(In the Case Of Linux, Stable Shell = TTY Shell)
If you ever get a NON-TTY-shell there are certain commands and stuff you won't be able to do or run.

You might get a NON-TTY-shell if, for example, you uploaded a reverse shell on a website and got a shell through that. Usually, your user on the target will be www-data or similar. These users have various limitations.

These limitations might make you frustrated and might not even allow you to do proper post-exploitation.
Anyways, if you get one of these shells you can upgrade it to a TTY shell using the following commands:

IN LINUX:

- `python -c 'import pty; pty.spawn("/bin/sh")'`
- `python3 -c 'import pty; pty.spawn("/bin/sh")'`
- `echo os.system('/bin/bash')`
- `/bin/sh -i`
- `perl –e 'exec "/bin/sh";'`
- `perl: exec "/bin/sh";`
- `ruby: exec "/bin/sh"`
- `lua: os.execute('/bin/sh')`

By the way, you can easily check if the shell is a tty shell or not by simply entering `tty` the command.

IN WINDOWS:
You don't normally need a TTY shell (or an equivalent) in windows operating systems.
If you've got a basic shell with system privilege you can possibly set up a reverse shell via Metasploit fairly easily.
This should allow you to do whatever you want.

Of course, some of these may work and some may not, depending upon your target. Maybe nothing would work on a particular system (they usually do but let's assume they don't), in such cases, **you can always google** for "methods to spawn TTY shell" and find many more methods.
Hacking (not just CTFs but in real life too) is 80% just research. Only those who love computers can manage to do so much research.

I hope this article was helpful for you,
Congratulations on learning about Network Penetration Testing.
It's time for Security Controls Evasion...
Let's Go!!!