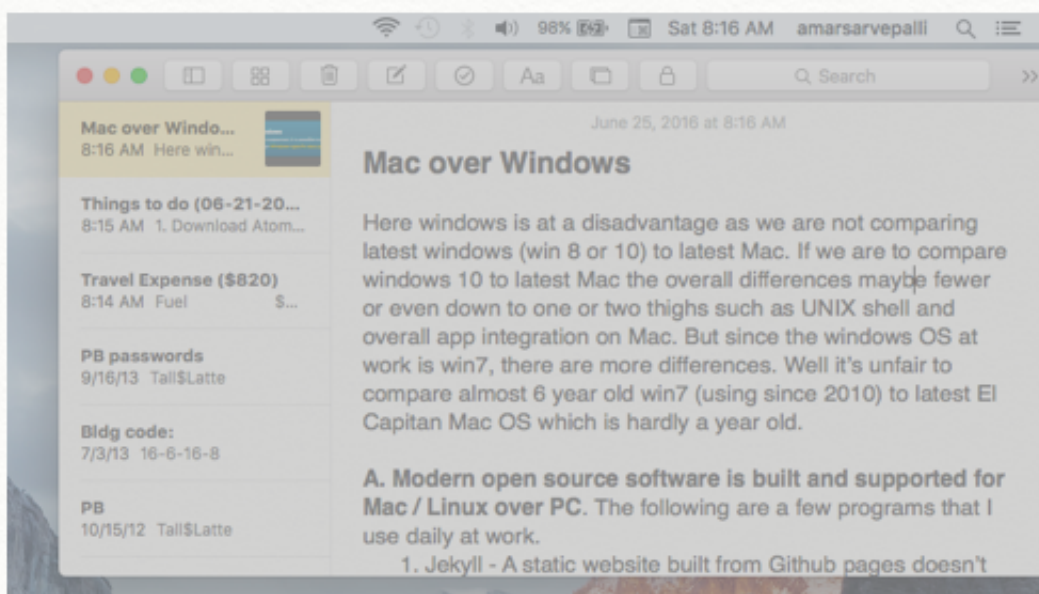
- a. **GitHub**: Command line reference is one the easiest ways to operate Github and cmd on windows is not quite the same as the UNIX bash or Mac terminal in both capacity and quality. Sometimes using git cmd is unsupported as it complains on windows for powershell versions 2.0 vs 3.0. User never knows which shell version is required. Instead of dealing with this I use UNIX bash on PC, which is nearly the same as terminal on Mac but at times it crashes.
- b. **Text editor**: Modern text editors have built-in capability to do sophisticated edits (like vertical, multi-selection or command line run). Such editors are Atom, Sublime, TextMate. However, there are significant difference between windows and mac versions as most of the windows are less supported and often buggy. Example Markdown viewer for Atom often crashes producing some log file pointing to windows drivers or kernel not being supported
- c. **Command Line Installers**: While we try to keep up-to-date on emerging technologies to produce impressive work to keep clients surprised and advance the state-of-practice, it does require some advanced tools such as command line installer. Instead of dealing with downloading a particular version of a software by going to their website and further checking for latest version, we now do via command line installers such as home brew and cask. These tools are not available for windows due to lack of UNIX integration.
- d. **Modern open source programs** (such as IBM Infosphere, Google Analytics API, IPython and most of its libraries) are all written for UNIX/ Linux, whether it's distributed systems / parallel computing or it's to run bigdata on cloud. Most of the Github is written towards Linux and several of its plugins / services are heavily supported for Linux / Mac over PC. In fact if one checks any youtube video on Git training or Python or R or BigData or another computing / analysis course, most of the trainers' screen show Mac and tutorials shortcuts are geared for Mac / Linux over PC. In most of these tutorials, there is a heavy mentioning of Mac shortcuts and tips. Although this is not a my point to argue for Mac but it does prove that modern languages and programs are well supported on Mac. Ruby, ubyGems, Ruby on Rails are another example of less supported modern language on windows . This is true even for newer versions of windows like (win 8 or win10).

B. Better hardware and Software Integrations

- a. Mac Book without any doubt comes with very nice and advanced hardware. I own a MacBook for about 6 years and has better hardware configuration than my PB work laptop which was 4 years old and died last week. Additionally, I had challenges connecting to newer routers with the PB PC laptops

over my MacBook Pro and the reason being the low end wireless cards couldn't catch latest frequencies.

- b. Battery: Again, my 6 year old MacBook is the proof that it has better batter life than my two work PCs which lasted less than 3 years befogging running into battery issues. Apart from this, single charge on Mac battery last longer than pc battery.
- c. Built-in Software: Most of the program software for Mac is either already installed such as python and gawk or easy to install with command line installers such as home brew. Installing programs on windows is not a nightmare, for example pandoc is a program to convert word document to markdown or html, that takes a challenge to install, find and run on windows. Also to mention it takes lot of man hours to constantly updating third party installations on windows as windows lack something like home brew where all packages can be updated.
- d. integration onto devices: Most of the apps are cross-platforms or cross-devices, meaning I can have a textPad app on MacBook, iPad and iPhone and start making notes on one device but can also edit on other devices. For instance I cannot write of all the differences between Mac and PC in one take and so I started this on my MacBook, and while I am walking I added a point or two to this edition from my iPhone and then when I relaxing on my patio I added couple of more points via iPad. I was able to do this as this app is available on all the devices (pre-installed) and is seamlessly synced via cloud account (gmail). Apps such as this makes taking notes lot easier and improves efficiency by not losing the thought.



C. Cross platform development

Our industry has not moved into 21st century yet as we are heavily dependent on programs and practices of the previous decades. For example, we still use

1. MS office for documentation, which hasn't improved in several years (lacks markdown and quick html rendering) and the only version control is to use track changes which is clumsy and gets cluttered if more than two people work on it. Hey, we are writing 120 page technical document and 200 page user's guide with lots of pictures, tables and graphs and there are lots of people who are contributing to this document including clients. We do need to move away fro just word documents. Similar things could be said about Excel's lack of functionality to save to JSON or XML formats. But the new generation documents are all up in cloud, managed through web-pages and version controls.
2. In this day and year, we don't need to rely on just doing work on one platform since that's only where our work actual runs (referring to exclusive based windows programs). The new generation is all about cross-platform development and web-based mobile applications. I should be able to edit a wiki-page (<https://github.com/4Step/NERPM43/releases>) or reply to clients comment or fix an issue (<https://github.com/4Step/NERPM43/issues/8>) for the user's guide I wrote while I am sipping my morning coffee rather than come-in to the office and work exclusive on a word document that is saved

only on the PB intranet. Working on newer windows (win10) is fine but since we are provided with win7, these things are not well integrated, making my day go worse in trying to get these things workings (sometime a webpage may have trouble integrating well with the text editor, sometimes it doesn't sync to cloud, sometimes UNIX terminal crashes, sometimes wifi is not connected due to low end old wifi-card and the list goes on).

- i. Increase ability to write programs in open source framework such as python or R or even newer C++ which works on multiple platforms. Unfortunately leading travel demand modeling softwares CUBE and TransCAD are still windows only applications keeping the entire practice at bay. However the two vendors are much aware of the trend and may unveil their Mac / Linux versions anytime and at that point we don't want to be the late adopters. It's not as simple as get a Mac install Mac version and build the program. This takes several months of practice ignorer to efficient build custom applications that are of impressive quality.

Overall Advantages

1. **Superior hardware**
2. **Better battery life**
3. **No bloatware/free updates**
4. **Updating is not a nightmare**
5. **Several built-in modern Apps** (such as this one where I can simple export to html and put in in my blog or write to a pbsag github page for future reference and someone can add more points and their experience and even counter this and argue for windows 11 and Linux kernel).

