Safety with sudo Quiz

```
sudo rm -rf some-file.txt
```

○ Safe

○ Dangerous

EXPLANATION

Yikes!This is **very**dangerous. If you don't have access to `rm`a file without `sudo`, there's likely a good reason. Never use `sudo`and `rm`together unless you are 100% certain you understand the potential consequences.

```
sudo ls /etc/init.d/
```

○ Dangerous

○ Safe

EXPLANATION

The `ls`command is non-destructive, so it's safe to `sudo`! You might use this while browsing lower-level files in your operating system.

```
sudo chmod +x my-script.sh
```

○ Dangerous

○ Safe

EXPLANATION

Using `chmod` to update the executable permission of a file is generally safe, but it's up to you to understand what that file will do when executed. You should never make scripts you've downloaded from the Internet executable unless you've read & understand the file's contents.

```
sudo chmod 777 ~/my-private-file
```

○ Safe

○ Dangerous

EXPLANATION

`777` is the octal permissions notation for "everyone can do everything to this file". It's very unlikely that you want any file totally accessible to every person that uses your system! Whenever you see this command, think carefully: is there a less-permissive way to grant access to only the users that need this file, maybe by adding them to a group?

```
sudo cp /var/www/index.html /var/www/index.js
```

○ Safe

○ Dangerous

EXPLANATION

The `cp` command changes the filesystem, but it doesn't remove any existing files - it just adds a new one! This is safe to `sudo` as you're very unlikely to cause negative side effects.

```
sudo mv /var/www/index.html /var/www/index.js
```

○ Safe

○ Dangerous

EXPLANATION

Using `mv` moves a file and might cause problems with other applications that depend on the original file. This is destructive behavior; you should never `sudo` destructive behavior.