Shortcuts:

**Windows:**

**- Alt + Tab 🡪**  Switchs between multiple windows:

**Microsoft Word:**

**- Ctr + left/Right Shift** 🡪 change paragraph direction (right 🡨🡪 left)

**VS Code:**

**Alt + F8 🡪** View problem

**Ctrl + x 🡪** delete the current line.

switch between the editor (code) and the integrated terminal:

**- Ctrl + `** → jump into terminal.

**- Ctrl + 1** → jump back to code editor.

**- Ctrl + Shift + `** → open a new terminal.

**- Ctrl + W** (inside terminal) → close that terminal.

یک سری کامندهای ریسکی در git هست که امکان حذف فایل ها و یا تغییرات commit نشده را به همراه دارد:

 git reset --hard <commit>

* overwrites local changes and deletes untracked files not in Git

 git checkout <commit> (detached HEAD)

* shows your repo exactly as it was at that commit.

 git clean -fd

* permanently deletes untracked files/folders.

 git rebase <commit>

قبل از کامندهای ریسکی git که در بالا نوشتیم چند رویکرد برای حفظ وضعیت موجود داریم:

1.

git add .

git commit -m "WIP: saving current work"

2.

git stash push -u -m "saving before risky command"

later you can see your stash list(stack) with :

git stash list

git stash apply stash@{0} # put changes back but keep them in stash list

git stash pop stash@{0} # put changes back and remove from stash list

4.

**Back up before big Git surgery**  
In real companies, you’d usually have a remote branch pushed before doing risky operations.  
That’s the safety net — you can always fetch from the remote.

**9/8/2025**

نمایش DoublyLinkedList:

    def \_\_repr\_\_(self) -> str:

        # return f"{self.\_\_class\_\_.\_\_name\_\_}([{', '.join(repr(v) for v in self)}])"

        return "{}([{}])".format(

            self.\_\_class\_\_.\_\_name\_\_,

            ", ".join(repr(v) for v in self)

        )

🡪 DoublyLinkedList([5, 7, 6, 4])