The following list is a cheat sheet for the most commonly used terminal commands. There are many others, but we'll focus on these for now.

* cd changes directory
* cd ~ changes to home directory
* cd .. moves up one directory
* ls lists files in folder
* pwd shows current directory
* mkdir <FOLDERNAME> creates new directory
* touch <FILENAME> creates a file
* open . opens the current folder (Mac only)
* Navigate to your desktop or directory: cd Desktop
* Create a new directory named test: mkdir test

**Note:** When naming your directories and files, we recommend using underscores in place of spaces. So, for example, instead of naming a directory terminal prework, name it terminal\_prework.

* Create a new text file named sample: touch sample.txt
* Navigate into the test directory: cd test
* Check the contents of the directory (it should be empty): ls
* Navigate back to your desktop: cd ..
* Move the sample.txt file into the test directory: mv sample.txt test
* Make sure the previous action was successful. Use cd test, press **Enter**, and then use ls.
* Time for some independent learning! Using Google, find a way to rename the sample.txt file to example.txt.

**How do I use Git and GitHub?**

[This handy video (Links to an external site.)](https://www.youtube.com/watch?v=seICQOd2qsY) walks you through the following steps:

1. Create a new repository.
2. Copy the link.
3. In the terminal, cd into the directory where you want your repo to live.
4. Clone the repo: git clone <repo name>
5. cd into the repo.
6. Add a file.
7. Stage the changes in your current working directory: git add .
8. Commit the changes: git commit -m "some message"
9. Push the changes: git push origin main

This will push your changes directly to the main branch. The main branch is the default branch of your repository and, as its name suggests, is the primary source for your project. As you might have guessed, we do not want to push directly to the main branch. In fact, when working on a team, you likely will not have access to do so.

To get around this, we use a workflow called **feature branching**. The main idea behind feature branching is that all feature development should take place in a dedicated branch rather than pushing directly to the main branch.

Here's how you can work in this kind of environment:

* Create a new branch.
* All changes to the repository (additions, modifications, and deletions) are made in this branch.
* Commit the changes and push them to the remote source (GitHub).
* Once the branch has been pushed to the remote, open a **pull request**.
* When the pull request has been made, automated tests are often run to ensure the modifications do not break the working application.
* Pull requests also initiate a discussion about the changes you've committed, which generally must be approved by someone else on the team. That person (or entire team) will review your changes to make sure that everything is in order.
* Once any concerns have been addressed and the pull request has been approved, the modified code is merged into the main branch.

A great benefit of this process is that, once merged, pull requests act as a record of changes to the codebase.

To create a new branch:

1. cd into the repo directory.
2. Create and switch to a new branch: git checkout -b new-branch-name
3. Make the changes to your repository.
4. Stage the changes: git add -A
5. Commit the changes: git commit -m "some message"
6. Push the changes: git push origin new-branch-name
7. Navigate to your GitHub repository in your browser.
8. Click the **Pull Requests** tab to create a new pull request.

To switch between existing branches, use git checkout branch-name. Note that you do not need the -b flag in your git checkout command this time because you are not creating a new branch.

To merge your changes into main and update your local repository with those changes:

1. In GitHub, merge the pull request you created.
2. On your computer, switch back to your main branch: git checkout main
3. Pull the updates from the remote server: git pull

[Previous](https://courses.bootcampspot.com/courses/1259/modules/items/349713)[Next](https://courses.bootcampspot.com/courses/1259/modules/items/349720)

© 2020 - 2022 Trilogy Education Services, a 2U, Inc. brand. All Rights Reserved.

There's a way to compare money today versus money tomorrow. Money today, or in the present, is called **Present Value**, or PV. **Money tomorrow**, or in the future, is called Future Value, or FV. The two are linked by the expected return each time period. For example, to find out what $10,000 is worth 5 years from now, assuming a 10% annual return and assuming that profits are reinvested annually, we use this formula:

FV = PV \* [1+(i/n)]^(n\*t)

Where FV is the future value, PV is the past value, i is the interest rate, n is the number of compounding periods within a year, and t is the number of years:

FV = $10,000 \* [1+(.10/1)]^(1\*5)

FV = $10,000 \* (1+.10)^5

FV = $16,105

Here, we assumed that our returns were re-invested each year. If we put these re-invested returns to work more quickly (perhaps if we received dividend payments every quarter, reinvesting as soon as they were received), our compounding period would be shorter. Since this means we'd be investing capital earlier, a more frequent compounding period means that we'll end up with slightly more investment return in the future. Taking the same example above, returns were the same, but the returns were re-invested quarterly, instead of annually (so that the annual 10% return were received and re-invested in four equal installments each year):

FV = $10,000 \* [1+(.10/4)]^(4\*5)

FV = $10,000 \* (1+.025)^20

FV = $16,386

The annual return was the same, but we ended up with $281 more after five years because we consistently had a bit of a head start when re-investing those returns. This is why compound returns are important.

It's important to note that if you rearrange the above formula, you can solve for present value:

FV = PV \* [1+(i/n)]^(n\*t)

-->

PV = FV / [1+(i/n)]^(n\*t)

A lot of financial models for valuing stocks, companies, and even real estate use this formula. Consider why with a simple example: If there was a building for sale that you knew would be worth $100,000 next year, but you were only considering investments yielding at least a 10% return, what's the maximum price you'd offer to buy it for?

Future Value: You expect it's worth $100,000

Your Required Return: 10%

Time Horizon: 1 year

Compounding Period: 1 (only at the end of the year do you finally sell the building and earn any cash)

Using the Formula:

PV = FV / [1+(i/n)]^(n\*t)

PV = $100,000 / (1+.10)^1

PV = $90,909

**Calculating TVM**

In order to build some familiarity with the TVM equation, let's run through a quick practice problem

**Instructions**

You are presented three opportunities for investment:

* An investment that will be worth $20,000 after 2 years, compounding semi-annually at 5% interest
* An investment that will be worth $15,000 after 1 year, compounding only once at 4% interest
* An investment that will be worth $18,000 after 1 year, compounding semi-annually at 8% interest

Which investment values your present cash the highest? Which one seems riskiest? Which would you want to invest in? (Hint: This question does not have a straightforward answer)

[Previous](https://courses.bootcampspot.com/courses/1259/modules/items/349756)[Next](https://courses.bootcampspot.com/courses/1259/modules/items/349759)